Machine Learning, 45, 77–114, 2001

Optimizing Epochal Evolutionary Search: Population-Size Dependent Theory

ERIK VAN NIMWEGEN* erik@golem.rockefeller.edu
JAMES P. CRUTCHFIELD chaos@santafe.edu
Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA

Editor: Leslie Pack Kaelbling

Abstract. Epochal dynamics, in which long periods of stasis in an evolving population are punctuated by a sudden burst of change, is a common behavior in both natural and artificial evolutionary processes. We analyze the population dynamics for a class of fitness functions that exhibit epochal behavior using a mathematical framework developed recently, which incorporates techniques from the fields of mathematical population genetics, molecular evolution theory, and statistical mechanics. Our analysis predicts the total number of fitness function evaluations to reach the global optimum as a function of mutation rate, population size, and the parameters specifying the fitness function. This allows us to determine the optimal evolutionary parameter settings for this class of fitness functions.

We identify a generalized error threshold that smoothly bounds the two-dimensional regime of mutation rates and population sizes for which epochal evolutionary search operates most efficiently. Specifically, we analyze the dynamics of epoch *destabilization* under finite-population sampling fluctuations and show how the evolutionary parameters effectively introduce a coarse graining of the fitness function. More generally, we find that the optimal parameter settings for epochal evolutionary search correspond to behavioral regimes in which the consecutive epochs are marginally stable against the sampling fluctuations. Our results suggest that in order to achieve optimal search, one should set evolutionary parameters such that the coarse graining of the fitness function induced by the sampling fluctuations is just large enough to hide local optima.

Keywords: genetic algorithm, statistical dynamics, evolutionary search, optimization, error threshold, marginal stability

1. Designing evolutionary search

Evolutionary search algorithms are a class of stochastic optimization procedures inspired by biological evolution, see for instance Bäck (1996), Goldberg (1989), Koza (1992), and Mitchell (1996): A population of candidate solutions evolves under selection and random "genetic" diversification operators. Evolutionary search algorithms have been successfully applied to a diverse variety of optimization problems, as illustrated by Belew and Booker (1991), Chambers (1995), Davis (1991), Eshelman (1995), and Forrest (1993) and references therein. Unfortunately, and in spite of a fair amount of theoretical investigation, the mechanisms constraining and driving the dynamics of evolutionary search on a given problem are often not well understood.

*Theoretical Biology and Bioinformatics Group, University of Utrecht, Padualaan 8, NL-3584-CH Utrecht, The Netherlands and current address: Center for studies in Physics and Biology, The Rockefeller University, 1230 York Avenue, New York, NY 10021 USA.

There are very natural difficulties that are responsible for this situation. In mathematical terms, evolutionary search algorithms are population-based discrete stochastic nonlinear dynamical systems. In general, the constituents of the search problem, such as the structure of the fitness function, selection, finite-population fluctuations, and genetic operators, interact in complicated ways to produce a rich variety of dynamical behaviors that cannot be easily understood in terms of the constituents individually. These complications make a strictly empirical approach to the question of whether and how to use evolutionary search problematic.

The wide range of behaviors exhibited by nonlinear population-based dynamical systems have been appreciated for decades in the field of mathematical population genetics. Unfortunately, this appreciation has not led to a quantitative predictive theory that is applicable to the problems of evolutionary search; something desired, if not required, for the engineering use of this stochastic search method.

We believe that a general, predictive theory of the dynamics of evolutionary search can be built incrementally, starting with a quantitative analytical understanding of specific problems and then generalizing to more complex situations. In this vein, the work presented here continues an attempt to unify and extend theoretical work in the areas of evolutionary search theory, molecular evolution theory, and mathematical population genetics. Our strategy is to focus on a class of problems that, despite their simplicity, exhibit some of the rich behaviors encountered in the dynamics of evolutionary search algorithms. Using analytical tools from statistical mechanics, dynamical systems theory, and the above mentioned fields, we developed a detailed and quantitative understanding of the search dynamics for a class of problems that exhibit epochal evolution. On the one hand, as we show here, this allows us to analytically predict optimal parameter settings for this class of problems. On the other hand, the detailed understanding of the behavior for this class of problems provides valuable insights into the emergent mechanisms that control the dynamics in more general settings of evolutionary search and in other population-based dynamical systems.

In previous papers (van Nimwegen & Crutchfield, 2000; van Nimwegen, Crutchfield, & Mitchell, 1997, 1999) we analyzed in some detail the metastable population dynamics of what 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. This *punctuated equilibrium* behavior is a common occurrence in both natural and artificial evolutionary processes, see for instance Adami (1995), Crutchfield and Mitchell (1995), Elena, Cooper and Lenski (1996), Fontana and Schuster (1998), Gould and Eldredge (1977), Mitchell, Crutchfield and Hraber (1994a).

For populations evolving under a static fitness function, which is typically the case in evolutionary search, it has been commonly assumed that *local optima* in the fitness "landscape" are responsible for metastability in the population dynamics. The geographic metaphor of a population crawling up the slopes of a fitness or adaptive landscape was originally introduced by the evolutionary biologist Sewall Wright, see e.g. Wright (1982). More recently, it has been assumed by several authors, such as Kauffman and Levin (1987), and Macken and Perelson (1989), that the typical fitness functions of combinatorial optimization and biological evolution can be modeled as "rugged landscapes". Rugged landscapes are fitness functions with wildly fluctuating fitnesses even at the smallest scales of single-point mutations. It is natural to assume that such rugged landscapes possess a large number of local optima. With this picture in mind, one would explain punctuated equilibria as the result of the population getting "pinned" at a local optimum in the landscape, until a rare lineage of mutants crosses a valley of low fitness to a different, higher local optimum.

In contrast, there has been an increasing realization in recent years that the large degeneracies that occur in biological fitness functions play an important role in evolutionary dynamics, as originally argued by Huynen, Stadler, and Fontana (1996). Such degeneracies have also been observed in evolutionary search problems—see for instance Crutchfield and Mitchell (1995)—and typically occur when there is redundancy in the genetic representation (*genotype*) of candidate solutions to a combinatorial optimization problem. When these degeneracies are operating, the set of all genotypes breaks 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 simply connected components—the members of which can be reached via paths made of single-mutation steps. Such components are generally referred to as *neutral networks* in molecular evolution theory (Fontana & Schuster, 1998; Huynen Stadler & Fontana 1996; Huynen, 1995; Reidys, Forst, & Schuster, 2001; Weber, 1996). Epochal behavior occurs in evolution under these fitness functions because members of the evolving population must search through most of the network of neutral variants before a connection to a neighboring network of higher fitness is discovered; during this time the average fitness is constant, up to fluctuations.

In our analysis of epochal evolution (van Nimwegen, Crutchfield, & Mitchell, 1997, 1999) we view large degeneracies in the genotype-to-fitness mapping as the main source of the epochal nature of the evolutionary dynamics. We have constructed a wide class of fitness functions that realize our view of genotype space decomposing into a relatively small collection of entangled neutral networks and analyzed the resulting evolutionary population dynamics. In a previous paper (van Nimwegen & Crutchfield, 2000) we showed how this detailed dynamical understanding can be turned to practical advantage by analytically determining the mutation rates to reach, in the fewest number of fitness function evaluations, the global optimum in this class of fitness functions. Here we recount our basic analytical approach and extend it to incorporate population-size-dependent dynamical effects. As will be explained below, population-size effects enter primarily through the dependence of the stability of an epoch's metastable population on finite-population sampling fluctuations. The result is a more general and accurate theory that analytically predicts the total number of fitness function evaluations needed on average for the algorithm to discover the global optimum of the fitness function as a function of both mutation rate and population size.

In addition, we develop a detailed understanding of the operating regime in parameter space for which the search is performed most efficiently. We believe this will provide useful guidance on how to set search algorithm parameters for more complex problems. In particular, our theory explains how optimal search occurs in the parameter regime where metastable populations are only marginally stable. The results raise the general question of whether it is desirable for optimal search to run in dynamical regimes that are a balance of stability and instability. More specifically, we show how the interplay of mutation, selection, and finite-population sampling fluctuations effectively induces a coarse graining of the fitness function. That is, genotypes with fitnesses within a narrow range of each other are effectively treated as equal by the evolutionary dynamics. Based on this, we conjecture that optimal search occurs when the level of this coarse graining is just enough to hide local optima by rendering them dynamically unstable.

2. Royal Staircase fitness functions

Choosing a class of fitness functions, whose population dynamics one wishes to analyze, is a delicate compromise between generality, mathematical tractability, and the degree to which the class is representative of problems often encountered in evolutionary search. A detailed knowledge of the fitness function is very *atypical* of evolutionary search problems. If one knew the fitness function in detail, one would not have to run an evolutionary search algorithm to find high-fitness solutions in the first place. The other extreme of assuming complete generality, however, cannot lead to enlightening results either, since averaged over all problems, all optimization algorithms perform equally well (or badly), as shown in Wolpert and Macready (1997). We thus focus on a specific subset of fitness functions, somewhere between these extremes, that we believe at least have ingredients typically encountered in evolutionary search problems and that exhibit dynamical behaviors widely observed in both natural and artificial evolutionary processes.

As explained in the previous section, we focus on fitness functions that induce a collection of entangled neutral networks: genotype space decomposes into a set of (large) networks of isofitness genotypes that are connected via point mutation steps. Consequently, the number of different fitness values that genotypes can take is much smaller than the number of different genotypes. We also assume that higher-fitness networks are smaller, i.e., contain fewer genotypes, than low-fitness networks. Finally, we assume that from any neutral network there exist connections to higher-fitness networks such that, taken as a whole, the fitness landscape has no local optima other than the global optimum.

Under these assumptions, genotype space takes on a particular type of architecture: *subbasins* of the neutral networks are connected by *portals* leading between them and so to higher or lower fitness. Stated in the simplest terms possible, the evolutionary population dynamics then becomes a type of diffusion constrained by this architecture. For example, individuals in a population diffuse over neutral networks until a portal to a network of higher fitness is discovered and the population moves onto this network.

Viewed from a somewhat different perspective, such neutral network architectures in genotype space may be induced from any fitness function by coarse graining the fitness values into a small number of fitness classes. Under such coarse graining, genotypes whose fitnesses fall into the same fitness class are treated as mutually neutral under selection. For instance, neutral networks in "*NK* fitness landscapes" (Kauffman, 1993) have been constructed in this way by Barnett (1997) and Newman and Engelhardt (1998). As we show below, explicit constructions may not be necessary: the evolutionary parameters themselves effectively induce a coarse graining of the fitness function. To some extent, this justifies our grouping fitness values into a relatively small number of fitness classes. Moreover, we will argue that an optimal setting of evolutionary parameters for efficient search is achieved when, in effect, these parameters induce the “right” coarse graining of the fitness function.

In order to model the evolutionary behavior associated with neutral network architectures, we defined, in a previous paper (van Nimwegen & Cruthchfield, 1999), the class of *Royal Staircase* fitness functions that capture the essential elements sketched above. Importantly, this class of fitness functions is simple enough to admit a fairly detailed quantitative mathematical analysis of the associated epochal evolutionary dynamics.

The Royal Staircase fitness functions are defined as follows.

1. Genotypes are specified by binary strings $s = s_1 s_2 \cdots s_L$, $s_i \in \{0, 1\}$, of length $L = NK$.
2. Reading the genotype from left to right, the number $I(s)$ of consecutive 1s is counted.
3. The fitness $f(s)$ of genotype s with $I(s)$ 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 the following with regard to this definition.

1. The fitness function has two parameters, the number N of blocks and the number K of bits per block. Fixing them determines a particular optimization problem.
2. There is a single global optimum: the genotype $s = 1^L$—namely, the string of all 1s—with fitness $f(s) = N + 1$.
3. The proportion ρ_n of genotype space filled by strings of fitness n is given by:

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

   for $n \leq N$. Thus, high-fitness strings are exponentially more rare than low-fitness strings.
4. For each block of K bits, the all-1s pattern is the one that confers increased fitness on a string. Without loss of generality, any of the other $2^K - 1$ configurations could have been chosen as the “correct” configuration, including different patterns for each of the N blocks. Furthermore, since the evolutionary search here does not use crossover, arbitrary permutations of the L bits in the fitness function definition leave the evolutionary dynamics unchanged.

By implementing the architecture of neutral networks in this way, high-fitness neutral networks are nested inside lower-fitness networks. Higher fitness strings are rarer since they require more bits in the genotype to be set “correctly”. Each step upward in fitness is associated with setting an additional K bits in the genotype correctly. One can only set correct bit values in sets of K bits at a time, creating an *aligned* block, and in blocks from left to right. A genotype’s fitness is proportional to the number of such aligned blocks. Since the $(n + 1)$st block only confers fitness when all n previous blocks are aligned as well, there is contingency between blocks.

