Evolutionary Search on Fitness Landscapes with Neutral Networks

Lionel Barnett

Submitted for the degree of D. Phil.

University of Sussex

August, 2003

Declaration

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for a degree.

Signature:

Acknowledgements

Cuando me buscas, no estoy
Cuando me encuentras, no soy yo

Manu Chao – *El Desaparecido*

To Inman, for the awkward questions.
To Txaro and Asier, for their support and impatience.

Evolutionary Search on Fitness Landscapes with Neutral Networks

Lionel Barnett

Summary

In the field of search and optimisation every theorist and practitioner should be aware of the so-called *No Free Lunch Theorems* (Wolpert & Macready, 1997) which imply that given any optimisation algorithm, should that algorithm perform better than random search on some class of problems, then there is guaranteed to exist another class of problems for which the same algorithm performs *worse* than random search. Thus we can say for certain that there is no such thing as an effective "general purpose" search algorithm. The obverse is that *the more we know about a class of problems, the better equipped we are to design effective optimisation algorithms for that class*. This thesis addresses a quite specific class of optimisation problems - and optimisation algorithms. Our approach is to analyse statistical characteristics of the problem search space and thence to identify the algorithms (within the class considered) which *exploit* these characteristics - we pay for our lunch, one might say.

The class of optimisation problems addressed might loosely be described as *correlated fitness landscapes with large-scale neutrality*; the class of search algorithms as *evolutionary search processes*. Why we might wish to study these problems and processes is discussed in detail in the Introduction. A brief answer is that problems of this type arise in some novel engineering tasks. What they have in common is huge search spaces and inscrutable complexity arising from a rich and complex interaction of the designed artifact with the "real world" - the messy world, that is, outside our computers. The huge search spaces and intractable structures - and hence lack of obvious design heuristics - suggests a *stochastic* approach; but "traditional" stochastic techniques such as Genetic Algorithms have frequently been designed with rather different search spaces in mind. This thesis examines how evolutionary search techniques might need to be be re-considered for this type of problem.

Submitted for the degree of D. Phil.

University of Sussex

August, 2003

Contents

List of Figures

Chapter 1

Introduction

Consider the following engineering problems:

- designing a controller for an autonomous robot
- designing an electronic circuit to perform a complex task
- designing a complex timetable

These problems have in common that they are difficult to solve from a traditional engineering perspective of design by analysis and heuristics - what we might term “hand design”.

This thesis is concerned with the use of stochastic and in particularly evolutionary techniques for “optimisation” of such complex problems. By an optimisation problem, we mean loosely the following: we have a range of possible designs representing possible means of achieving some task. To each such design we can attach a numerical value representing how well the design performs the task at hand. We shall assume that the larger this value the better the performance; the numerical value is then known as the *fitness* of a design. It is further assumed that we are going to attempt to solve optimisation problems using computers. Our design must therefore be representable in a form which may be stored in computer memory. We do not necessarily demand that the evaluation of performance of a design take place purely within a computer, however. For many of the types of problems which we have in mind to address, evaluation of designs - execution of the task to be performed - takes place “in the real world”; that is, in the world outside of the computer environment within which we manipulate designs.

The methodology we address is variously known as “evolutionary search”, “evolutionary computation” or “genetic algorithms”, by analogy with natural evolution. Specifically we imagine our designs to be *phenotypes* and the fitness to be analogous to biological fitness in some Darwinian sense. To extend the biological analogy, putative designs are coded for by sequences of symbols or numbers representing a *genotype* which maps (usually unambiguously) to a specific phenotype. The resulting mapping of genotype (via phenotype/design) to numerical fitness is often referred to as specifying a *fitness landscape*, a concept introduced to the study of biological evolution by Sewall Wright (S. Wright, 1932). An evolutionary process is then performed on a *population* of

such genotypes in computer memory, with the objective of evolving genotypes which map to high fitness phenotypes.

The arena of the evolutionary process is thus the fitness landscape. Given a genotype → phenotype encoding and a phenotype → fitness mapping we may consider the fitness landscape thereby defined *in abstracto* and consider optimisation on the fitness landscape as an object of study in itself. Of course there will be many possible genotype → phenotype and phenotype → fitness mappings corresponding to a given design problem, which may be more or less tractable to solution by evolutionary methods; designing suitable mappings may in itself be a highly non-trivial problem. In this thesis, we are not principally concerned with "fitness landscaping" (which may, for all we know, need to be performed "by hand" and with goodly measures of skill, experience and ingenuity). We generally consider the fitness landscape as *given*; our concern is then how best to deploy evolutionary methods in optimising on a given fitness landscape. The situation is not quite that straightforward, however. As we explain in the next section, the choice of optimisation technique is inextricably bound to *what we know* about our fitness landscape; and what we know will depend on how the landscape was designed . . . Nevertheless we will tend to sidestep this issue as far as possible. If it happens that our approach has something to say to the design of genotype → phenotype and phenotype → fitness mappings so to the good; this is not, however, our primary concern.

In this thesis we will not be addressing evolutionary search in any general sense. Specifically, we will be examining evolutionary search and evolutionary processes on fitness landscapes which posses two (statistical) features: *fitness correlation* and *selective neutrality*. The first (as we argue in the next section) might be regarded as inevitable, in the sense that it would not be appropriate to apply evolutionary methods on a landscape *lacking* this property. The second property, neutrality, while not (like the first) a prerequisite, has received increasing attention over the past few years. There is gathering evidence that it is prevalent in many real-world engineering problems and that, furthermore, appropriate techniques for effective evolutionary search on landscapes featuring substantial neutrality may well differ quite radically from more traditional approaches to evolutionary search. Thus much of what we shall have to say may seem jarring to those schooled in a more traditional approach. We would like to assure the reader that there is no intention to be controversial - rather we would like the reader to bear in mind that, particularly as regards neutrality, we are addressing a rather specific class of problem and possibly one rather different from the more traditional optimisation scenario.

We should also like to remark the following: from the inception of the somewhat hazy area that has come to be known as Artificial Life (under the ambit of which evolutionary search could be said to fall), there has always been a hope that the study of artificial life-*like* phenomena might feed back fruitfully into the study of *real* life-as-we-know-it. While perhaps as much true of evolutionary search, this is not our primary concern and any relevance this work may hold for biological evolution should be viewed as purely serendipitous.

1.1 Evolutionary Search

The essence of our problem may be stated as follows: suppose it were possible to collect *all possible* fitness landscapes (i.e. genotype → fitness mappings). Suppose then that a notional

problem-poser picks one fitness landscape out of this collection, hands it to us and asks us to optimise (i.e. find high fitness genotypes) on the landscape he has chosen for us. How would we go about this? One possibility is that we simply evaluate the fitness of every genotype for the given landscape and then keep (one of) the highest fitness genotype(s) we found. This would certainly solve the problem at hand. In practice however, for non-trivial optimisation problems, the "space" of all genotypes generally turns out to be vast - so vast, in fact, that even with the fastest technology available (or even imaginable) the time required to process every possible genotype tends to be measured in units of "age of the universe". Exhaustive enumeration of fitnesses is simply not an option.

Thus it is clear that, depending on the time and technology resources available to us we are only going to be able to evaluate a fraction of the genotypes before us. How should we choose which ones to evaluate? The uncomfortable answer (and perhaps surprisingly there is one if the above scenario is stated in precise mathematical terms) is that *we can do no better than enumerating fitnesses* - until we run out of time or patience or, with a lot of luck, stumble upon a genotype of high enough fitness to keep us happy. The technical version of this result is a variety of what have been aptly dubbed *No Free Lunch Theorems* (Wolpert & Macready, 1997). The point is that, as we have presented the problem, we simply don't know enough about the genotype $\rightarrow$ fitness mapping to be able to make better than arbitrary choices about which genotypes to examine.

In short, to have a fighting chance of optimising anything, we must know something about our fitness landscape. Our problem-poser cannot simply take a lucky dip from the bag of all possible landscapes. He must, effectively, bias his choice - and he must tell us what this bias is! But why should our problem-poser bias his choices? The brief (and somewhat circular) answer, is that he knows we are going to use evolutionary techniques to solve his problem and therefore he will attempt to design the fitness landscape so as to be amenable to such techniques! The question then becomes: how should one design a fitness landscape so as to be amenable to an evolutionary approach? To answer this we need some knowledge as to how evolution finds fitter genotypes.

Evolution proceeds via *inheritance with random variation* and *selection*. That is, new "offspring" genotypes are created from existing "parent" genotypes by some "error-prone" procedure (inheritance with random variation) and genotypes are eliminated (selection). Why should this yield a viable search mechanism? The essential point is that *variation should have a "better than arbitrary" chance of finding fitter genotypes*. To see how this might occur we turn to natural evolution. Natural evolution is *incremental*; new fitter phenotypes do not evolve via huge speculative jumps in "phenotype space" - so-called "hopeful monsters" (Goldschmidt, 1933; Goldschmidt, 1940; Dennett, 1995) - they arise through series of small changes. Note that this statement implies that phenotype space is *structured*; i.e. there is a notion of "similarity" or "nearness" of phenotypes - a *metric* structure. The mechanisms of variation in natural evolution are *mutation* and *recombination* of genotypes. Now if these mechanisms produced arbitrary change to phenotypes (via the genotype $\rightarrow$ phenotype mapping) - that is to say that a phenotype produced by mutation/recombination had no tendency to resemble its "parent" genotype(s) - we would not expect to see this incrementality. The inference to be drawn is that the variation mechanisms have a tendency to produce *small* changes to the phenotype. Now it may be argued that, for example, most mutations of the genotype of a real organism will be fatal - surely a large jump in pheno-

type space! But we are not saying that *all* variation produces small changes to the phenotype - merely that there is a better than random chance that it might. Natural evolution does not search phenotype space at random.

We can re-phrase the above by saying that the mechanisms of variation at the genotype level, the *genetic operators*, respect (in a probabilistic sense) the metric structure of phenotype space - the phenotypes of a genotype and its offspring are *correlated*. To complete this chain of reasoning, we note that the *fitness* of similar phenotypes also tend to be similar; phenotype and fitness too are correlated. Thus the fitness of genotypes and their offspring are correlated. We might indeed claim that it is *precisely* this parent-offspring/fitness correlation that makes evolution feasible as a search technique for higher fitness phenotypes. For if no such correlation existed our "evolutionary" search would simply be random - which is a little worse than exhaustive enumeration!

To return, then, to the question facing our problem-poser as to how he should design his fitness landscape to be amenable to evolutionary search, we have a (partial) answer: he should ensure that there are suitable genetic operators whereby the fitness of parents and their offspring are correlated. How might he be able to do this? A vague answer is that the problem he is attempting to solve will suggest a suitable design... we illustrate this by an example - the evolutionary design of a controller for an autonomous robot (Jakobi & Quinn, 1998; Jakobi, Husbands, & Smith, 1998).

1.1.1 The Fitness Landscaper - An Anecdotal Example

Our problem-poser wishes to build a controller for a robot that is to perform a navigation task. He has an actual robot with well-defined sensory-motor capabilities and an on-board computer capable of interacting with its sensory-motor hardware. He wishes to supply the robot with software that will cause it to perform the navigation task to his satisfaction. The design he seeks will thus take the form of a computer program that will run in the robot's processor. He will evaluate the fitness of a design for control software by actually running the robot with the designated program through the navigation task and awarding the design a fitness score according as to how well the task is performed; i.e. better performance is awarded higher fitness.

Let us suppose that he has tried to hand-code programs to control his robot but found it simply too complex and difficult. An attempt at writing a decision-making rule-based AI program foundered on the combinatorial explosion of case scenarios. An attempt to model hierarchies or networks of discrete behavioural modules was slightly more successful but still ran into a wall of complexity. At this point he considered using stochastic techniques and decided to attempt an evolutionary approach. Since the goal of the exercise is to produce a suitable computer program, his initial inclination was to attempt to evolve programs that could be compiled to run on the robot's processor. However a problem became apparent: as a programmer himself, he was well aware that introducing *any* kind of variation into a viable computer program almost always breaks it or, worse still, introduces syntax errors. This would immediately fall foul of the correlation requirement - small changes to a phenotype invariably produce huge (and detrimental) changes in fitness. There does not seem to be enough incrementality inherent in the approach.

The next line of attack seemed more promising. Noting that the desired behaviour of his robot might not be dissimilar to that of a simple organism facing a comparable navigation task, he considered using a phenotypic control mechanism modelled on that used by real organisms - a neural

network. Of course his artificial neural network would be simply an analogy - it would not begin to approach in sophistication or complexity that of any organism capable of solving the navigation task - but at least it did appear to have some desirable features. In particular, there seemed to be a natural metric (of sorts) on the phenotype space - two neural networks could be considered "nearby" in neural network space if they had similar network topologies and the connection weights and other parameters were similar. Importantly, a small change to the phenotype with respect to this metric - changing a network weight slightly or even adding a new node, for instance - did not necessarily have too drastic an effect on fitness; fitnesses of nearby phenotypes appeared to be correlated.

It remained to devise a genotype $\rightarrow$ phenotype mapping and some genetic operators which would respect the metric structure of the phenotype space; that is, applying a genetic operator to a parent genotype (or parent genotypes) would produce offspring genotypes whose phenotypes would be close to that of the parent(s). This turned out not to be too difficult. The network topology could be easily described as a string of bits, such that flipping one or two bits made small changes to the network topology (such as adding or deleting a single node or connection). The weights and other numerical parameters could easily be coded as floating-point numbers; applying a small displacement (via a pseudo-random number generator generating small Gaussian deviates, say) produced nearby values. He even found that by Gray-coding parameters rather than using floating-point coding, flipping single bits would in general produce smallish changes in value; the entire phenotype, including network topology and numerical parameters, could be coded for in a single bit-string. In fact a computer-friendly description of an artificial network virtually yielded *in itself* a suitable candidate for a genotype. The genetic operators looked a lot like natural mutation - they applied to a single genotype, the bits looked like alleles and so on. He experimented with recombination, but this turned out to be trickier; offspring produced by "crossing over" neural network designs did not tend to be so pleasantly correlated in fitness with their parents (the problem seemed to be that, in terms of behaviour of the controller, there was too much synergy between separate elements of a neural network - they did not seem to fall apart into convenient modules that could be successfully mix-and-matched).

We the optimisers, however, were not aware of these details - in fact we weren't even sure what problem he was trying to solve. Our problem-poser merely passed us his fitness landscape (he agreed, of course, to evaluate the fitness of any bit-strings we produced). He did, however, mention that if we took a genotype and flipped a few bits, the fitness of the resulting offspring genotype was quite likely to be nearby that of the parent. He also mentioned, although he doubted whether it would be of any interest to us, an observation he had made while experimenting with codings: surprisingly frequently, flipping a few bits of a genotype would produce an offspring of not merely *similar*, but rather *identical* fitness (we were in fact quite interested). In short, he told us something about the bias in his choice of landscape. We quickly evolved neural network controllers which solved his robot navigation problem; how we did so is the subject of this thesis.