Using the same analysis as presented below one can analyze more complex cases in which different blocks have different numbers of bits and networks are entangled in more complicated ways than the simple nesting chosen here. However, the main conclusions of our analysis can be more transparently presented using this relatively simple Royal Staircase class. The reader is referred to Crutchfield and van Nimwegen (1999) for an outline of the application of our analysis to a broad class of more complex fitness functions.

3. 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, 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, i.e., change each bit in each string of the new population with probability q.
6. Go to step 2.

When the algorithm terminates there have been $E = Mt_{\mathrm{opt}}$ fitness function evaluations.

Notice that this algorithm omits the often-used crossover operator. The main reason for excluding crossover is that it greatly simplifies the analysis. However, with respect to epochal evolution, the addition of crossover does not significantly alter or improve the evolutionary search behavior. We will provide some arguments for this claim below. For a more detailed discussion of crossover's lack of effectiveness in improving in *epochal* evolutionary search, the reader is referred to van Nimwegen and Crutchfield (2000).

Our GA effectively has two parameters: the mutation rate q and the population size M. A given optimization problem is specified by the fitness function in terms of N and K. Stated most prosaically, then, the central goal of the following analysis is to find those settings of M and q that minimize the average number $\langle E \rangle$ of fitness function queries for given N and K required to discover the global optimum genotype of fitness $N + 1$. Our approach is to develop analytical expressions for E as a function of N, K, M, and q and then to study the *search-effort surface* $E(q, M)$ at fixed N and K. Before beginning the analysis, however, it is helpful to develop an appreciation of the basic dynamical phenomenology of evolutionary search on this class of fitness functions. Then we will be in a position to lay out the evolutionary equations of motion and analyze them.

4. Observed population dynamics

The typical behavior of a population evolving under a fitness function that induces connected neutral networks, such as defined above, alternates between long periods (*epochs*) of stasis in the population's average fitness and sudden increases (*innovations*) in the average fitness. We now briefly recount the experimentally observed behavior of typical Royal Staircase GA runs in which the parameters q and M are set close to their optimal setting. The reader

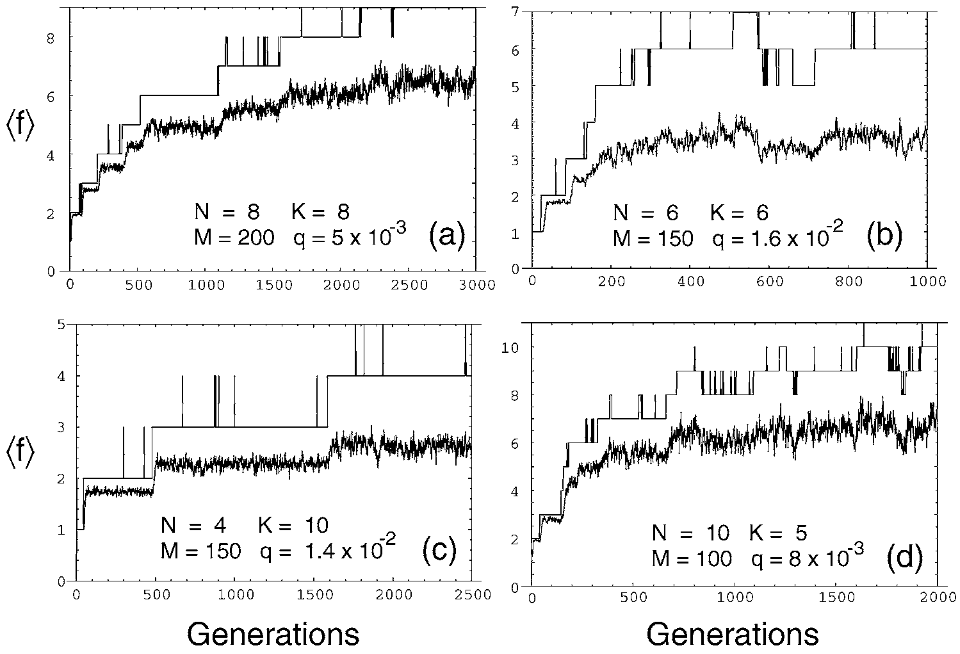


Figure 1. Examples of the Royal Staircase GA population dynamics with different parameter settings. The four plots show best fitness in the population (upper lines) and average fitness in the population (lower lines) as a function of time, measured in generations. The fitness function and GA parameters 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.

is referred to van Nimwegen, Crutchfield, and Mitchell (1999) for a detailed discussion of the other dynamical behaviors this type of GA exhibits over a range of different parameter regimes.

Figure 1 illustrates the GA's behavior at four different parameter settings. Each individual figure plots the best fitness in the population (upper lines) and the average fitness $\langle f \rangle$ in the population (lower lines) as a function of the number of generations. Each plot 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 each run, especially for those in figure 1(b) and (d), $\langle f \rangle$ tends to have higher fluctuations and the epochal nature of the dynamics becomes unclear.

In the GA runs, the population starts out with strings that only have relatively low fitness, say fitness n. Selection and mutation then establish an equilibrium in the population until a string aligns the nth block and descendants of this string with fitness $n + 1$ spread through the population. A new equilibrium is then established until a string of fitness $n + 2$ is discovered and so on, until finally a string of fitness $N + 1$ is discovered. In the figure we let the runs continue past the generation at which the global optimum was first discovered. One observes that, except for the run in figure 1(a), the strings of fitness $N + 1$ do not manage to stabilize themselves in the population.

Notice that the behavior of the average fitness $\langle f \rangle$ roughly tracks the epochal behavior of the best fitness in the population. Every time a newly discovered higher-fitness string has spread through the population, $\langle f \rangle$ reaches a new, higher equilibrium value around which it fluctuates. As a run progresses to higher epochs, $\langle f \rangle$ tends to have higher fluctuations and the epochal nature of the dynamics is obscured. This is a result of the fact that for the highest epochs the difference between $\langle f \rangle$ in consecutive epochs is smaller than the average fitness fluctuations induced by the finite-population sampling; see van Nimwegen, Crutchfield, and Mitchell (1999) for an analytical treatment of this particular phenomenon.

Notice, too, that often the best fitness shows a series of brief jumps to higher fitness during an epoch. When this occurs, strings of higher fitness are discovered but, rather than spreading through the population, are lost within a few generations.

For each of the four settings of N and K we have chosen values of q and M such that the average total number $\langle E \rangle$ of fitness function evaluations to reach the global optimum for the first time is minimal. Thus, the four plots illustrate the GA's typical dynamics close to optimal (q, M)-parameter settings.

Despite what appears at first blush to be relatively small variations in fitness function and GA parameters, there is a large range, almost a factor of 10, in times to reach the global optimum across the runs. One concludes that there can be a strong parameter dependence in search times. It also turns out that the standard deviation σ of the mean total number $\langle E \rangle$ of fitness function evaluations is of the same order as $\langle E \rangle$. (See Table 1.) Thus, there are large run-to-run variations in the time to reach the global optimum. This is true for all parameter settings with which we experimented, of which only a few are reported here.

Having addressed the commonalities between runs, we now turn to additional features that each illustrates. Figure 1(a) shows the results of a GA run with $N = 8$ blocks of $K = 8$ bits each, a mutation rate of $q = 0.005$, and a population size of $M = 200$. During the later epochs, the best fitness in the population hops up and down several times before it finally jumps up and the new more-fit strings stabilize in the population. In this particular run, it took the GA approximately 3.4×10^5 fitness function evaluations (1700 generations) to discover the global optimum for the first time. Over 500 runs, the GA takes on average 5.3×10^5 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 fewer than 10^5 fitness function evaluations and that others take more than 10^6. As our analysis will make clear, these large run-to-run variations are endogenous to the GA dynamics and cannot be

Table 1. Mean $\langle E \rangle$ and standard deviation σ of the expected number of fitness function evaluations for the Royal Staircase fitness functions and GA parameters shown in the runs of figure 1.

N	K	M	q	$\langle E \rangle$	σ
8	8	200	0.005	5.3×10^5	2.1×10^5
6	6	150	0.016	5.5×10^4	3.0×10^4
4	10	150	0.014	1.9×10^5	1.0×10^5
10	5	100	0.008	1.2×10^5	4.9×10^4

The estimates were made from 500 GA runs and so the standard error in our estimates for $\langle E \rangle$ are the values of σ divided by $\sqrt{500}$.

reduced by changes in the parameters q and M. In fact, the run-to-run variations are minimal where the mean $\langle E \rangle$ itself is minimal.

Figure 1(b) plots a run with $N = 6$ blocks of length $K = 6$ bits, a mutation rate of $q = 0.016$, and a population size of $M = 150$. The GA discovered the global optimum after approximately 4.8×10^4 fitness function evaluations (325 generations). For these parameters, the GA uses approximately 5.5×10^4 fitness function evaluations on average to reach the global fitness optimum. Notice that the global optimum is only consistently present in the population between generations 530 and 570. After that, the global optimum is lost again until after generation 800. As we will show, this is a typical feature of the GA's behavior for parameter settings close to those that give minimal $\langle E \rangle$. 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 (the optimum).

Figure 1(c) shows a run for a small number ($N = 4$) of large ($K = 10$) blocks. The mutation rate is $q = 0.014$ and the population size is again $M = 150$. As in the three other runs we see that $\langle f \rangle$ goes through epochs punctuated by rapid increases in $\langle f \rangle$. We also see that the best fitness in the population typically jumps several times before the population fixes on a higher-fitness string. The GA takes about 1.9×10^5 fitness function evaluations on average to discover the global optimum for these parameter settings. In this run, the GA first discovered the global optimum after 2.7×10^5 fitness function evaluations. Notice that the optimum never stabilized in the population.

Finally, figure 1(d) shows a run with a large number ($N = 10$) of relatively small ($K = 5$) 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 8–10 before it reaches (fleetingly) the global fitness optimum of 11. Quickly after it has discovered the global optimum, it disappears again and the best fitness in the population largely 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 $\langle f \rangle$. It appears to be lost in the "noise" of the average fitness fluctuations. The GA takes about 1.2×10^5 fitness function evaluations on average at these parameter settings to reach the global optimum; while in this particular run the GA took 1.6×10^5 fitness function evaluations (1640 generations) to briefly reach the optimum for the first time.

5. Statistical dynamics of evolutionary search

In previous papers (van Nimwegen, Crutchfield, & Mitchell, 1997, 1999) we developed the statistical dynamics of genetic algorithms to analyze the behavioral regimes of a GA searching the Royal Road fitness functions, which are closely related to the Royal Staircase fitness functions that we study here. The analysis here builds on those results and, additionally, is a direct extension of the optimization analysis and calculations that we published previously (van Nimwegen & Crutchfield, 2000). We briefly review the essential points from these previous papers. We refer the reader to van Nimwegen, Crutchfield, and Mitchell (1999) for a detailed description of the similarities and differences of our theoretical approach with other theoretical approaches such as the work by Prügel-Bennett, Rattray, and

Shapiro (Prügel-Bennett & Shapiro, 1994, 1997; Rattray & Shapiro, 1996), the diffusion equation methods from mathematical population genetics developed by Kimura (Kimura, 1964, 1983), and the quasispecies theory of molecular evolution (Eigen, McCasKill, & Schuster, 1989).

5.1. Macrostate space

Formally, the state of a population in an evolutionary search algorithm is only specified when the frequency of occurrence of each of the 2^L genotypes is given. Since 2^L is typically very large, the dimension of the corresponding microscopic state space is very large as well. One immediate consequence is that the evolutionary dynamic, on this level, is given by a stochastic (Markovian) operator of size (at least) $\mathcal{O}(2^L \times 2^L)$. Generally, using such a microscopic description makes analytical and quantitative predictions of the GA's behavior unwieldy. Moreover, since the practitioner is generally interested in the dynamics of some more macroscopic statistics, such as best and average fitness, a microscopic description is uninformative unless an appropriate projection onto the desired macroscopic statistic is found.

With these difficulties in mind, we choose to describe the macroscopic state of the population by its fitness distribution, denoted by a vector $\vec{P} = (P_1, P_2, \ldots, P_{N+1})$, where the components $0 \leq P_f \leq 1$ are the proportions of individuals in the population with fitness $f = 1, 2, \ldots, N + 1$. We refer to $\vec{P}$ as the *phenotypic quasispecies*, following its analog in molecular evolution theory (Eigen, 1971; Eigen, McCaskill, & Schuster, 1989; Eigen & Schuster, 1977). Since $\vec{P}$ is a distribution, it is normalized:

$$\sum_{f=1}^{N+1} P_f = 1. \tag{2}$$

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

$$\langle f \rangle = \sum_{f=1}^{N+1} f P_f. \tag{3}$$

5.2. The evolutionary dynamic

The fitness distribution $\vec{P}$ does not uniquely specify the microscopic state of the population. That is, there are many microstates (genotype distributions) with the same fitness distribution. An essential ingredient of the statistical dynamics approach is to assume a maximum entropy distribution over microstates conditioned on the macroscopic fitness distribution. Note that our approach shares a focus on fitness distributions and maximum entropy methods with that of Prügel-Bennett, Rattray, and Shapiro (Prügel-Bennett & Shapiro, 1994, 1997; Rattray & Shapiro, 1996). In our case, the maximum entropy assumption entails that, given a fitness distribution $\vec{P}(t)$ at generation t, each microscopic population state with this fitness distribution is equally likely to occur.

A few comments on this maximum entropy method are in order. For the maximum entropy assumption to be useful it is not strictly necessary that the population takes on all genotype distributions equally often over the ensemble of instances for which a given fitness distribution occurs. (In fact, it is not difficult to find counterexamples for which this is almost certainly false—and false for several reasons.) In order for the method to work we only require that the dynamics on the level of the fitness distributions, as calculated using the maximum entropy assumption, corresponds to the actual dynamics of the fitness distribution. That is, as long as the deviations of the actual genotype distributions from the maximum entropy distribution do not introduce a "bias" on the level of fitness distributions, the predictions will not be affected. Deciding whether or not the maximum entropy assumption works follows from comparing the theoretical predictions to data from simulations. In the case that the maximum entropy assumption *does* break down, it simply points out that additional macroscopic variables are needed to describe the macroscopic dynamics in which we are interested. For instance, in the following, ultimately we are only interested in the dynamics of the best fitness in the population. However, taking the best fitness as the only variable describing the population and then introducing the maximum entropy assumption leads to unacceptably poor theoretical predictions. Rather we need to use the entire fitness distribution.

Given the maximum entropy assumption on the level of fitness distributions, we can construct a generation operator **G** that acts on the current fitness distribution and gives the *expected* fitness distribution of the population at the next time step. In the limit of infinite populations, which is similar to the thermodynamic limit in statistical mechanics, the fluctuations due to the finite size of the population are damped out, and this expected distribution is always exactly realized at the next generation. That is, the operator **G** maps the current fitness distribution $\vec{P}(t)$ deterministically to the fitness distribution $\vec{P}(t+1)$ at the next time step;

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

Simulations indicate that for very large populations ($M \gtrsim 2^L$) the dynamics on the level of fitness distributions is indeed deterministic and given by the above equation; thereby justifying the maximum entropy assumption at least in this infinite-population limit.

The operator **G** consists of a selection operator **S** and a mutation operator **M**:

$$\mathbf{G} = \mathbf{M} \cdot \mathbf{S}.$$

The selection operator encodes the fitness-level effect of selection on the population; and the mutation operator, the fitness-level effect of mutation. Appendices A and B review the construction of these operators for our GA and the Royal Staircase fitness functions.

For now, we note that the infinite-population dynamics can be obtained by iteratively applying the operator **G** to the initial fitness distribution $\vec{P}(0)$. Thus, the solutions to the macroscopic equations of motion, in the limit of infinite populations, are formally given by

$$\vec{P}(t) = \mathbf{G}^{(t)}[\vec{P}(0)]. \tag{4}$$

Recalling Eq. (1), it is easy to see that the initial fitness distribution $\vec{P}(0)$ is given by:

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

and

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

As we showed previously (van Nimwegen, Crutchfield, & Mitchell, 1997, 1999), the equations of motion of Eq. (4) can be linearized in a straightforward manner by introducing a linearized generator operator $\tilde{\mathbf{G}}$. The tth iterate $\mathbf{G}^{(t)}$ can then be directly obtained by solving for the eigenvalues and eigenvectors of the linearized version $\tilde{\mathbf{G}}$.

For large ($M \gtrsim 2^L$) and infinite populations the dynamics of the fitness distribution is qualitatively very different from the behavior shown in figure 1: $\langle f \rangle$ increases smoothly and monotonically to an asymptote over a small number of generations. That is, there are no epochs. The reason is that for an infinite population, all genotypes are present in the initial population. Instead of the evolutionary dynamics *discovering* fitter strings over time, it essentially only expands the proportion of globally optimal strings already present in the initial population at $t = 0$. In spite of the qualitatively different dynamics for large populations, the (infinite population) operator $\mathbf{G}$ is the essential ingredient for describing the finite-population dynamics with its epochal dynamics as well, as we will now discuss.

5.3. *Finite-population sampling*

There are two important differences between the finite- and infinite-population dynamics. The first is that with finite populations the components P_n cannot take on continuous values between 0 and 1. Since the number of individuals with fitness n in the population is necessarily an integer, the values of P_n are quantized in multiples of $1/M$. Thus, the space of allowed finite-population fitness distributions turns into a regular lattice in $N + 1$ dimensions with a lattice spacing of $1/M$ within the simplex specified by the normalization Eq. (2).

Second, due to the sampling of members in the finite population, the dynamics of the fitness distribution is no longer deterministic. In general, we can only determine the conditional probabilities $\Pr[\vec{Q} \mid \vec{P}]$ that a given fitness distribution $\vec{P}$ leads to another $\vec{Q}$ in the next generation. These probabilities $\Pr[\vec{Q} \mid \vec{P}]$ are given by a multinomial distribution with mean $\mathbf{G}[\vec{P}]$:

$$\Pr[\vec{Q} \mid \vec{P}] = M! \prod_{n=1}^{N+1} \frac{(\mathbf{G}_n[\vec{P}])^{m_n}}{m_n!}, \tag{5}$$

where the components Q_i are multiples of $1/M : Q_i = m_i/M$, with integers $0 \leq m_i \leq M$.

Equation (5) can be most easily understood as follows. The population for the next generation is created by selecting, copying, and mutating M times in the same way from the current population $\vec{P}$. This implies that each of the M individuals in the next generation has equal and independent probabilities q_i to be of fitness i. These probabilities q_i also give the *expected* proportions q_i of individuals with fitness i in the next generation. The *actual* proportions Q_i of individuals with fitness i in the next generation are then given by a multinomial sample of size M from the distribution of expected proportions q_i. Since in the limit $M \rightarrow \infty$ of infinite populations we have that the expected proportions equal the actual proportions, we necessarily have that $q_i = \mathbf{G}_i[\vec{P}]$. In other words, the probabilities $\Pr[\vec{Q} \mid \vec{P}]$ are given by a multinomial sampling distribution of sample size M with mean $\mathbf{G}[\vec{P}]$; just as Eq. (5) expresses. In mathematical population genetics, such multinomial sampling Markov models are known as Wright-Fisher models (Hartl & Clark, 1989, pp. 66–70) and (Ewens, 1979).

Thus, for any finite-population fitness distribution $\vec{P}$ the (infinite population) operator $\mathbf{G}$ still gives the GA's *average* dynamics over one time step. Note that the components $\mathbf{G}_i[\vec{P}]$ 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 allowed lattice points in the finite-population state space. Since the variance around the expected distribution $\mathbf{G}[\vec{P}]$ is proportional to $1/M$, $\vec{Q}$ tends to be one of the lattice points close to $\mathbf{G}[\vec{P}]$. This finite-population dynamics is illustrated in figure 2.

5.4. Epochal dynamics

We will now discuss how the epochal behavior of the dynamics for a finite population comes about within the mathematical framework presented above.

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

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

Assuming that some component $\langle dP_i \rangle$ and its variance are 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

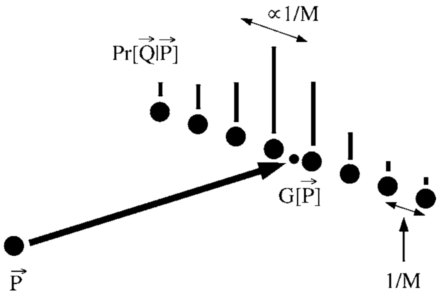


Figure 2. Illustration of the stochastic dynamics involved in going from one generation to the next with a finite population. The arrow points from the current distribution $\vec{P}$ to the expected next generation distribution $\mathbf{G}[\vec{P}]$. The large dots indicate points of the lattice of allowed finite-population fitness distributions. The columns above these lattice points indicate the size of $\Pr[\vec{Q} \mid \vec{P}]$. The width of this distribution is inversely proportional to the population size M. Note that the expected distribution $\mathbf{G}[\vec{P}]$ (small dot) does not occur at a lattice point.

spacing $(1/M)$ for the finite population, we expect the fitness distribution to not change in direction (fitness) i. In earlier work (van Nimwegen, Crutchfield, & Mitchell, 1997, 1999) we showed that, from a mathematical point of view, this is the mechanism by which finite populations cause epochal dynamics.

For the Royal Staircase fitness functions, we have that whenever fitness n is the highest in the population, such that $P_i = 0$ for all $i > n$, the rate at which higher-fitness strings are discovered, is very small. More precisely, for population sizes M that are not too large, the expected number of higher-fitness individuals that are created per generation is much smaller than 1, i.e. $\langle dP_i \rangle \ll 1/M$ for all $i > n$. A period of stasis (an evolutionary *epoch*) thus corresponds to the time the population spends before it discovers a string of fitness higher than n. More formally, each epoch n corresponds to the population being restricted to a region in the n-dimensional lower-fitness subspace consisting 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 the experimental runs of figure 1 illustrated, each epoch in the average fitness is associated with a (typically) 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 the entire fitness distribution $\vec{P}$ fluctuates in an approximately Gaussian way around some constant fitness distribution $\vec{P}^n$ during the epoch n—the generations when n is the highest fitness in the population.

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 $1 \leq f \leq n$, as shown in van Nimwegen, Crutchfield, and Mitchell (1999). 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, then we have:

$$\mathbf{G}^n[\vec{P}^n] = \vec{P}^n. \tag{6}$$

Intuitively, the operator $\mathbf{G}^n$ gives the average change in the fitness distribution *conditioned* on no strings of fitness higher than n being generated. The epoch distribution is then the fixed point of this operator. The uniqueness and construction of these fixed points will be discussed further below. By Eq. (3), then, the average fitness f_n in epoch n is given by:

$$f_n = \sum_{j=1}^{n} j P_j^n.$$

To summarize at this point, the statistical dynamics analysis is tantamount to the following qualitative picture. The global dynamics can be viewed as an incremental discovery of successively more (macroscopic) dimensions of the fitness distribution space. Initially, only strings of low fitness are present in the initial population. The population stabilizes on the epoch fitness distribution $\vec{P}^n$ corresponding to the best fitness n in the initial population. The fitness distribution fluctuates around the n-dimensional vector $\vec{P}^n$ until a string of fitness $n+1$ is discovered and spreads through the population. The population then settles into the $(n+1)$-dimensional 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 $\vec{P}^n$, unfolding a new macroscopic dimension of the fitness distribution space each time a higher-fitness string is discovered.

Whenever mutation creates a string of fitness $n+1$, this string may either disappear before it spreads, seen as the transient jumps in best fitness in figure 1, or it may spread, leading the population to fitness distribution $\vec{P}^{n+1}$. We call the latter process an *innovation*. Through an innovation, a new (macroscopic) dimension of fitness distribution space becomes stable.

Figure 1 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. Through a destabilization, a dimension can, so to speak, collapse. For some parameter settings, such as shown in figure 1(a) and (c), this is very rare. In these cases, 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 figure 1(b) and (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 is especially important in the current setting since, as we will show, the optimized GA often operates in this type of marginally stable parameter regime where later epochs destabilize quite easily.

6. Quasispecies distributions and epoch fitness levels

During epoch n the quasispecies fitness distribution $\vec{P}^n$ is given by a fixed point of the (projected) 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:

$$\mathbf{G}^n = \frac{1}{\langle f \rangle} \tilde{\mathbf{G}}^n, \tag{7}$$

where $\langle f \rangle$ is the average fitness of the fitness distribution that $\mathbf{G}^n$ acts upon; see Appendix A. The operator $\tilde{\mathbf{G}}^n$ is just an ordinary (linear) matrix operator and therefore, the fixed point equation (6) simply becomes an eigenvector equation. Since all components of $\tilde{\mathbf{G}}^n$ are positive, this fixed point is unique, from the positive matrix theorem of Perron (Gantmacher, 1959). The quasispecies fitness distribution $\vec{P}^n$ is given by the principal eigenvector of the matrix $\tilde{\mathbf{G}}^n$ (normalized in probability). From Eq. (7) it also follows that the principal eigenvalue f_n of $\tilde{\mathbf{G}}^n$ equals the average fitness of the quasispecies distribution. In this way, obtaining the quasispecies distribution $\vec{P}^n$ reduces to calculating the principal eigenvector of the matrix $\tilde{\mathbf{G}}^n$; see Appendix C.