1.1.2 Model Landscapes

The preceding discussion and example raise one rather curious philosophical question: was it, in fact, necessary for the problem-solver to have informed us of the correlation properties of the

fitness landscape? For if this property were not present we know that any attempt to optimise by evolutionary means would probably be pointless. Why should we not just *assume* correlation? The answer seems to be that we may as well. Furthermore, as will become clear, we would in any case find out sooner rather than later if there were no correlation. We thus assume that, as evolutionary searchers, we are always dealing with at least *some* degree of correlation; this will be our minimal assumption.

More broadly, the other side of the "No Free Lunch" coin is that the more we know *a priori* about the landscape we are attempting to optimise, the better we can analyse search processes and thus hone search techniques. Thus, for instance, while we are already assuming *some* correlation it might indeed be useful to know a bit more; *how much*, for instance, or how correlation varies with parameters (such as mutation rate) of our genetic operators. Considering the example from the previous section it might seem unlikely that our problem-poser would be able to tell us much more, short of virtually solving the problem himself. One way out of this conundrum might be as follows: in practice we consider structural or statistical knowledge about a landscape as constituting a "model". We then, when faced with an actual fitness landscape, assume some model, for which we have deduced analytically effective search techniques. If our model assumption (such as correlation) happens to have been wrong then our chosen search technique will quite likely not prove to be effective. We are then free to choose a weaker or alternative model.

It might be said that useful models for real-world fitness landscapes (and in particular those featuring large-scale neutrality) are sorely lacking in the literature. We would argue that, in fact, the study of evolutionary search techniques has been somewhat skewed by the study of inappropriate models. A large part of this thesis is devoted to the presentation of several (hopefully useful) models, generally described in terms of statistical properties, for which analysis may be performed and optimisation techniques derived. Whether or when these models might apply may be based on empirical evidence, heuristics, guesswork, hearsay or wishful thinking. If the assumptions of a particular model turn out to apply to a given problem, or class of problems, well and good; the model is then *useful*. If not we may pick another model off the shelf (perhaps after re-examination of the problem at hand) and try again.

1.1.3 Landscape Statistics

As regards statistical knowledge (or assumptions) regarding a landscape we seek to optimise, an awkward point presents itself. Statistical statements about a landscape, or class of landscapes, tend to be phrased in terms of *uniformly random sampling* of genotype space. This is the case, for example, for the usual definition of fitness correlation statistics; it is normal to talk about the correlation of fitness between a parent genotype chosen *uniformly at random* and its offspring. But, we must ask, will this be relevant to analysis of a particular evolutionary process on the landscape in question? For in the course of execution of a search process the sample of genotypes encountered is, almost by definition, likely to be far from random - in particular, it will hopefully be biased toward higher-fitness genotypes! Thus, for instance, if some landscape statistic suggests that the fitness distribution of the offspring of a parent genotype chosen uniformly at random from the landscape takes a particular form, can we suppose that a similar distribution will hold for a genotype encountered *in the course of an evolutionary process* on that landscape? The answer

would seem to be an unequivocal "no". For many real-world optimisation problems, for example, it is frequently the case that "most" genotypes turn out to be of very low fitness. This is certainly true of natural evolution where an arbitrary genotype (e.g. a string of DNA with randomly chosen nucleotides) is almost certain to be inviable. In this case the very part of the landscape we are interested in - the higher fitness genotypes - would hardly "show up" in a statistic based on uniform random sampling.

A partial answer might be to consider statistics *conditional on fitness*. This, at least, addresses the fitness bias inherent in any useful search process and we shall indeed consider fitness-conditional statistics. It will not address other biases, some of which will be identified in the course of this thesis. Ultimately the argument becomes circular: the statistics relevant to a particular search process depend on the process itself; the analysis of potentially effective search processes depends on the statistics available to the analyst. In practice we are forced to assume that an available (or assumed) statistic, be it based on whatever sample, at least *approximates* the "real" statistic that would apply to the sampling performed by the process under analysis. The situation is somewhat akin to the *maximum entropy* approximations made in statistical physics. Later we shall explicitly introduce comparable approximations to our analysis.

This leads us to the following: in the course of execution of an evolutionary process on a given fitness landscape, we are evaluating the fitness of genotypes encountered along the way. We can thus compile statistics pertaining to the landscape structure (at least at those genotypes we have encountered so far) with a view, perhaps, to altering "on the fly" our search strategy so as to exploit this extra structural information. Apart from the caveats of the preceding paragraph this seems reasonable. There is, however, no guarantee that the statistics we compile in the future course of a process will resemble those gathered up to the current time, even taking into account fitness-conditional structure; the fitness landscape may be far from "homogeneous". Thus to analyse a "self-tuning" search strategy as suggested homogeneity, or perhaps more accurately fitness-conditional homogeneity, may have to be introduced as a further approximation.

1.2 Why Neutrality?

The phenomenon of *selective neutrality*, the significance of which has been (and periodically continues to be) much debated in population and molecular genetics, was thrust centre-stage by Kimura (Kimura, 1983; Crow & Kimura, 1970), who questioned the preeminence of *selection* as the sole mediator of the dynamics of biological evolution, at least at a molecular level. Interest in selective neutrality was re-kindled more recently by the identification of *neutral networks* - connected networks of genotypes mapping to common phenotypes (and therefore equal fitness) - in models for bio-polymer sequence $\rightarrow$ structure mappings; in particular for RNA secondary structure folding (Schuster, Fontana, Stadler, & Hofacker, 1989; Fontana et al., 1993; Grüner et al., 1996) and protein structure (Babajide, Hofacker, Sippl, & Stadler, 1997; Bastolla, Roman, & Vendruscolo, 1998). This research, performed largely *in silico*, was expedited by the availability of increasing computing power, the development of fast and effective computational algorithms for modelling the thermodynamics of bio-polymer folding (Zuker & Sankoff, 1984; Hofacker et al., 1994; Tacker, Fontana, Stadler, & Schuster, 1994) and also the increased sophistication of *in vitro* techniques (Ekland & Bartel, 1996; M. C. Wright & Joyce, 1997; Landweber & Pokrovskaya,

1999). Our interest stems from the growing evidence that such large-scale neutrality - and indeed neutral networks in the sense intended by RNA researchers - may be a feature of fitness landscapes which arise for the type of complex real-world engineering problems described above (Cliff, Husbands, & Harvey, 1993; Thompson, 1996; Harvey & Thompson, 1996; Thompson & Layzell, 1999; Layzell, 2001; McIlhagga, Husbands, & Ives, 1996; Smith, Philippides, Husbands, & O'Shea, 2002; Smith, Husbands, & O'Shea, 2001). This neutrality, it would seem, stems from the following: in a complex design involving many parameters and, perhaps, many "features" contributing to overall fitness, tweaking a particular feature will frequently have no effect on fitness since the feature tweaked may in fact be making no discernible contribution to fitness - at least within the "context" of other features. Thus, for instance, in an electronic circuit, changing the configuration of a circuit element will make no difference to the behaviour of the circuit if the element is not - in the current design context - actually connected[1] to the output on which fitness is evaluated! This effect is, in fact, precisely the basis for one of our classes of model landscapes - the NKp model of Chapter 6. It may also be that tweaking a parameter has no discernible effect on fitness because that parameter is set to some "extreme" value and a mere tweak is not enough to make it less than extreme. An example of this might be a weight in a neural network set to such a high value that it "saturates" a node for which the corresponding connection acts as an input. This variety of neutrality might be said to stem from the *encoding* of the parameter.