The matrices $\tilde{\mathbf{G}}^n$ are generally of modest size: i.e., their dimension is smaller than the number of blocks N and substantially smaller than the dimension of genotype space. Due to this we can easily obtain numerical solutions for the epoch fitnesses f_n and the epoch quasispecies distributions $\vec{P}^n$. For a clearer understanding of the functional dependence of the epoch fitness distributions on the GA's parameters, however, App. C recounts analytical approximations to the epoch fitness levels f_n and quasispecies distributions $\vec{P}^n$.

The result is that the average fitness f_n in epoch n is

$$f_n = n(1-q)^{(n-1)K} \tag{8}$$

The epoch quasispecies is given by:

$$P_i^n = \frac{(1-\lambda)n\lambda^{n-1-i}}{n\lambda^{n-1-i}-i}\prod_{j=1}^{i-1}\frac{n\lambda^{n-j}-j}{n\lambda^{n-1-j}-j}, \tag{9}$$

where $\lambda = (1-q)^K$ is the probability that a block will undergo no mutations. For the following, we are actually interested in the most-fit quasispecies component P_n^n in epoch n. For this component, Eq. (9) reduces to

$$P_n^n = \lambda^{n-1}\prod_{j=1}^{n-1}\frac{f_n - f_j}{f_n - \lambda f_j}, \tag{10}$$

where we have expressed the result in terms of the epoch fitness levels $f_j = j\lambda^{j-1}$.

7. Mutation rate optimization

In the previous sections we argued that the GA's behavior can be viewed as stochastically hopping from one epoch to the next—when the search discovers a string with increased fitness that spreads in the population. Assuming that the total time to reach this global optimum is dominated by the time the GA spends in the epochs, we developed (van Nimwegen & Crutchfield, 2000) a way to tune the mutation rate q such that the time the GA spends in an epoch is minimized. We briefly review these results here before moving on to the more general theory that includes population-size effects and epoch destabilization.

To move from epoch n to epoch $n+1$, a string of fitness $n+1$ has to be discovered and spread through the population. During epoch n, the population is in a metastable state where it fluctuates around a constant fitness distribution $\vec{P}^n$. To a good approximation, we can assume that in each generation there is an equal and independent probability that epoch n will end by creating a fitness $n+1$ string that spreads through the population. Note that this immediately implies that the distribution of epoch times is geometric for each individual epoch.

The creation of a fitness $n+1$ string is most likely to occur through a string of fitness n mutating its nth block to the correct configuration of all 1s. Optimizing the mutation rate now amounts to finding a balance between two opposing effects of varying mutation rate. On the one hand, when the mutation rate is increased, the average number of mutations in the unaligned blocks goes up, thereby increasing the probability of creating newly aligned blocks. On the other hand, due to the increased number of deleterious mutations, the equilibrium proportions P_n^n of individuals in the highest fitness class during each epoch n decreases and so the number of individuals that are likely to discover a string of fitness $n+1$ decreases.

We previously (van Nimwegen & Crutchfield, 2000) 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. This is given by

$$C_{n+1} = MP_n^n P_a \pi_n(\lambda), \tag{11}$$

where $P_a = (1-\lambda)/(2^K - 1)$ is the probability of aligning a block (see Appendix B) and $\pi_n(\lambda)$ is the probability that a string of fitness $n+1$ will spread, as opposed to being lost through a fluctuation or a deleterious mutation. This spreading probability π_n can be calculated using a diffusion-equation approximation similar to the ones developed in population genetics by Kimura (1964). In van Nimwegen, Crutchfield, and Mitchell (1999) we showed how to adapt this diffusion-equation method to the present type of problem. We found that the spreading probability π_n largely depends on the relative average fitness difference of epoch $n+1$ over epoch n. Denoting this difference as

$$\gamma_n = \frac{f_{n+1} - f_n}{f_n} = \left(1 + \frac{1}{n}\right)\lambda - 1, \tag{12}$$

where we have used Eq. (8), one finds:

$$\pi_n(\lambda) = \frac{1 - \left(1 - \frac{1}{M}\right)^{2M\gamma_n + 1}}{1 - \left(1 - P_{n+1}^{n+1}\right)^{2M\gamma_n + 1}}. \tag{13}$$

If $P_{n+1}^{n+1} \gg 1/M$, this reduces to a population-size independent estimate of the spreading probability

$$\pi_n \approx 1 - e^{-2\gamma_n}. \tag{14}$$

If one were to allow for changing mutation rates between epochs, one would minimize the time spent in each epoch by maximizing C_{n+1} of Eq. (11) using Eqs. (10), (12), and (14). Note that C_{n+1} depends on q only through λ. The optimal mutation rate in each epoch n is determined by estimating the optimal value λ_o of λ for each n. We found that λ_o is well approximated (van Nimwegen & Crutchfield, 2000) by

$$\lambda_o(n) \approx 1 - \frac{1}{3n^{1.175}}.$$

For large n this gives the optimal mutation rate as

$$q_o \approx \frac{1}{3Kn^{1.175}}, n \gg 1. \tag{15}$$

Thus, the optimal mutation rate drops as a power-law in both n and K. This implies that if one is allowed to adapt the mutation rate during the run, the mutation rate should decrease as a GA run progresses so that the search will find the global optimum as quickly as possible.

We now turn to the simpler problem of optimizing mutation rate for the case of a *constant* mutation rate throughout a GA run. In van Nimwegen and Crutchfield (2000) we used Eq. (11) to 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 assumed that the GA visits all epochs, that the time spent in innovations between them is negligible, and that epochs are *always* stable. The epoch stability assumption entails that it is assumed to be 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 figure 1(a) and (c). They may hold even for the parameters of figure 1(b), but they most likely do not for figure 1(d). For the parameters of figure 1(d), we see that the later epochs ($n = 9$ and 10) easily destabilize a number of times before the global optimum is found. Although we will develop a generalization that addresses this more complicated behavior in the next sections, it is useful to work through the optimization of mutation rate under the stability assumption first.

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 (average) number E_n of fitness function evaluations in epoch n is given by MT_n. More explicitly, we have:

$$E_n = \left(P_n^n P_a \pi_n\right)^{-1}. \tag{16}$$

That is, the total number of fitness function evaluations in each epoch is independent of the population size M. This is due to two facts, given our approximations. First, the epoch lengths, measured in generations, are inversely proportional to M, while the number of fitness function evaluations per generation is M. Second, since for stable epochs $P_n^n \gg 1/M$, the probability π_n is also independent of population size M; recall Eq. (14).

The total number of fitness function evaluations $E(\lambda)$ to reach the global optimum is simply given by substituting into Eq. (16) our analytical expressions for P_n^n and π_n, Eqs. (10) and (14), respectively, and then summing $E_n(\lambda)$ over all epochs n from 1 to N. We then have:

$$E(\lambda) = \sum_{n=1}^{N} \frac{1}{P_a \pi_n(\lambda)} \prod_{i=1}^{n-1} \frac{n\lambda^{n-i-1} - i}{n\lambda^{n-i} - i}. \tag{17}$$

Note that in the above equation we set $\pi_N = 1$ by definition because the algorithm terminates as soon as a string of fitness $N + 1$ is found. That is, strings of fitness $N + 1$ need not spread through the population, they just need to be discovered once. The optimal mutation rate for an entire run is then obtained by minimizing Eq. (17) with respect to λ.

Figure 3 shows for $N = 4$ blocks of length $K = 10$ bits the dependence of the average total number $E(q)$ of fitness function evaluations on the mutation rate q. The dashed line is the theoretical prediction of Eq. (17); while the solid lines show the experimentally estimated values of $\langle E \rangle$ for four different population sizes. Each experimental data point is an estimate obtained from 250 GA runs. Figure 3 illustrates in a compact form our previous findings (van Nimwegen & Crutchfield, 2000), which can be summarized as follows.

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, which is accurately predicted by the theory.
2. The curve $E(q)$ is everywhere concave.

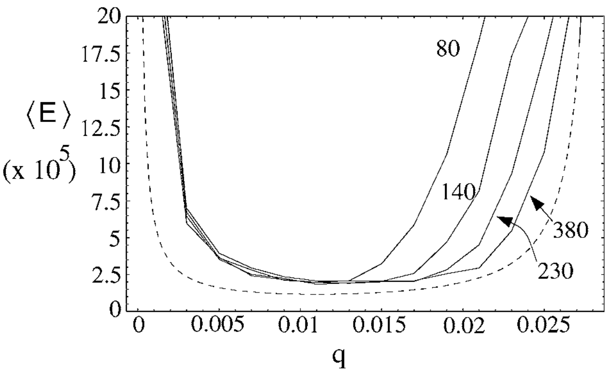


Figure 3. Average total number $\langle E\rangle$ of fitness function evaluations as a function of mutation rate q, from the theory (dashed), Eq. (17), and from experimental estimates (solid). The fitness function parameter settings are $N = 4$ blocks of length $K = 10$ bits. The mutation rate runs from $q = 0.001$ to $q = 0.028$. Experimental data points are estimates over 250 runs each. The experimental curves show four different population sizes; $M = 80$, $M = 140$, $M = 230$, and $M = 380$.

3. The theory slightly underestimates the experimentally obtained $\langle E\rangle$.
4. The optimal mutation rate q_o roughly occurs in the regime (between $q = 0.01$ and $q = 0.015$) where the highest epochs are marginally stable; see figure 1.
5. For mutation rates lower than q_o the experimentally estimated total number of fitness function evaluations $\langle E\rangle$ grows steadily and becomes almost independent of the population size M. (This is where the experimental curves in figure 3 overlap). For mutation rates larger than q_o the total number of fitness function evaluations does depend on M, which is not explained by the preceding theory.
6. There is a mutational error threshold in q that bounds the upper limit in q for which the GA can discover the optimum *at all*. Above the threshold, which is population-size independent, suboptimal strings of fitness N cannot stabilize in the population, even for very large population sizes. This error threshold is also correctly predicted by the theory. It occurs around $q_c = 0.028$ for $N = 4$ and $K = 10$.

Before embarking on the population-size dependent analysis, it is useful to make a few comments about the role of crossover if it had been included in our GA. As is explained in more detail in van Nimwegen and Crutchfield (2000), during epoch n *all* individuals in the population are relatively recent descendants from a string of fitness n. More precisely, strings with fitness $i < n$, have on average less than 1 offspring with fitness i. Therefore, lineages entirely consisting of suboptimal strings tend to be short lived. This, in turn, implies that all individuals are relatively recent descendants of a fitness n string. This means that all strings in the population share the genetic content of most of their blocks. That is, they differ only by a relatively small number of mutations. In particular, *all* strings in the population are unlikely to have their nth block aligned. This implies that it is almost impossible for a crossover event to create a string of fitness $n + 1$. Such an event can only occur when strings of fitness n are crossed over, the crossover point falls *within* the nth block, *and* the corresponding subblocks form a new aligned block. In general, the contribution of such beneficial events is marginal, especially taking into account the deleterious effects that crossover also produces, when high- and low-fitness parents combine to form two low-fitness offspring. Thus, the role of crossover during epochal evolution is marginal.

One should note that these observations extend to fitness functions such as Royal Road fitness functions—see the discussion in van Nimwegen, Crutchfield, and Mitchell 1999—that are explicitly constructed such that crossover may recombine structurally different genotypes of fitness n to create genotypes with fitness larger than n. Each epoch n is founded by a *single* individual of fitness n. All individuals of fitness n that occur in the population during epoch n are therefore descendants of a single genotype of fitness n. It is thus very unlikely, in epochal evolution, that structurally different fitness-n individuals occur side by side in the population. This in turn implies that "recombining building blocks" (Holland, 1975) is unlikely to play a major role in epochal evolution.

8. Epoch destabilization: Population-size dependence

We now extend the above analysis to account for E's dependence on population size. This not only improves the parameter-optimization theory, but also leads us to consider a number of issues and mechanisms that shed additional light on how GAs work near their optimal parameter settings. Since it appears that optimal parameter settings often lead the GA to run in a behavioral regime were the population dynamics is marginally stable in the higher epochs, we consider how epoch destabilization dynamics affects the time to discover the global optimum.

We saw in figure 1(b) and (d) that, around the optimal parameter settings, the best fitness in the population can show intermittent behavior. Apparently, fluctuations sometimes cause an epoch's current best strings (of fitness n) in the population to disappear. The best fitness then drops to $n - 1$. Often, strings of fitness n are rediscovered later on. What happens is that for these higher epochs, the fluctuations in the proportion P_n of strings with fitness n becomes comparable to the average P_n^n of this proportion. That is, a fluctuation may bring P_n very close to 0, and so all strings of fitness n may be lost. As we will see, the probability of a destabilization is sensitive to the population size M, introducing population-size dependence in the average total number $\langle E \rangle$ of fitness function evaluations.

As we just noted, the theory for $E(q)$ used in van Nimwegen and Crutchfield (2000) assumed that destabilizations do not occur, leading to a population-size independent theory. However, as is clear from figure 1(d), it is possible that epoch n destabilizes several times to epoch $n - 1$ before the population moves to epoch $n + 1$, and this will considerably alter E. For example, if during epoch n the population is 3 times as likely to destabilize to epoch $n - 1$ compared to innovating to epoch $n + 1$, then we expect epoch n to destabilize three times on average before moving to epoch $n + 1$. Assuming that epoch $n - 1$ is stable, this means that epoch n has to be rediscovered 3 times on average before epoch $n + 1$ is discovered. This will effectively increase the time spent in epoch n by 3 times the average number of generations spent in epoch $n - 1$.

To make these ideas precise we introduce a Markov chain model to describe the "hopping" up and down between the epochs. The Markov chain has $N + 1$ states, each representing an epoch. In every generation there are probabilities p_n^+ to innovate from epoch n to epoch $n+1$ and p_n^- to destabilize, falling from epoch n to epoch $n-1$. The globally optimum state $N+1$ is an absorbing state. Starting from epoch 1 we calculate the expected number T of generations for the population to reach the absorbing state for the first time.

The innovation probabilities p_n^+ to move from epoch n to $n+1$ are just given by the C_{n+1} of Eq. (11):

$$p_n^+ = C_{n+1} = \frac{M}{E_n},$$

where E_n is given by the approximation of Eq. (16). Note that when MP_n^n approaches 1 the spreading probability π_n, as given by Eq. (13), becomes population-size dependent as well, and we use Eq. (13) rather than Eq. (14). To obtain the destabilization probabilities p_n^- we assume that in each generation the population has an equal and independent probability to destabilize to epoch $n-1$. This probability is given by the inverse of the average time until a destabilization occurs.

To calculate the average time D_n that the population spends in epoch n before it destabilizes we have to analyze the dynamics of fluctuations in the proportion P_n of individuals with fitness n. This can be done most easily using a diffusion-equation approximation. (See Kimura (1964) for an introduction to using diffusion equations in these settings). Below we sketch this diffusion-equation analysis, which is described in more detail in van Nimwegen, Crutchfield, and Mitchell (1999).

We introduce the deviation $x(t)$ from the mean proportion P_n^n of individuals with fitness n by defining $P_n(t) = P_n^n + x(t)$. We then calculate the expected change $\langle \delta x \rangle = \langle x(t+1) \rangle - x(t)$ and the second moment $\langle (\delta x)^2 \rangle$ of the expected change in x. The dynamics of the probability distribution $\Pr(x, t)$ that the deviation will be x at time t is then approximated by the Fokker-Planck equation:

$$\frac{\partial}{\partial t}\Pr(x,t) = -\frac{\partial}{\partial x}\langle \delta x \rangle \Pr(x,t) + \frac{1}{2}\frac{\partial^2}{\partial x^2}\langle (\delta x)^2 \rangle \Pr(x,t). \tag{18}$$

Once we have expressions for $\langle \delta x \rangle$ and $\langle (\delta x)^2 \rangle$ we can use this diffusion equation to solve for such statistics as the variance $\mathrm{Var}(P_n)$ of the proportion of highest fitness individuals during epoch n and the average time D_n for P_n to reach values smaller than $1/M$—the time at which all strings of fitness n are lost. See, for instance, Gardiner (1985) for the details of such calculations.

For small x, the values $\langle (\delta x)^2 \rangle$ can be approximated by the value of $\langle (\delta x)^2 \rangle$ when x is *zero*. When $x = 0$, we have that $P_n = P_n^n$, and the *variance* in P_n over one generation is simply given by the variance due to the multinomial sampling of Eq. (5):

$$\langle (\delta x)^2 \rangle = \frac{P_n^n\left(1 - P_n^n\right)}{M}.$$

This is the approximation we will use for $\langle (\delta x)^2 \rangle$ in Eq. (18).

The calculation of $\langle \delta x \rangle$ is somewhat more involved and will not be presented here in detail because of space constraints. The reader is referred to van Nimwegen, Crutchfield, and Mitchell (1999) for the details of this calculation. In general we find the intuitive result that the deviation x is expected to be scaled down by a constant factor each generation:

$$\langle \delta x \rangle = -\mu_n x.$$

The scale factor μ_n is a function of the epoch distributions $\vec{P}^i$ and epoch fitnesses f_i. Said simply, the fluctuations in P_n are the result of fluctuations in the directions of all lower lying epochs $\vec{P}^i$ with $i < n$. The fluctuations in the direction of epoch i are scaled down at an average rate $(f_n - f_i)/f_n$ per generation. Thus, fluctuations in the direction of low epochs are scaled down most rapidly. The quantity μ_n is then a weighted sum:

$$\mu_n = \frac{\sum_{i=1}^{n-1}(f_n - f_i)B_i}{f_n \sum_{i=1}^{n-1} B_i},$$

where B_i is a measure of how much of the multinomial sampling fluctuations occur in the direction of epoch i. The expected fluctuations in components i and j due to multinomial sampling are given by

$$\langle dP_i dP_j \rangle = \frac{P_i^n \left(\delta_{ij} - P_j^n\right)}{M},$$

when $x = 0$. We calculate the weights B_i by calculating the overlap of these fluctuations with the epoch distributions $\vec{P}^i$ for each $i < n$. These can be calculated by introducing a matrix $\mathbf{R}$ that contains the epoch distributions in its columns:

$$\mathbf{R}_{ij} = P_i^j,$$

The overlaps B_i are then calculated using the inverse of $\mathbf{R}$:

$$B_i = \frac{1}{M} \sum_{k,m=1}^{n-1} \mathbf{R}_{ik}^{-1} \mathbf{R}_{im}^{-1} P_k^n \left(\delta_{km} - \vec{P}_m^n\right).$$

Generally, μ_n decreases monotonically as a function of n since fluctuations in the proportion P_n^n of individuals in the highest-fitness class n decay more slowly for higher epochs.

Continuing at this level of summary, the variance $\text{Var}(P_n)$ is simply given by $\text{Var}(P_n) = P_n^n(1 - P_n^n)/(M\mu_n)$, and the average time until destabilization is approximately given by

$$D_n = \frac{MP_n^n}{1 - P_n^n} + \frac{\pi}{2\mu_n} \text{erfi}\left[\sqrt{\frac{M\mu_n P_n^n}{1 - P_n^n}}\right] \text{erf}\left[\sqrt{\frac{M\mu_n (1 - P_n^n)}{P_n^n}}\right], \tag{19}$$

where $\text{erf}(x)$ is the error function and $\text{erfi}(x) = \text{erf}(ix)/i$ is the imaginary error function.

Notice that the argument of erfi(x), $\sqrt{M\mu_n P_n^n/(1-P_n^n)}$, is the ratio between the mean proportion P_n^n and standard deviation of the number of individuals with fitness n. The function $\operatorname{erfi}(x)$ is a very rapidly growing function of its argument: $\operatorname{erfi}(x) \approx \exp(x^2)/x$ for x larger than 1. Therefore, $\sqrt{M\mu_n P_n^n/(1-P_n^n)}$ being either smaller (larger) than 1 is a reasonable criterion for the instability (stability) of an epoch. When the standard deviation of P_n is (much) smaller than the mean P_n^n, epochs are stable for a very long time; while they become unstable very quickly as the ratio of the standard deviation and the mean approaches 1.

The above formula Eq. (19) is analogous to error thresholds in the theory of molecular evolution. Generally, error thresholds denote the boundary in parameter space between a regime where a certain high fitness string, or an equivalence class of high-fitness strings, is stable in the population and a regime where it is unstable. In the case of a single high-fitness *master sequence* one speaks of a genotypic error threshold, see Alves and Fontanari (1998), Eigen, McCaskill, and Schuster (1989), Nowak and Schuster (1989), Swetina and Schuster (1982). In the case of an equivalence class of high-fitness strings, one speaks of a *phenotypic* error threshold, see Huynen, Stadler, and Fontana (1996), Reidys, Forst, and Schuster (2001).

A sharply defined error threshold generally only occurs in the limit of infinite populations and infinite string length (Leuthäusser, 1987), but extensions to finite population cases have been studied by Alves and Fontanari (1998), Nowak and Schuster (1989), and Reidys, Forst, and Schuster (2001). In Reidys, Forst, and Schuster (2001), for example, the occurrence of a finite-population phenotypic error threshold was defined by the equality of the standard deviation and the mean of the number of individuals of the highest-fitness class. This definition is in accord with Eq. (19), as we explained above.

The average time until destabilization is thus given by D_n of Eq. (19), and so the average probability per generation for a destabilization to occur is simply its inverse:

$$p_n^- = \frac{1}{D_n}.$$

Finally, note that the probability to remain in epoch n is $1 - p_n^+ - p_n^-$.

We now have expressions for all of the Markov chain's transition probabilities. With these it is straightforward to calculate the average number T of generations before the GA discovers the global optimum for the first time. This is done by calculating the average time for the Markov chain to reach its absorbing state, starting from epoch 1. Following, for instance, Section 7.4 of Gardiner (1985) the result is

$$T = \sum_{n=1}^{N} \phi_n \sum_{k=1}^{n} \frac{1}{p_k^+ \phi_k}, \tag{20}$$

where ϕ_n is defined as:

$$\phi_n = \prod_{k=2}^{n} \frac{p_k^-}{p_k^+}, \quad n \geq 2,$$

and

$$\phi_1 = 1.$$

Since Eq. (20) gives the average number T of generations, the average number of fitness function evaluations $E(q, M)$ is given by:

$$\begin{aligned} E(q, M) &= MT \\ &= E_N + E_{N-1}\left(1 + \frac{E_N}{MD_N}\right) \\ &\quad + E_{N-2}\left(1 + \frac{E_{N-1}}{MD_{N-1}}\left(1 + \frac{E_N}{MD_N}\right)\right) + \dots, \end{aligned} \tag{21}$$

where the E_n are given by Eq. (16) and where the last equality is obtained by rewriting the sums in Eq. (20). As epochs become arbitrarily stable ($D_n \to \infty$), the terms with D_n in the denominator go to zero, and Eq. (21) reduces to Eq. (17), as it should.

9. Theory versus experiment

We can now compare this population-size dependent approximation, Eq. (21), with the experimentally measured dependence on M of the average total number $\langle E \rangle$ of fitness function evaluations. Figure 4 shows the dependence of $\langle E \rangle$ on the population size M for two different parameter settings of N and K and for a set of mutation rates q.

The upper figures, figure 4(a) and (c), give the dependence of the experimentally estimated $\langle E \rangle$ on the population size M. The lower figures, figure 4(b) and (d), give the theoretical predictions from Eq. (21). The upper left figure, figure 4(a), shows $\langle E \rangle$ as a function of M for $N = 4$ blocks of length $K = 10$ for four different mutation rates: $q \in \{0.013, 0.015, 0.017, 0.019\}$. The population size ranges from $M = 50$ to $M = 320$. The total number of fitness function evaluations on the vertical axis ranges from $\langle E \rangle = 0$ to $\langle E \rangle = 15 \times 10^5$. Each data point was obtained as an average over 250 GA runs. Figure 4(b) shows the theoretical predictions for the same parameter settings. Figure 4(c) gives the experimental estimates for $N = 6$ blocks of length $K = 6$, over the range $M = 30$ to $M = 300$, for four mutation rates: $q \in \{0.018, 0.02, 0.022, 0.024\}$. The total number of fitness function evaluations on the vertical axis range from $\langle E \rangle = 0$ to $\langle E \rangle = 7 \times 10^5$. Figure 4(d) shows the theoretical predictions for the same range of M and the same four mutation rates.

We see that as the population size becomes "too small" destabilizations make the total number of fitness function evaluations increase rapidly. The higher the mutation rate, the higher the population size at which the sharp increase in $\langle E \rangle$ occurs. These qualitative effects are captured accurately by the theoretical predictions from Eq. (21). Although our analysis involves several approximations (e.g. as in the calculations of D_n), the theory does quantitatively capture the population-size dependence well, both with respect to the predicted absolute number of fitness function evaluations and the shape of the curves as a function of M for the different mutation rates. From figure 4(c) and (d) it seems that the theory overestimates the growth of $\langle E \rangle$ for the larger mutation rates as the population size decreases. Still, the theory correctly captures the sharp increase of $\langle E \rangle$ around a population size of $M = 50$.