It is also becoming clear that the *dynamics* of evolutionary processes on fitness landscapes with neutrality are qualitatively very different from evolutionary dynamics on rugged landscapes (Huynen, Stadler, & Fontana, 1996; Forst, Reidys, & Weber, 1995; Reidys, Forst, & Schuster, 1998; Nimwegen, Crutchfield, & Mitchell, 1997; Nimwegen, Crutchfield, & Mitchell, 1997; Nimwegen & Crutchfield, 1998; Nimwegen & Crutchfield, 1998; Nimwegen & Crutchfield, 1999; Harvey & Thompson, 1996; Barnett, 1997; Barnett, 1998; Barnett, 2001). A major impetus for this body of work, then, is quite simply the lack of suitable models - and indeed theory - for such landscapes and the perception that the common (rugged, multi-modal and non-neutral) perception of landscape structure in the GA literature is inapplicable, if not actively misleading, for optimisation of the class of evolutionary scenarios that we intend to address.

1.3 Overview

1.3.1 Organisation

In brief, the organisation of this thesis is as follow:

Chapter 2 is largely formal: in the first Section we introduce the concepts of *sequence space* and *fitness landscape* (in the context of artificial evolution) and the partitioning of a fitness landscape into *neutral networks*. In the second Section we introduce mutation and the *mutation matrix* for a neutral partitioning; the remainder of the Section is devoted to setting up a framework for the study of the structural aspects of a fitness landscape (with respect to mutation) which depend only on a (neutral) partitioning of the landscape rather than on actual fitness values. In particu-

[1] (Layzell, 2001) relates an amusing instance where certain elements of an artificial evolution-designed circuit on an FPGA chip, whilst apparently physically unconnected to the "active" part of the circuit, manifestly *did* affect the behaviour of the circuit. It transpired that the element *was* effectively 'connected" - by electromagnetic interference. Other evolved circuits have been found (or encouraged) to exploit quantum-mechanical effects (Thompson & Wasshuber, 2000). Evolution is, as Orgel's Second Rule has it, "cleverer than you" (Dennett, 1995).

lar, we define statistics based on uniform sampling of neutral networks, which will become the basis for a "maximum entropy-like" assumption in the following Chapter. The third Section examines fitness-dependent structural aspects of fitness landscapes (again with respect to mutation), in particular the *mutant fitness distribution*, *parent-mutant fitness correlation* and *evolvability*. It is shown that the optimal mutation mode for a neutral network involves mutating a fixed number of loci (rather than a per-locus mutation probability). The final Section examines how the concepts and measures introduced carry over to families of *random fitness landscapes*.

Chapter 3 is also concerned largely with formalities: the first Section introduces the notion of a *population* of sequences on a fitness landscape. The next Section introduces *evolutionary processes* (a formalisation/generalisation of *Genetic Algorithms*) which are defined by generational selection/mutation-based *evolutionary operators*. A range of evolutionary processes are presented as examples, including various *stochastic hill-climbers*. The third Section introduces a *maximum entropy approximation* for an evolutionary process, based on the coarse-graining of a fitness landscape into neutral networks. This *statistical dynamics* approach (Nimwegen et al., 1997) - modelled after comparable ensemble techniques in statistical mechanics - is presented as an analytic tool. The following Section describes and analyses the generic "epochal" dynamics of evolutionary processes on fitness landscapes with neutral networks, characterised by the successive breaching of *entropy barriers*, and contrasts this with the more conventional viewpoint of evolution on "rugged" landscapes featuring *fitness barriers*. The final Section looks at how the effectiveness of evolutionary processes in optimisation may be measured and compared.

Chapter 4 examines how, why and when *neutral drift* on neutral networks might benefit the search effectiveness of an evolutionary process. The *nervous ant neutral walk* is presented as a "tunable" analytical tool for the study of the effects of neutral drift. It is conjectured - and proved in a weak sense - that (modulo some strictures on *a priori* knowledge of landscape structure and evolutionary dynamics) drift will always benefit the search capabilities of an evolutionary process.

Chapter 5 introduces ε-*correlated landscapes*, characterised by a "ladder-like" structure controlled by a small scale parameter. Optimal mutation rates for neutral networks on ε-correlated landscapes are calculated and shown to obey (to a first approximation) a particularly simple heuristic, the $1/e$ *Neutral Mutation Rule*. Results from the previous Chapter are deployed to argue that the optimal evolutionary process on an ε-correlated landscape is a variety of stochastic hill-climber dubbed the *netcrawler*. An *adaptive* form of the netcrawler (based on the $1/e$ Neutral Mutation Rule) is described. Statistics are calculated explicitly for *Royal Road* landscapes - a class of ε-correlated landscapes - and theoretical results tested against Monte Carlo simulations. A range of evolutionary processes is trialled on Royal Road landscapes and results analysed in some detail.

Chapter 5 introduces *NKp landscapes*, a family of random landscapes with tunable ruggedness and neutrality. The first Section discusses background and motivations for NKp landscapes and details their construction. The following Section analyses the global (ensemble) statistics of NKp landscapes; in particular it is shown that auto-correlation on (generalised) NK landscapes does not depend on the underlying fitness distribution and that, consequently, ruggedness and neutrality are statistically independent for NKp landscapes. Neutral and "lethal" mutation are analysed via statistics based on the distribution of *contributing features*. The third Section analyses fitness-dependent (ensemble) statistics. In particular, *mean mutant fitness* is calculated and

NKp landscapes are shown to have the *linear correlation* property (thus providing another proof of the independence of ruggedness and neutrality). The fitness dependence of neutral and lethal mutation is calculated and the full mutant fitness distribution and evolvability statistics calculated for a *Gaussian* underlying fitness distribution. Optimal mutation rates are calculated (based on ensemble evolvability) and a new derivation for the $1/e$ Neutral Mutation Rule is given. The next Section discusses NKp landscapes as models for landscapes in artificial evolution. Baseline parameters are set up to test theoretical results empirically. The neutral network structure is investigated in more detail and some preliminary results on optimisation on NKp landscapes (with implications for the neutral network structure) are presented.

Previous Chapters have expressly rejected *recombination* as an effective mechanism in evolutionary optimisation on "real-world" artificial fitness landscapes; Chapter 7 addresses this prejudice. The first Section reviews some problems with the so-called *Building Block Hypothesis* and the *Schema Theorem*; in particular whether we should expect to find suitable "building blocks" in realistic problems and, if so, whether recombination is likely to be able to assemble them usefully. The following Section examines some well-known problems affecting the effectiveness of recombination as a result of finite-population sampling, or *genetic drift*. The phenomena of "premature convergence" and *hitch-hiking* are discussed. The third Section presents original work by the author on possible deleterious effects of recombination - identifiable in the infinite-population limit but exacerbated by finite-population sampling - which may arise as a result of local features of the fitness landscape. Through a *quasi-species* analysis, a *bi-stability barrier* and lowering of the *(mutational) error threshold* are identified in the presence of "non-synergistic" epistasis. Implications for evolutionary optimisation are discussed.

1.3.2 Summary of Conclusions

Perhaps our most radical conclusions will be that for the class of fitness landscapes considered - landscapes that might arise in real-world optimisation problems, featuring some correlation and large-scale neutrality:

1. *Recombination* is not likely to be an effective genetic operator. The driving mechanism behind evolutionary search will be *mutation*.
2. The most appropriate evolutionary search process is likely to be a population-of-1 *stochastic hill-climber* rather than a population-based GA. It should exploit *neutral drift*.
3. We may be able, under certain reasonable assumptions, to estimate an optimum mutation mode/rate; alternatively, we might deploy an *adaptive* regime.

The argument toward these conclusions involves several stages and extends over the entire thesis.

Chapter 2

Fitness Landscapes in Artificial Evolution

In this chapter we formally define *fitness landscapes* and introduce several statistical features associated with a landscape, notably *neutrality*, *correlation*, *percolation* and *evolvability*. Before we proceed, one possible source of confusion needs to be cleared up: to the biologist, "fitness" denotes a measure of survival and reproduction for genotypes (Maynard Smith, 1998; Crow & Kimura, 1970) in a population of organisms. In a simple scenario, this may mean something like "the expected number of offspring produced over the reproductive lifetime of an organism with that genotype". Fitness, then, is a measure of a genotype's propensity to reproduce itself *within a particular environment*, where "environment" may embrace other genotypes in the population under consideration, competing species, a changing geographical backdrop, *etc*. Thus to Sewall Wright, a fitness landscape denoted a landscape that, over time, might deform with the changing makeup of an evolving population and other ecological factors. To the optimiser, on the other hand, fitness is something rather more static and pre-ordained: fitness denotes the "objective function" - the quantity that is to be optimised. In this work we use "fitness" (and fitness landscape) exclusively in the optimiser's understanding of the term. We treat a fitness landscape henceforth, simply as a fixed mapping of genotype to a (numerical) fitness value. We stress again that our genotypes will always comprise sequences of *discrete* symbols, rather than continuous parameters and again warn the reader against the temptation to extrapolate results to the case of optimisation with continuous parameters[1]. As a further remark, we assume that as regards fitness, bigger means better; the object of optimisation is to *maximise* fitness. The reader should be aware that in some of the literature (particularly in work inspired by physics where "objective function" often equates to "energy") the object may be to *minimise* a corresponding quantity.

An (unavoidable) presentational difficulty that will frequently arise in this Chapter is the following: the statistical features of a fitness landscape that will be of interest to us are generally motivated by our analysis of the dynamics of evolutionary processes, which constitutes the subject matter of the following Chapter. There will thus inevitably be forward references.

[1] If continuous parameters are encoded into discrete representations (eg. via a binary or Grey coding scheme) then our framework will indeed apply. It is not clear, however, when (or why) one might want to deploy a discrete encoding for a problem with "natural" continuous parameters, as opposed to working directly with the continuous parameters themselves...

2.1 Definitions

Definition 2.1.1. Let $\mathcal{A}$ be a finite set and let $L > 0$. We call an element of $\mathcal{A}^L$, the set of L-tuples of elements from $\mathcal{A}$, a ***sequence of length*** L over the ***alphabet*** $\mathcal{A}$. Given a sequence $x = a_1 a_2 \dots a_L \in \mathcal{A}^L$ we refer to a_n as the ***allele*** at the n-th ***locus*** of x.

There is a natural (non-directed, regular) graph structure on $\mathcal{A}^L$, the ***Hamming graph*** structure, whereby two sequences are adjacent iff they differ at a single locus.

Definition 2.1.2. We call $\mathcal{A}^L$ with the Hamming graph structure the ***sequence space*** of sequences of length L over $\mathcal{A}$. Given sequences $x = a_1 a_2 \dots a_L$ and $y = b_1 b_2 \dots b_L$ in $\mathcal{A}^L$ the ***Hamming distance*** between x and y is defined by:

$$h(x,y) = L - \sum_{n=1}^{L} \delta(a_n, b_n) \tag{2.1}$$

where $\delta(a,b)$ is 1 if $a = b$ and 0 otherwise. Thus the Hamming distance between sequences is simply the number of loci at which the sequences have different alleles. Hamming distance defines a ***metric*** on $\mathcal{A}^L$.

Definition 2.1.3. A ***fitness landscape*** is a triple $\mathcal{L} = (\mathcal{A}, L, f)$, where $f : \mathcal{A}^L \longrightarrow \mathbf{R}$ is the ***fitness function***.

We will often, by abuse of terminology, refer to $\mathcal{L}$ as a fitness landscape over the sequence space $\mathcal{A}^L$. Throughout most of this thesis we shall restrict our attention to the ***binary alphabet*** $\mathcal{A} = \{0,1\}$; most constructions and results generalise straightforwardly to higher cardinality alphabets. Unless otherwise stated, the binary alphabet should be assumed.

2.1.1 Neutral Partitionings and Neutral Networks

As will be seen in the next Chapter, our approach to the analysis of evolutionary dynamics will be based on a *coarse-graining* of the fitness landscape coupled with a *maximum entropy* approximation. This will suppose a partitioning of the sequence space in a manner that respects the fitness mapping, in the sense that all sequences in an equivalence class map to the same fitness. We thus define:

Definition 2.1.4. A ***neutral partitioning*** of a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ is an equivalence relation on $\mathcal{A}^L$ such that $\forall x, y \in \mathcal{A}^L$, we have $x \equiv y \Rightarrow f(x) = f(y)$. The sequence space is thus a disjoint union $\mathcal{A}^L = \bigcup_{i=1}^{N} \Gamma_i$ where the N equivalence classes Γ_i are the ***neutral networks*** (or just ***networks***) of $\mathcal{L}$ with respect to the partitioning. As a notational convenience, for $x \in \mathcal{A}^L$ we write $\tilde{x}$ for the equivalence class associated with x. We also write $\widetilde{\mathcal{A}^L} = \{\Gamma_1, \Gamma_2, \dots, \Gamma_N\}$ for the set of neutral networks of the partitioning, which we identify when convenient with its index set $\{1, 2, \dots, N\}$. We call a neutral network ***connected*** iff it is connected with respect to the Hamming graph structure on the sequence space.

There is a natural partial ordering of neutral partitionings, whereby partitioning $1 \leq$ partitioning 2 iff $x \equiv_1 y \Rightarrow x \equiv_2 y$; we then say that partitioning 1 is ***finer*** than partitioning 2 and partitioning 2 is ***coarser*** than partitioning 1.

Examples of neutral partitionings are:

Definition 2.1.5. The ***trivial neutral partitioning*** of a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ is that defined by the equivalence relation: $x \equiv y \Leftrightarrow x = y$; i.e. the neutral networks of this partitioning comprise single sequences. The trivial neutral partitioning is minimal with respect to the partial ordering of neutral partitionings.

Definition 2.1.6. The ***maximal neutral partitioning*** of a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ is that defined by the equivalence relation: $x \equiv y \Leftrightarrow f(x) = f(y)$. The neutral networks of this partitioning are defined to be the ***maximal neutral networks*** of $\mathcal{L}$. The maximal neutral partitioning is maximal with respect to the partial ordering of neutral partitionings.

Definition 2.1.7. The ***maximal connected neutral partitioning*** of a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ is that defined by the equivalence relation: $x \equiv y \Leftrightarrow x, y$ are connected with respect to the Hamming graph structure on $\mathcal{A}^L$.

Definition 2.1.8. In the Introduction we described a fitness landscape loosely as a genotype $\rightarrow$ fitness mapping. Often, as in the case of the example presented in the Introduction (and indeed in natural evolution) there is an obvious *phenotype* and the genotype $\rightarrow$ fitness mapping takes the form of: genotype $\rightarrow$ phenotype $\rightarrow$ fitness. Since we are primarily interested in the genotype $\rightarrow$ fitness mapping we shall not, in general, allude to phenotypes. However, if there is a phenotype, the pre-images of the genotype $\rightarrow$ phenotype mapping define a neutral partitioning, which we refer to as a ***phenotypic neutral partitioning***.