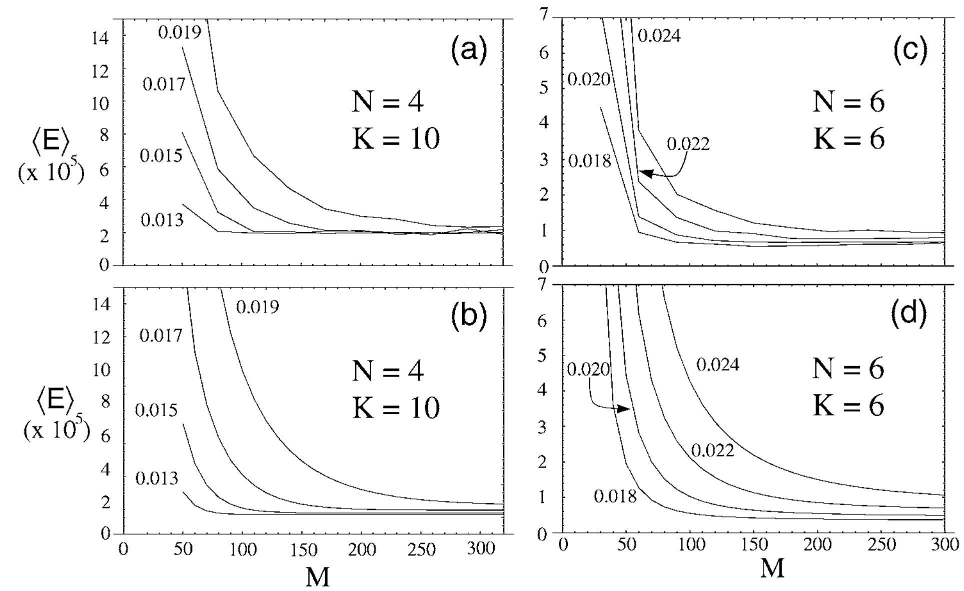


Figure 4. Average total number $\langle E \rangle$ of fitness function evaluations as a function of the population size M for two different fitness function parameters and four mutation rates each, both experimentally (figure (a) and (c), top row) and theoretically (figure (b) and (d), bottom row). In each figure each solid line gives $E(M)$ for a different mutation rate. Each experimental data point is an average over 250 GA runs. Figures (a) and (b) have $N = 4$ blocks of length $K = 10$. The upper figure (a) shows the experimentally estimated $E(M)$ as a function of M for the mutation rates $q \in \{0.013, 0.015, 0.017, 0.019\}$. The lower figure (b) shows the theoretical results, as given by Eq. (21), for the same parameter settings. In both, the population size ranges from $M = 50$ to $M = 320$ on the horizontal axis. Figures (c) and (d) have $N = 6$ blocks of length $K = 6$. Figure (c) shows the experimental averages and figure (d) the theoretical predictions for the same parameter settings. The population sizes on the horizontal axis run from $M = 30$ to $M = 300$. The mutation rates shown in (c) and (d) are $q \in \{0.018, 0.02, 0.022, 0.024\}$.

As the population size increases beyond approximately $M = 200$, we find experimentally that the average total number of fitness function evaluations $\langle E \rangle$ starts rising very slowly as a function of M. This effect is not captured by our analysis. It is also barely discernible in figure 4(a) and (c). We believe that the slow increase of $\langle E \rangle$ for large population sizes derives from two sources.

First, by the maximum entropy assumption, our theory assumes that all individuals in the highest fitness class are genetically *independent*, apart from the sharing of their aligned blocks. Under that assumption, the average number of fitness function evaluations to discover a string of fitness $n + 1$ in epoch n is independent of M. A population of size $2M$ is assumed to take half as many generations to discover a higher-fitness string as a population of size M. This is not true in general. The sampling during the selection process introduces genetic correlations in the individuals of the highest fitness class. Due to these correlations, the MP_n^n strings of fitness n are not searching for a higher fitness string *independently* and therefore the probability (per generation) to discover a higher-fitness string grows somewhat slower than linearly with M. Since the number of fitness function evaluations *does* grow linearly with M, the correlation effect leads to a slow growth of $\langle E \rangle$ with M. Unfortunately, this effect is very hard to address analytically and quantitatively.

The second reason for the increase of E with increasing population size comes from the time the population spends in the short innovations between the different epochs. During the innovation, the single mutant of fitness $n+1$ amplifies in the population until its epoch equilibrium value P_{n+1}^{n+1} is reached. Up to now, we have neglected these innovation periods. Generally, they only contribute marginally to E. We previously (van Nimwegen, Crutchfield, & Mitchell, 1999) calculated the approximate number g_n of generations that the population spends in the innovation from epoch n to epoch $n+1$ and found that:

$$g_n = \frac{2+\gamma_n}{\gamma_n} \log[M],$$

where γ_n is the fitness differential given by Eq. (12). That is, the number of generations taken is proportional to the logarithm of the population size and grows with decreasing fitness differentials γ_n. The GA expends a total of

$$I = M \log[M] \sum_{n=1}^{N-1} \frac{2+\gamma_n}{\gamma_n}, \tag{22}$$

fitness function evaluations in the innovations. Notice that this number grows as $M \log[M]$. This is the second source for the increase of E with M. Since the terms in the above sum are generally much smaller than E_n, the contribution of I only leads to a slow increase.

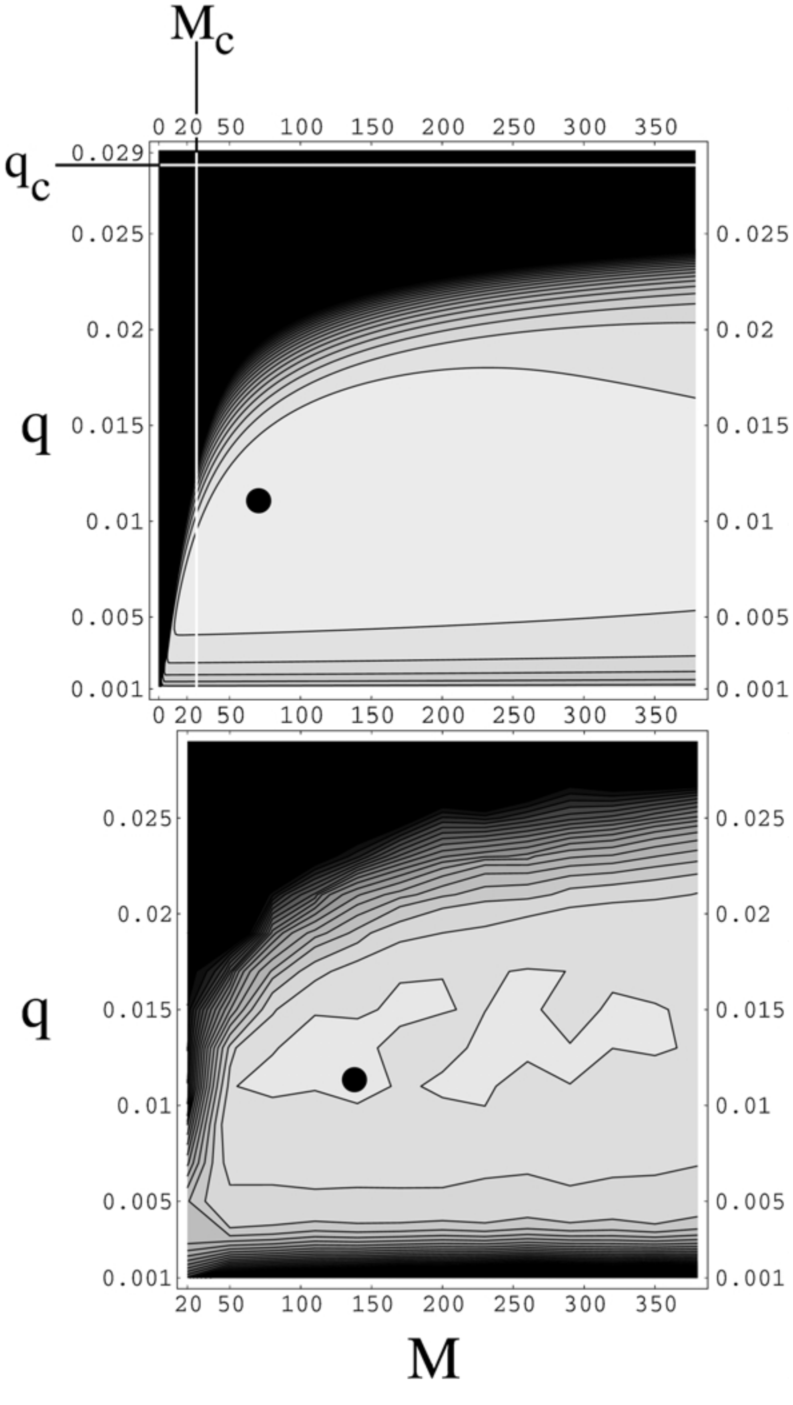
10. Search-effort surface and generalized error thresholds

We summarize our theoretical and experimental findings for the entire *search-effort surface* $E(q, M)$ of the average total number of fitness function evaluations in figure 5.

The figure shows the average total number $E(q, M)$ of fitness function evaluations for $N=4$ blocks of length $K=10$ bits; the same fitness function as used in figures 1(c), 3 and 4(a), and (b). The top plot shows the theoretical predictions, which now include the innovation-time correction from Eq. (22); the bottom, the experimental estimates. The

→

Figure 5. Contour plots of the search-effort surface $E(q, M)$ of the average total number of fitness function evaluations for the theory (upper), Eqs. (21) and (22), and for experimental estimates (lower). The parameter settings are $N = 4$ blocks of length $K=10$ bits. The population size M runs from $M = 1$ to $M=380$ on the horizontal axis on the upper plot and from $M=20$ to $M=380$ on the lower. The mutation rate runs from $q=0.001$ to $q=0.029$ on the vertical. The contours are plotted over the range $E(q, M)=0$ to $E(q, M)=2\times 10^6$ with a contour at each multiple of 10^5. The experimental surface was interpolated from 195 equally spaced data points, 13 increments of $\Delta M=30$ on the horizontal axis by 15 increments of $\Delta q=0.002$ on the vertical. The theoretical surface was interpolated over a grid using $\Delta M=1$ and $\Delta q=0.00025$. The optimal theoretical parameter setting, $(q_o, M_o)=(0.011, 60)$, and the estimated optimal experimental parameter setting, $(q_o, M_o)=(0.011, 140)$, are marked in their respective plots with a dot.


M_c
q_c
0 20 50 100 150 200 250 300 350
0.029
0.025
0.02
0.015
0.01
0.005
0.001
q
20 50 100 150 200 250 300 350
0.025
0.02
0.015
0.01
0.005
0.001
q
M

horizontal axis ranges from a population size of $M = 1$ ($M = 20$, experimental) to a population size of $M = 380$ with steps of $\Delta M = 1$ ($\Delta M = 30$, experimental). The vertical axis runs from a mutation rate of $q = 0.001$ to $q = 0.029$ with steps of $\Delta q = 0.00025$ in the theoretical plot and $\Delta q = 0.002$ in the experimental. The experimental search-effort surface is thus an interpolation between 195 data points on an equally spaced lattice of parameter settings. Each experimental data point is an average over 250 GA runs. The contours range from $E(q, M) = 0$ to $E(q, M) = 2 \times 10^6$ with each contour representing a multiple of 10^5. Note that the lowest values of E lie between 10^5 and 2×10^5. Lighter gray scale corresponds to smaller values of $E(q, M)$.

The first observations that can be made from figure 5 were already implied in the data from figures 3 and 4. First, the theory correctly predicts the relatively large region in parameter space where the GA searches most efficiently. Second, the theory predicts the location of the optimal parameter settings, indicated by the dot in the upper plot of figure 5 approximately correctly. The optimum occurs for somewhat higher population size in the experiments, as indicated by the dot in the lower plot of figure 5. Due to the large variance in E from run to run (recall Table 1) and the rather small differences in the experimental values of $\langle E \rangle$ near this regime, however, it is hard to infer from the experimental data exactly where the optimal population size occurs. Third, the theory underestimates the absolute magnitude of $E(q, M)$ somewhat. Fourth, at small mutation rates the theory underestimates the increase of $E(q, M)$ for decreasing q (moving down vertically in figure 5). Apart from this, though, the plots illustrate the general shape of the search-effort surface $E(q, M)$ and indicate that the theory accurately captures this shape.