We remark that the "network" terminology might often appear to be inappropriate, insofar as one's intuitive notion of "network" implies some degree of connectivity. Thus, for example, there is no reason to suppose in general that the maximal neutral "networks" of a fitness landscape are likely to be connected in the graph-theoretical sense; indeed, "neutral subspace" might appear to be a safer term. Nevertheless we shall adhere to the "network" terminology, firstly for historical reasons (the original terminology arose within the context of RNA secondary-structure folding landscapes, where the neutral networks do, in fact, posses a high degree of connectivity (Schuster et al., 1989; Grüner et al., 1996) and secondly because connectivity with respect to the Hamming structure *per se* will not necessarily be relevant to our analysis of evolutionary dynamics. When relevant we shall refer to *connected components* to denote the maximally connected sub-graphs of a neutral network with respect to its (inherited) Hamming graph structure.

2.2 Fitness-Independent Structure

Given a neutral partitioning of a fitness landscape, we shall call statistical properties dependent only on the partitioning as opposed to actual fitness values ***fitness-independent***; although our definition of a neutral partitioning pre-supposes a fitness function, all results in this Section hold unchanged for *any* partitioning of sequence space into equivalence classes.

2.2.1 Mutation Modes and Mutation Operators

As mentioned in the Introduction, the primary genetic operator will be *(random) mutation*. Given a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ and a sequence $x \in \mathcal{A}^L$ a sequence $y \in \mathcal{A}^L$ is said to be a ***point***

mutation of x if it differs from x at a single locus; i.e. $h(x,y) = 1$ where $h(\cdot,\cdot)$ is Hamming distance on $\mathcal{A}^L$. We wish to consider a "general" mutation as comprising a number of "random" point mutations.

Definition 2.2.1. A ***mutation mode*** is a random variable $\mathcal{U}$ taking values in $\{0,1,\ldots,L\}$. $u_n = \mathbf{P}(\mathcal{U} = n)$ is to be considered the probability that, during a mutation event, exactly n loci, selected uniformly at random from the $\binom{L}{n}$ possible choices of n loci, undergo point mutations. The ***per-sequence mutation rate*** for the mutation mode $\mathcal{U}$ is simply the expected number of point mutations, $\bar{u} = \mathbf{E}(\mathcal{U}) = \sum_{n=1}^{L} n \cdot u_n$.

Note that our definition of mutation is *independent of locus*: whatever the mutation mode, the probability that a point mutation occur at a locus during a mutation event will be the same for every locus. We remark that in the GA literature it is perhaps rare to encounter a mutation operator for which this is not the case[2]. In the absence of specific knowledge to the contrary (e.g. that optimisation may benefit from maintaining different mutation rates across the sequence) there seems little motivation to allow bias. It is conceivable, however, that during the course of optimisation one might detect that mutation at specific loci are particularly beneficial/detrimental and one might then construct an adaptive scheme to exploit this knowledge. In this thesis we shall always use locus-independent mutation as described in Def. 2.2.1.

Some examples of mutation modes are:

Poisson (or binomial) mutation: Here $\mathcal{U}$ has the *binomial* distribution:

$$u_n = \binom{L}{n} \mu^n (1-\mu)^{L-n} \tag{2.2}$$

for some $0 \leq \mu \leq 1$, so that $\bar{u} = L\mu$. We may think of this mutation mode arising from each locus of a sequence independently undergoing a point mutation with probability μ. We call μ the ***per-locus*** or ***point*** mutation rate.

In the *long sequence length limit* $L \to \infty$, keeping the per-sequence mutation rate $\bar{u} = L\mu$ constant, the mutation probabilities tend towards the *Poisson distribution*:

$$u_n \to e^{-\bar{u}} \frac{\bar{u}^n}{n!} \tag{2.3}$$

Although in practice the sequence lengths we shall encounter are of course finite, they are generally long enough that Eq. (2.3) is a good approximation and although in fact the number of mutations has in reality a binomial distribution Eq. (2.2), by abuse of language we shall still frequently refer to "Poisson" mutation[3].

Constant or n-point mutation: Here $u_k = \delta_{k,n}$ for some $0 \leq n \leq L$. That is, precisely n (uniform randomly selected) loci undergo point mutation (a.s.). We have $\bar{u} = n$.

Completely random mutation: This is simply binomial mutation with per-locus mutation rate $\mu = \frac{|\mathcal{A}|-1}{|\mathcal{A}|}$. Thus after completely random mutation the allele at any particular locus will be any $a \in \mathcal{A}$ with equal probability $\frac{1}{|\mathcal{A}|}$ - the sequence is effectively "randomised".

Trivial mutation: This is simply 0-point mutation; i.e. no point mutation occurs (a.s.).

[2] In biological evolution this may *not* necessarily be true: mutation rates may be different at different loci.

[3] We might also remark that when *simulating* mutation, it is generally computationally cheaper to compute Poisson than binomial deviates when the sequence length is reasonably long.

Given a mutation mode $\mathcal{U}$ with $\mathbf{P}(\mathcal{U}=n)=u_n$ and a sequence $x\in\mathcal{A}^L$ we now define the random variable $\mathcal{U}(x)$ with values in $\mathcal{A}^L$ by:

$$\mathbf{P}(\mathcal{U}(x)=y)=\left[\binom{L}{n}(|\mathcal{A}|-1)^n\right]^{-1}u_n \tag{2.4}$$

for any $y\in\mathcal{A}^L$ with $h(x,y)=k$. The random variable $\mathcal{U}(x)$ should be thought of as "the sequence x mutated using mutation mode $\mathcal{U}$". Note that $\binom{L}{n}(|\mathcal{A}|-1)^n$ is simply the number of sequences Hamming distance n from a given sequence. Eq. (2.4) thus says that given $x\in\mathcal{A}^L$ there is a probability $u_n=\mathbf{P}(\mathcal{U}=n)$ of mutating to a sequence Hamming distance n away from x - i.e. of n point mutations occurring - and that there is a *uniform* probability of mutating to any such sequence.

Now we want to admit the situation where different sequences may mutate according to different mutation modes. We thus define:

Definition 2.2.2. A ***mutation operator*** is a mapping U which assigns to each $x\in\mathcal{A}^L$ a mutation mode $\mathcal{U}_x$. Given a mutation operator $U:\ x\mapsto\mathcal{U}_x$ we may define for each $x\in\mathcal{A}^L$ the random variable $U(x)$ taking values in the sequence space $\mathcal{A}^L$ to be simply $\mathcal{U}_x(x)$ - i.e. x mutated "by its own mutation mode $\mathcal{U}_x$". By abuse of language we shall also refer to the mapping $x\mapsto U(x)$ as a mutation operator.

If $\mathcal{U}_x$ is the same for every $x\in\mathcal{A}^L$ - i.e. there is some mutation mode $\mathcal{U}$ such that $\mathcal{U}_x=\mathcal{U}$ and thus $U(x)=\mathcal{U}(x)\ \forall x\in\mathcal{A}^L$ - we call the mutation operator $U:\ x\mapsto\mathcal{U}$ ***uniform***. In this case every sequence mutates according to the same mutation mode.

Given a neutral partitioning $\mathcal{A}^L=\bigcup_{i=1}^N\Gamma_i$ we say that the mutation operator $U:\ x\mapsto\mathcal{U}_x$ is ***compatible*** with the partitioning iff $x\equiv y\Rightarrow\mathcal{U}_x=\mathcal{U}_y$ - i.e. the mutation mode is the same for all sequences in a given neutral network. We then have for $i=1,2,\ldots,N$ a well-defined mutation mode $\mathcal{U}_i$. If we have a neutral partitioning we shall generally assume, unless stated otherwise, that a mutation operator is compatible with the given partitioning. The motivation for and implications of this assumption will be discussed in the next Chapter[4].

Note that for a *uniform* mutation operator mutation is *symmetric*, in the sense that $\forall x,y$ we have:

$$\mathbf{P}(U(x)=y)=\mathbf{P}(U(y)=x) \tag{2.5}$$

This may be seen immediately from Eq. (2.4). Some additional notation will be required. Let X be a random variable taking values in $\mathcal{A}^L$ and U a mutation operator. Define the random variable $U(X)$, jointly distributed with X, by:

$$\mathbf{P}(U(X)=y\mid X=x)=\mathbf{P}(U(x)=y) \tag{2.6}$$

$U(X)$ can be thought of as the result of mutating the "random sequence" X using the mutation operator U. We shall frequently use $U(X)$ where X is a *uniform* random variable on $\mathcal{A}^L$. Note that as an immediate corollary of Eq. (2.5) we have that if X is uniform and U is uniform then $U(X)$ is also a uniform random variable on $\mathcal{A}^L$.

[4] In much of what follows it is not strictly necessary that U be compatible with the neutral partitioning. Nonetheless we generally restrict ourselves to this case.

Given a neutral partitioning and a mutation operator U we note that for $x \in \mathcal{A}^L$ we can consider $\widetilde{U(x)}$ as a random variable taking values in the set of equivalence classes (or equivalently the set of indices $i = 1, 2, \ldots, N$) of the partitioning. For notational convenience we write $\widetilde{U}(x) = \widetilde{U(x)}$ and similarly for a random variable X taking values in $\mathcal{A}^L$ we write $\widetilde{U}(X) = \widetilde{U(X)}$.

2.2.2 The Mutation Matrix

Suppose we are given a neutral partitioning of a fitness landscape. We wish to consider (for reasons that will become clearer in the next Chapter) the probability that a sequence selected uniformly at random from one neutral network mutates to another neutral network. We thus define:

Definition 2.2.3. Given a neutral partitioning $\mathcal{A}^L = \bigcup_{i=1}^N \Gamma_i$, and a (compatible) mutation operator U, we define the ***mutation matrix*** (for the given partitioning and mutation operator) to be:

$$m_{ij}(U) = \mathbf{P}\left(U(X) \in \Gamma_i \mid X \in \Gamma_j\right) = \mathbf{P}\left(\widetilde{U}(X) = i \;\middle|\; \widetilde{X} = j\right) \tag{2.7}$$

for $i, j = 1, 2, \ldots, N$, where X is a *uniform* random variable on $\mathcal{A}^L$. In matrix notation we write $\boldsymbol{m}(U)$ for the matrix with entries $m_{ij}(U)$. Note that it is a *stochastic* matrix. We also define the ***neutrality*** of Γ_i with respect to U to be:

$$\nu_i(U) = m_{ii}(U) = \mathbf{P}\left(U(X) \in \Gamma_i \mid X \in \Gamma_i\right) = \mathbf{P}\left(\widetilde{U}(X) = i \;\middle|\; \widetilde{X} = i\right) \tag{2.8}$$

We should think of $m_{ij}(U)$ as the probability that a sequence picked uniformly at random from neutral network Γ_j ends up in Γ_i after mutation (note the order of indices).

Now given any (compatible) mutation operator U we can build its mutation matrix from the uniform mutation matrices of the constant mutation operators. Let $\boldsymbol{m}^{(n)} = \boldsymbol{m}\left(U^{(n)}\right)$ where $U^{(n)}$ is the (unique) uniform mutation operator for the constant mutation mode with rate n. Then we have, in matrix notation:

$$m_{ij}(U) = \sum_{n=0}^{L} u_{j,n} \cdot m_{ij}^{(n)} \tag{2.9}$$

with:

$$u_{j,n} = \mathbf{P}\left(\mathcal{U}_j = n\right) \tag{2.10}$$

where (recalling that the mutation operator U is compatible with the neutral partitioning) $\mathcal{U}_j$ is the mutation mode for U on Γ_j. In this sense the $\boldsymbol{m}^{(n)}$ for $n = 0, 1, \ldots, L$ define the mutation structure of the partitioning: if we know the $\boldsymbol{m}^{(n)}$ and the mutation modes $\mathcal{U}_j$ then we know the mutation matrix $\boldsymbol{m}(U)$. Note that $\boldsymbol{m}^{(0)}$ is just the $N \times N$ identity matrix.

Another quantity of interest is the ***relative volume*** $\upsilon_i = |\mathcal{A}|^{-L}|\Gamma_i|$ of the neutral networks Γ_i. We note that this can be expressed in terms of any (non-trivial) *uniform* mutation operator U. To see this, note that if X is uniform on $\mathcal{A}^L$ then $\upsilon_j = \mathbf{P}\left(X \in \Gamma_j\right)$ and:

$$\begin{aligned}
\upsilon_i &= \mathbf{P}\left(U(X) \in \Gamma_i\right) \qquad \text{since } U(X) \text{ is uniform} \\
&= \sum_j \mathbf{P}\left(U(X) \in \Gamma_i \mid X \in \Gamma_j\right)\, \mathbf{P}\left(X \in \Gamma_j\right) \\
&= \sum_j m_{ij}(U)\, \mathbf{P}\left(X \in \Gamma_j\right) \\
&= \sum_j m_{ij}(U)\, \upsilon_j \qquad \text{since } X \text{ is uniform}
\end{aligned}$$

or, in matrix notation:

$$\boldsymbol{m}(U)\cdot\upsilon=\upsilon \tag{2.11}$$

where υ is the (column) vector with components υ_i. Recalling that $\boldsymbol{m}(U)$ is stochastic and is by assumption non-trivial, we find that υ is the (unique) *eigenvector* of $\boldsymbol{m}(U)$ with $|\upsilon|=\sum_{i=1}^{N}\upsilon_i=1$, with (principal) eigenvalue 1 (Gantmacher, 1959; Seneta, 1973). This holds in particular for $U=U^{(n)}$ for $n=1,2,\ldots,L$ and we could in principal calculate υ from, say, $\boldsymbol{m}^{(1)}=\boldsymbol{m}\left(U^{(1)}\right)$.

2.2.3 Subspace Entropy and Markov Partitionings

Anticipating the next section somewhat, we will be dealing with Markov processes defined on the sequence space $\mathcal{A}^L$ which depend on a mutation operator and which, in a sense to be made precise, "respect" the fitness structure of our fitness landscape. Now given a neutral partitioning of our landscape, such a Markov process on $\mathcal{A}^L$ induces naturally a stochastic process on the set of neutral networks - this induced process is *not*, however, necessarily Markovian; this is because the transition probabilities don't necessarily "respect" the *mutation* structure of the partitioning. We will, however, *approximate* the induced process by a Markov process, defined by application of a *maximum entropy assumption* - essentially we assume that any sequence behaves like a sequence drawn uniformly at random from the neutral network to which it belongs. The extent to which this process models the original process accurately depends on the degree to which this maximum entropy assumption holds. Here we present a measure of the extent to which the maximum entropy approximation might be expected to model a Markov process.