There is a relatively large area of parameter space around the optimal setting (q_o, M_o) for which the GA runs efficiently. Moving away from this optimal setting horizontally (changing M) increases $E(q, M)$ only slowly at first. For decreasing M one reaches a "wall" relatively quickly around $M = 30$. For population sizes lower than $M = 30$, the higher epochs become so dynamically unstable that it is difficult for the population to reach the global optimum string *at all*. In contrast, moving in the opposite direction, increasing population size, $E(q, M)$ increases slowly over a relatively large range of M. Thus, choosing the population size too small is far more deleterious than setting it too large.

Moving away from the optimal setting vertically (changing q) the increase of $E(q, M)$ is also slow at first. Eventually, as the plots make clear, increasing q one reaches the same "wall" as encountered in lowering M. This occurs at $q \approx 0.02$ in figure 5. For larger mutation rates the higher epochs become too unstable in this case as well, and the population is barely able to reach the global optimum.

The wall in (q, M)-space is the two-dimensional analogue of a phenomenon known as the *error threshold* in the theory of molecular evolution. As pointed out in Section 8, in our case, error thresholds form the boundary between parameter regimes where epochs are stable and unstable. Here, the *error boundary* delimits a regime in parameter space where the optimum is discovered relatively quickly from a regime, in the black upper-left corners of the plots, where the population essentially never finds the optimum. For too-high mutation rates or too-low population sizes, selection is not strong enough to maintain high fitness strings—in our case those close to the global optimum—in the population against sampling fluctuations and deleterious mutations. Strings of fitness N will not stabilize in the population but will almost always be immediately lost, making the discovery of the global optimum string of fitness $N + 1$ extremely unlikely.

Note that the error boundary rolls over with increasing M in the upper-left corner of the plots. It bends all the way over to the right, eventually running horizontally, thereby determining a population-size *independent* error threshold. For our parameter settings this occurs around $q \approx 0.028$. Thus, beyond a critical mutation rate of $q_c \approx 0.028$ the population almost never discovers the global optimum, even for very large populations.

The value of this horizontal asymptote q_c can be roughly approximated by calculating for which mutation rate q_c the spreading probability $\pi(N - 1)$ goes to 0—i.e., find q_c such that $f_N \approx f_{N-1}$. For those parameters, strings of fitness N will generally not spread in the population and the population is thus under no selective pressure to move from epoch $N - 1$ to epoch N. Using our analytic approximations, we find that the critical mutation rate q_c is simply given by:

$$q_c = 1 - \sqrt[\kappa]{\frac{N-1}{N}}.$$

For the parameters of figure 5 this gives $q_c = 0.0284$. This asymptote is indicated there by the horizontal line in the top plot.

Similarly, below a critical population size M_c, it is also practically impossible to reach the global optimum, even for low mutation rates. This M_c can also be roughly approximated by calculating the population size for which the sampling noise is equal to the fitness differential between the last two epochs. Formally, this occurs when $1/\sqrt{M}$ becomes equal to γ_{N-1}. We then find:

$$M_c = \left(\frac{N-1}{N\lambda - N + 1}\right)^2.$$

For the parameters of figure 5 this gives $M_c \approx 27$ around (the optimum) $q = 0.011$. This threshold estimate is indicated by the vertical line in figure 5.

Further, notice that for small mutation rates, at the bottom of each plot in figure 5, the contours run almost horizontally. That is, for small mutation rates relative to the optimum mutation rate q_o, the total number $E(q, M)$ of fitness function evaluations is insensitive to the population size M. Decreasing the mutation rate too far below the optimum rate increases $E(q, M)$ quite rapidly. According to our theoretical predictions it increases roughly as $1/q$ with decreasing q. The experimental data indicate that this is a slight underestimation. In fact, $E(q, M)$ appears to increase as $1/q^\alpha$ where the exponent α lies somewhere between 1 and 2.

Globally, the theoretical analysis and empirical evidence indicate that the search-effort surface $E(q, M)$ is everywhere concave. That is, for any two points (q_1, M_1) and (q_2, M_2), the straight line connecting these two points is everywhere above the surface $E(q, M)$. We believe that this is always the case for mutation-only genetic algorithms with a static fitness function that has a unique global optimum. This feature is useful in the sense that a steepest descent algorithm on the level of the GA *parameters* q and M will always lead to the unique optimum (q_o, M_o).

Finally, it is important to emphasize once more that there are large run-to-run fluctuations in the total number of fitness evaluations to reach the global optimum. (Recall Table 1). Theoretically, each epoch has a geometrically distributed length since there is an equal and independent innovation probability of leaving it at each generation. The standard deviation of an exponential distribution is equal to its mean. Since the total time $E(q, M)$ is dominated by the last epochs, the total time $E(q, M)$ has a standard deviation close to its mean.

One conclusion from this is that, if one is only going to use a GA for a few runs on a specific problem, there is a large range in parameter space for which the GA's performance is statistically equivalent. In this sense, fluctuations lead to a large "sweet spot" of GA parameters. On the other hand, these large fluctuations reflect the fact that individual GA runs do not reliably discover the global optimum within a fixed number of fitness function evaluations. Notice that this is not a feature of the parameter settings or the analysis that we perform but a feature of the GA dynamics itself.

11. Conclusions

We derived explicit analytical approximations to the total number of fitness function evaluations that a GA takes on average to discover the global optimum as a function of both mutation rate and population size. The class of fitness functions so analyzed describes a general subbasin-portal architecture in genotype space. The GA's dynamics on this class of fitness functions consists of alternating periods of stasis (epochs) in the fitness distribution of the population, with short bursts of change (innovations) to higher average fitness. During the epochs the most-fit individuals in the population diffuse over neutral networks of iso-fitness strings until a portal to a network of higher fitness is discovered. Then descendants of this higher-fitness string spread through the population.

The time to discover these portals depends both on the fraction of the population that is located on the highest-fitness neutral net in equilibrium and the speed at which these population members diffuse over the network. Although increasing the mutation rate increases the diffusion rate of individuals on the highest neutral network, it also increases the rate of deleterious mutations that cause these members to "fall off" the highest-fitness network. The mutation rate is optimized when these two effects are balanced so as to maximize the total amount of explored volume on the neutral network per generation. The optimal mutation rate, as given by Eq. (15), is dependent on the neutrality degree (the local branching rate) of the highest-fitness network and on the fitnesses of the lower lying neutral networks onto which the mutants are likely to fall.

With respect to optimizing population size, we found that the optimal population size occurs when the highest epochs are just barely stable. That is, given the optimal mutation rate, the population size should be tuned such that only a few individuals are located on the highest-fitness neutral network. The population size should be large enough such that it is relatively unlikely that all the individuals disappear through a deleterious fluctuation, but not much larger than that. In particular, if the population is much larger, so that many individuals are located on the highest-fitness network, then the sampling dynamics causes these individuals to correlate genetically. Due to this genetic correlation, individuals on the highest-fitness network do not independently explore the neutral network. This leads, in turn, to a deterioration of the search algorithm's efficiency. Therefore, the population size should be as low as possible without completely destabilizing the last epochs. Given this, one cannot help but wonder how general the association of efficient search and marginal stability is.

11.1. Genetic algorithms versus hill climbers

It would appear that the GA wastes computational resources when maintaining a population quasispecies that contains many suboptimal fitness members; that is, those that are not likely to discover higher-fitness strings. This is precisely the reason that the GA performs so much more poorly than a simple hill climbing algorithm on this particular set of fitness functions, as originally reported in Mitchell, Holland, and Forrest (1994b). To be more specific, let's compare the GA at its optimal parameter settings with a Random Mutation Hill Climber, which performs a random bit flip at each time step and accepts this change when no lowering of fitness occurs. When the fitness of the string is n, mutations in blocks 1 through $n-1$ are always deleterious, and mutations in blocks $n+1$ through N are always neutral. Only 1 in every N mutations occurs in block n. Roughly 2^K mutations have to occur in a block before it is aligned for the first time. Thus, aligning block n takes the random mutation hill climber roughly $N2^K$ time steps on average, independent of n. In other words, the hill climber spends $N2^K$ time steps on average in each "epoch". Since N blocks have to be aligned in total, the random mutation hill climber takes roughly $N^2 2^K$ time steps to reach the global optimum.

In contrast, by numerically determining the optimal parameter settings from Eq. (21) we find that at its optimal parameter settings, the GA takes approximately $E \approx 2.2N^{3.1}2^K$ fitness function evaluations to reach the global optimum. That is, roughly a factor N more than the random mutation hill climber. This factor N is the result of the many suboptimal fitness individuals that are maintained in the population. The deleterious mutations together with the nature of the selection mechanism drive up the fraction of lower-fitness individuals in the quasispecies, and this fraction of the population plays no role in the search for higher-fitness strings.

If we allowed ourselves to tune the selection strength, we could have tuned selection so high that only the most-fit individuals would ever be selected. In this "infinite selection" limit, we would have *only* strings of fitness n being selected during epoch n. It is easy to calculate the optimal parameter settings for this regime. With $\lambda = (1-q)^K$, we have that the probability P_{disc} for a fitness n string to turn into a fitness $n+1$ string is $P_{\text{disc}} \approx \lambda^{n-1}(1-\lambda)/2^K$. This probability is maximal for $\lambda = (1 - 1/n)$. Using this as optimal parameter settings, the average number of fitness function evaluations during epoch n becomes

$$E_n \approx n2^K \left(1 - \frac{1}{n}\right)^{1-n}.$$

For n not too small, the last factor is roughly equal to e. We can then sum over n from 1 to N and have for the total time $E \approx eN(N+1)2^K/2$. In short, in the limit of infinite selection strength, both hill climbing algorithms and the GA have a scaling of the total number of fitness function evaluations E as $\mathcal{O}(N^2 2^K)$.

11.2. Coarse graining the fitness function

For the fitness functions we analyzed, setting the selection strength infinitely high thus turns out to be the best strategy. Of course, this is largely the result of the fact that none of the fitness functions in our set has local optima. However, it is a common belief that fitness functions typical of combinatorial optimization problems possess many local optima. In general, tuning the selection strength infinitely high causes the population to become "pinned" on the tiniest of local peaks. Thus, this cannot be a good general strategy. With this in mind, let us step back from our detailed analysis and consider the broader implications of our results.

We believe that the results point to an interesting interplay between *neutrality*, *local optima*, and *marginal stability*—an interplay that is potentially quite general. Neutrality refers to the phenomenon that, for any genotype, there are always some single-mutant neighbors that have the same fitness. When neutrality is present, a population does not become pinned to any particular point or island in genotype space, but instead has the possibility of diffusing through genotype space. In the Royal Staircase functions used here, this neutrality was explicitly built in from the start. Epochs corresponded to times during which the population diffused over the current highest-fitness network in search of a connection to higher-fitness networks.

We found that the GA searches most efficiently when population size and mutation rate are set such that these epochs are *marginally stable*. That is, the GA dynamics is as "stochastic" as possible without destabilizing the current and later epochs. Strings of the current highest fitness are (only slightly) preferentially reproduced over strings with lower fitness, and the population size is just large enough to protect these highest-fitness strings from disappearing through deleterious sampling fluctuations. Thus, for fitness functions consisting of interwoven neutral networks, our analysis shows that *marginal stability* of the highest-fitness strings corresponds to optimal search.

The role of marginal stability in more general cases, involving fitness functions with many local optima, can be best understood by perturbing away from our Royal Staircase class. Assume, for instance, that we add small fitness fluctuations to the fitnesses of each genotype. That is, a genotype that previously had fitness n, now receives fitness $n + \epsilon$ where ϵ is some small (random) number. These fluctuations are likely to induce many local optima in the fitness function. The random mutation hill climber will easily become pinned on this type of "landscape". However, the GA, in the regime of optimal parameter settings, will hardly be affected in its search dynamics. The reason for this is that selection simply does not "notice" these small fitness differences. In the optimal parameter regime, strings of fitness n are barely distinguished from strings with fitness $n - 1$, let alone from strings of fitness $n + \epsilon$. That is, fitness differentials between strings have to be above some minimal size to be "noticed" by the dynamics.

This intuitive idea, which is closely related to the nearly neutral theories of molecular evolution (Ohta, 1973; Ohta & Gillespie, 1996), can be made more precise. Let us assume that the current most-fit strings in the population have a fitness f and that strings with fitness f typically have d *defining bits*—i.e., these bits are deleterious when changed. Under the assumption that mutations from low- to high-fitness strings are negligible, this leads to an average fitness of $\langle f \rangle = f(1-q)^d$ in the population. Assume that the population discovers one or more strings of higher fitness $f + \delta f$, which themselves have an *additional* b defining bits. If strings of fitness $f + \delta f$ were to stabilize, the average fitness would become $\langle f \rangle = (f + \delta f)(1 - q)^{d+b}$. However, such strings can only stabilize when the relative difference between these two average fitnesses is larger than the finite-population sampling fluctuations; and the latter are of order $1/\sqrt{M}$. Thus, for the higher-fitness strings to be noticed by the dynamics, the condition

$$\delta f \geq \frac{f}{(1-q)^b}\left(\frac{1}{\sqrt{M}} + 1 - (1-q)^b\right)$$

must be met. Below this fitness differential, strings of fitness $f + \delta f$ are effectively neutral with respect to strings with fitness f.

The net result is that the search parameters, such as q and M, determine a *coarse graining* of fitness levels, where strings in the band of fitness between f and $f + \delta f$ are treated as having equal fitness. This is important since it shows that even for fitness functions that do not have explicit neutrality, neutrality may still be effectively *induced* by the dynamics.

By tuning the evolutionary parameters q and M one effectively coarse grains the fitness "landscape", as if one were squinting at it. What we found in the preceding analysis is that for the Royal Staircase fitness functions, search was optimal when the staircase fitness steps were just discernible in this sense. It seems intuitive then that fitness functions containing many local optima, but that by definition have no neutrality, might still be searched efficiently by a genetic algorithm provided that there exists a level of coarse graining which turns the "rugged fitness landscape" into an architecture of interwoven neutral networks. These would be fitness functions that, at the smallest scales, possess many local optima, but that at some well chosen scale have their local optima hidden by the induced coarse graining. To put it somewhat differently, a fitness function may be efficiently searched by a GA if there is some scale of coarse graining at which higher-fitness strings "point the way" to strings of even higher fitness.

With these observations in mind, our analysis suggests that the question of what fitness functions can be efficiently searched by evolutionary methods translates into the question of what fitness functions can be turned into neutral network architectures by a suitable coarse graining.

Appendix A: Selection operator

Since the GA uses 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, $P_i^s = ciP_i$. The constant c can be determined by demanding that the distribution remains normalized. Since the average fitness $\langle f \rangle$ of the population is given by Eq. (3), we have $P_i^s = iP_i/\langle f \rangle$. In this way, we define a (diagonal) operator **S** that acts on a fitness distribution $\vec{P}$ and produces the fitness distribution $\vec{P}^s$ after selection by:

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

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

Appendix B: Mutation operator

The component $\mathbf{M}_{ij}$ of the mutation operator gives the probability that a string of fitness j is turned into a string with fitness i under mutation.

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 ith block in the string. Multiplying the probabilities that the preceding $i - 1$ blocks remain aligned and that the ith block becomes unaligned we have:

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

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

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

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 aligned configuration. Thus, the diagonal components are given by:

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

Finally, we calculate the probabilities for increasing-fitness transitions $\mathbf{M}_{ij}$ with $i > j$. These transition probabilities depend on the states of the unaligned blocks j through i. Under the maximum entropy assumption all these blocks are random. The jth 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, all the first $j - 1$ aligned blocks have to remain aligned, then the jth 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 ith block has to be unaligned. Putting these together, we find that:

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

The last factor does not appear for the special case of the global optimum, $i = N + 1$, since there is no $(N + 1)$st block.

Appendix C: Epoch fitnesses and quasispecies

The generation operator $\mathbf{G}$ is given by the product of the mutation and selection operators derived above; i.e. $\mathbf{G} = \mathbf{M} \cdot \mathbf{S}$. The operators $\mathbf{G}^n$ are defined as the projection of the operator $\mathbf{G}$ onto the first n dimensions of the fitness distribution space. Formally:

$$\mathbf{G}_i^n[\vec{P}] = \mathbf{G}_i[\vec{P}], \quad i \leq n,$$

and, of course, the components with $i > n$ are zero.

The epoch quasispecies $\vec{P}^n$ is a fixed point of the operator $\mathbf{G}^n$. As in Section 6, we take out the factor $\langle f \rangle$ to obtain the matrix $\tilde{\mathbf{G}}^n$. The epoch quasispecies is now simply the principal eigenvector of the matrix $\tilde{\mathbf{G}}^n$ and this can be easily obtained numerically.

However, in order to obtain analytically the form of the quasispecies distribution $\vec{P}^n$ during epoch n we approximate the matrix $\tilde{\mathbf{G}}^n$. As shown in Appendix B, the components $\mathbf{M}_{ij}$ (and so of $\tilde{\mathbf{G}}^n$) 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. These terms are generally much smaller than the terms that only involve 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 the rate of aligned-block creation—what was called P_a in Appendix B. Under this approximation for the components of $\tilde{\mathbf{G}}^n$, we have:

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

and

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

The components with $i > j$ are now all zero.

Note first that all components of $\tilde{\mathbf{G}}^n$ only depend on q and K through $\lambda \equiv (1-q)^K$, the probability that an aligned block is not destroyed by mutation. Note further that in this approximation $\tilde{\mathbf{G}}^n$ is upper triangular. As is well known in 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 $\tilde{\mathbf{G}}^n$. That is,

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

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

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

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

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

For the component P_n^n this reduces to

$$P_n^n = \prod_{j=1}^{n-1} \frac{n\lambda^{n-j} - j}{n\lambda^{n-1-j} - j}.$$

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

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

Acknowledgments

The authors thank the referees for helpful suggestions. This work was supported at the Santa Fe Institute by NSF Grant IRI-9705830, ONR grant N00014-95-1-0524, and DARPA contract F30002-00-2-0583.

References

Adami, C. (1995). Self-organized criticality in living systems. *Phys. Lett. A 203*, 29–32.

Alves, D. & J. F. Fontanari. (1998). Error thresholds in finite populations. *Phys. Rev. E 57*, 7008–7013.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York, NY: Oxford University Press.

Barnett, L. (1997). Tangled webs: Evolutionary dynamics on fitness landscapes with neutrality. Master's thesis, School of Cognitive Sciences, University of East Sussex, Brighton. http://www.cogs.susx.ac.uk/lab/adapt/nnbib.html.

Belew, R. K. & Booker, L. B. (Eds.) (1991). In *Proceedings of the Fourth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann.

Chambers, L. (Ed.) (1995). *Practical Handbook of Genetic Algorithms*. Boca Raton: CRC Press.

Crutchfield, J. P. & Mitchell, M. (1995). The evolution of emergent computation. *Proc. Natl. Acad. Sci. U.S.A. 92*, 10742–10746.

Crutchfield, J. P. & van Nimwegen, E. (1999). The evolutionary unfolding of complexity. In L. F. Landweber, E. Winfree, R. Lipton, & S. Freeland (Eds.), *Evolution as Computation*. SFI Working Paper 99-02-015; http://xxx.lanl.gov/abs/adap-org/9903001.

Davis, L. D. (Ed.). (1991). *The Handbook of Genetic Algorithms*. Van Nostrand Reinhold.

Eigen, M. (1971). Self-organization of matter and the evolution of biological macromolecules. *Naturwissen., 58*, 465–523.

Eigen, M., McCaskill, J., & Schuster, P. (1989). The molecular quasispecies. *Adv. Chem. Phys., 75*, 149–263.

Eigen, M. & Schuster, P. (1977). The hypercycle. a principle of natural self-organization. Part A: Emergence of the hypercycle *Naturwissen.*, *64*, 541–565.

Elena, S. F., Cooper, V. S., & Lenski, R. E. (1996). Punctuated evolution caused by selection of rare beneficial mutations. *Science*, *272*, 1802–1804.

Eshelman, L. (Ed.) (1995). In *Proceedings of the Sixth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann.

Ewens, W. J. (1979). *Mathematical Population Genetics* (Vol. 9 of *Biomathematics*). Springer-Verlag, Berlin.

Fontana, W. & Schuster, P. (1998). Continuity in evolution: On the nature of transitions. *Science*, *280*, 1451–1455.

Forrest, S. (Ed.) (1993). In *Proceedings of the Fifth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann.

Gantmacher, F. R. (1959). *Applications of the Theory of Matrices*. New York: Interscience Publishers.

Gardiner, C. W. (1985). *Handbook of Stochastic Methods*. Springer-Verlag, Berlin.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

Gould, S. J. & Eldredge, N. (1977). Punctuated equilibria: The tempo and mode of evolution reconsidered. *Paleobiology*, *3*, 115–251.

Hartl, D. L. & Clark, A. G. (1989). *Principles of Population Genetics*. (2nd edn.) Sunderland, MA: Sinauer Associates.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: The University of Michigan Press.

Huynen, M., Stadler, P. F., & Fontana, W. (1996). Smoothness within ruggedness: The role of neutrality in adaptation. *Proc. Natl. Acad. Sci.*, *93*, 397–401.

Huynen, M. A. (1995). Exploring phenotype space through neutral evolution. *J. Mol. Evol.*, *43*, 165–169.

Kauffman, S. (1993). *The Origins of Order*. Oxford University Press.

Kauffman, S. A. & Levin, S. (1987). Towards a general theory of adaptive walks in rugged fitness landscapes. *J. Theo. Bio.*, *128*, 11–45.

Kimura, M. (1964). Diffusion models in population genetics. *J. Appl. Prob.*, *1*, 177–232.

Kimura, M. (1983). *The Neutral Theory of Molecular Evolution*. Cambridge University Press.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Leuthäusser, I. (1987). Statistical mechanics of Eigen's evolution model. *J. Stat. Phys.*, *48*, 343–360.

Macken, C. A. & Perelson, A. S. (1989). Protein evolution in rugged fitness landscapes. *Proc. Nat. Acad. Sci.*, *86*, 6191–6195.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

Mitchell, M., Crutchfield, J. P., & Hraber, P. T. (1994a). Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D*, *75*, 361–391.

Mitchell, M., Holland, J. H., & Forrest, S. (1994b). When will a genetic algorithm outperform hill climbing? In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in Neural Information Processing Systems 6* (Vol. 6). San Mateo, CA: Morgan Kaufmann.

Newman, M. & Engelhardt, R. (1998). Effects of selective neutrality on the evolution of molecular species. *Proc. R. Soc. London B.*, *265*, 1333–1338.

Nowak, M. & Schuster, P. (1989). Error thresholds of replication in finite populations, mutation frequencies and the onset of Muller's ratchet. *J. Theo. Bio.*, *137*, 375–395.

Ohta, T. (1973). Slightly deleterious mutant substitutions in evolution. *Nature*, *247*, 96–98.

Ohta, T. & Gillespie, J. H. (1996). Development of neutral and nearly neutral theories. *Theo. Pop. Bio.*, *49*, 128–142.

Prügel-Bennett, A. & Shapiro, J. L. (1994). Analysis of genetic algorithms using statistical mechanics. *Phys. Rev. Lett.*, *72*(9), 1305–1309.

Prügel-Bennett, A. & Shapiro, J. L. (1997). The dynamics of a genetic algorithm in simple random Ising systems. *Physica D*, *104*(1), 75–114.

Rattray, M. & Shapiro, J. L. (1996). The dynamics of a genetic algorithm for a simple learning problem. *J. Phys. A*, *29*(23), 7451–7473.

Reidys, C. M., Forst, C. V., & Schuster, P. K. (2001). Replication and mutation on neutral networks. *Bull. Math. Bio.*, *63*(1), 57–94.

Swetina, J. & Schuster, P. (1982). Self replicating with error, a model for polynucleotide replication. *Biophys. Chem.*, *16*, 329–340.

van Nimwegen, E. & Crutchfield, J. P. (2000). Optimizing epochal evolutionary search: Population-size independent theory. In D. Goldberg & K. Deb (Eds.), *Computer Methods in Applied Mechanics and Engineering, special issue on Evolutionary and Genetic Algorithms in Computational Mechanics and Engineering*, (Vol. 186), pp. 171–194.

van Nimwegen, E., Crutchfield, J. P., & Mitchell, M. (1997). Finite populations induce metastability in evolutionary search. *Phys. Lett., A 229*, 144–150.

van Nimwegen, E., Crutchfield, J. P., & Mitchell, M. (1999). Statistical dynamics of the Royal Road genetic algorithm. A. Eiben & G. Rudolph, (Eds.), *Theoretical Computer Science, special issue on Evolutionary Computation*, (Vol. 229), (pp. 41–102). SFI working paper 97-04-35.

Weber, J. (1996). Dynamics of neutral evolution. A case study on RNA secondary structures. Ph.D. Thesis, Biologisch-Pharmazeutischen Fakultät der Friedrich Schiller-Universität Jena.

Wolpert, D. H. & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Trans. Evol. Comp.*, *1*, 67–82.

Wright, S. (1982). Character change, speciation, and the higher taxa. *Evolution*, *36*, 427–43.

Received October 29, 1998
Revised September 21, 1999
Accepted November 12, 1999
Final manuscript December 1, 1999