Thus suppose given a neutral partitioning $\mathcal{A}^L=\bigcup_{i=1}^{N}\Gamma_i$ and a (compatible) mutation operator U. We want to make precise the statement that, given an arbitrary sequence, the neutral network it ends up in after mutation does not depend on the particular sequence but only on the neutral network it came from. We can express this in information-theoretic terms as follows: let X be a uniform random variable on $\mathcal{A}^L$. X then mutates under U to the sequence $U(X)$, which belongs to the neutral network $\widetilde{U}(X)$. We would like to say then, that *knowing the actual sequence X gives no further information about $\widetilde{U}(X)$ than merely knowing $\widetilde{X}$, the network to which X belongs*. This motivates the following definition:

Definition 2.2.4. Given a neutral partitioning $\mathcal{A}^L=\bigcup_{i=1}^{N}\Gamma_i$ and a compatible mutation operator U let Γ_j be a neutral network and let the random variable X_j be uniform on Γ_j. We define the ***entropy*** of Γ_j with respect to U to be:

$$H_j(U)=\mathbf{H}\left(\widetilde{U}(X_j)\right) \tag{2.12}$$

To calculate $H_j(U)$ note that if X_j is uniform on Γ_j then $\mathbf{P}\left(\widetilde{U}(X_j)=i\right)=m_{ij}(U)$, so that:

$$H_j(U)=-\sum_{i=1}^{N}m_{ij}(U)\log_2\left(m_{ij}(U)\right) \tag{2.13}$$

We note that the entropy of a neutral network is constrained by its neutrality. In particular, it is easy to show that if $\nu=\nu_j(U)$ then we have:

$$h(\nu)\leq H_j(U)\leq h(\nu)+(1-\nu)\log_2(N-1) \tag{2.14}$$

where $h(p) = -p\log_2 p - (1-p)\log_2(1-p)$ is the entropy of a Bernoulli trial (biased coin toss) with probability $0 \leq p \leq 1$. Since $0 \leq h(p) \leq 1$ we see that if the number of neutral networks N is reasonably large then maximum possible entropy of a neutral network with neutrality ν (with respect to some mutation operator) is $\approx (1-\nu)\log_2 N$. Essentially then, increasing neutrality reduces the (possible) uncertainty as to which neutral network a mutation is likely to take us to. If we wish to "factor out" the effects of neutrality on entropy, we may consider the entropy of $\widetilde{U}(X_j)$ *given that the mutation is non-neutral*; i.e. conditional on $\widetilde{U}(X_j) \neq j$. We thus define the ***neutral-adjusted*** entropy to be:

$$H'_j(U) = \frac{H_j(U) - h(\nu)}{1-\nu} \tag{2.15}$$

so that:

$$0 \leq H'_j(U) \leq \log_2(N-1) \tag{2.16}$$

Definition 2.2.5. Given a neutral partitioning $\mathcal{A}^L = \bigcup_{i=1}^N \Gamma_i$ let U and X_j be as above. We define the ***Markov coefficient*** of Γ_j with respect to $\mathcal{U}$ to be:

$$\mathcal{M}_j(U) = \mathbf{H}\left(\widetilde{U}(X_j)\right) - \mathbf{H}\left(\widetilde{U}(X_j) \mid X_j\right) \tag{2.17}$$

We note that $\mathcal{M}_j(U)$ is always ≥ 0; we say that Γ_j is ***Markov*** with respect to U iff:

$$\mathcal{M}_j(U) = 0 \tag{2.18}$$

We can interpret $\mathcal{M}_j(U)$, the *mutual information* between X_j and $\widetilde{U}(X_j)$, as "the information about $\widetilde{U}(X_j)$ gained by knowing the actual value of X_j". Thus the vanishing of $\mathcal{M}_j(U)$ means that knowing the *particular* $x \in \Gamma_j$ tells us no more about the neutral network to which x is likely to mutate to under U than merely knowing that x is (uniformly randomly) selected from Γ_j. To calculate $\mathcal{M}_j(U)$, for $x \in \mathcal{A}^L$ let us set:

$$m_i(x,U) = \mathbf{P}\left(\widetilde{U}(x) = i\right) \tag{2.19}$$

i.e. $m_i(x,U)$ is the probability that x mutates to neutral network Γ_i. Let us set:

$$H(x,U) = \mathbf{H}\left(\widetilde{U}(x)\right) = -\sum_{i=1}^{N} m_i(x,U)\log_2(m_i(x,U)) \tag{2.20}$$

We then find that:

$$\mathcal{M}_j(U) = H_j(U) - \frac{1}{|\Gamma_j|}\sum_{x\in\Gamma_j} H(x,U) \tag{2.21}$$

We can write this compactly in terms of a random variable X_j uniform on Γ_j as:

$$\mathcal{M}_j(U) = H_j(U) - \mathbf{E}\left(H(X_j,U)\right) \tag{2.22}$$

which says intuitively that the Markov coefficient of Γ_j is "*the uncertainty of* $\widetilde{U}(X_j)$ *less the mean uncertainty of* $\widetilde{U}(x)$ *averaged over* $x \in \Gamma_j$". In the next Chapter we will take $\mathcal{M}_j(U)$ as a measure of how well the maximum entropy assumption is likely to work for a given neutral network Γ_j.

We can also introduce a *global* measure as follows:

Definition 2.2.6. The ***(global) Markov coefficient*** of a partitioning with respect to the compatible mutation operator U is defined to be:

$$\mathcal{M}(U) = \mathbf{H}\left(\widetilde{U}(X) \mid \widetilde{X}\right) - \mathbf{H}\left(\widetilde{U}(X) \mid X\right) \tag{2.23}$$

where X is uniform random on $\mathcal{A}^L$. Again $\mathcal{M}(U) \geq 0$ and we define the partitioning to be ***Markov*** with respect to U iff:

$$\mathcal{M}(U) = 0 \tag{2.24}$$

We interpret $\mathcal{M}(U)$ as "the information about $\widetilde{U}(X)$ conveyed by X given $\widetilde{X}$" and the vanishing of $\mathcal{M}(U)$ means that knowing $x \in \mathcal{A}^L$ tells us no more about the neutral network to which it is likely to mutate to than merely knowing the neutral network to which x belongs. A straightforward calculation shows that:

$$\mathcal{M}(U) = \sum_{j=1}^{N} \upsilon_i \mathcal{M}_j(U) \tag{2.25}$$

where υ_i is the relative volume of Γ_i. We can write this compactly in terms of a random variable X uniform on $\mathcal{A}^L$ as:

$$\mathcal{M}(U) = \mathbf{E}\left(\mathcal{M}_{\widetilde{X}}(U)\right) \tag{2.26}$$

so $\mathcal{M}(\mathbf{U})$ can be interpreted as the "*mean of the Markov coefficients of the neutral networks, weighted by relative volume*". In particular a partitioning is Markov if and only if all its neutral networks are Markov.

2.2.4 Multiplicative Mutation Approximations

Given two mutation modes $\mathcal{U}$ and $\mathcal{U}'$ we can define the mutation mode $\mathcal{U}\mathcal{U}'$ by ***composition*** which, intuitively, signifies the application of $\mathcal{U}'$ followed by $\mathcal{U}$. It is clear that this gives a new mutation mode. In principle we can calculate combinatorially the probabilities $\mathbf{P}(\mathcal{U}\mathcal{U}' = n)$ in terms of the $u_n = \mathbf{P}(\mathcal{U} = n)$ and $u'_n = \mathbf{P}(\mathcal{U}' = n)$; this is not quite straightforward in general, as we must take into account the probabilities that several loci may be hit by *two* point mutations and that if this happens the net result may be *no* point mutation at that locus. A simple example is when $\mathcal{U}$ is binomial with per-locus rate μ and $\mathcal{U}'$ is binomial with per-locus rate μ'. It is then easy to calculate that $\mathcal{U}\mathcal{U}'$ is binomial with per-locus rate

$$\mu + \mu' - \frac{|\mathcal{A}|}{|\mathcal{A}| - 1}\mu\mu' \tag{2.27}$$

Suffice to note that if the $u_n = \mathbf{P}(\mathcal{U} = n)$ and $u'_n = \mathbf{P}(\mathcal{U}' = n)$ are small enough to be negligible unless $n \ll L$ we can ignore the probability that successive point mutations might occur at the same locus and we have the approximation:

$$\mathbf{P}\left(\mathcal{U}\mathcal{U}' = n\right) \approx \sum_{r+s=n} u_r u'_s = \mathbf{P}\left(\mathcal{U} + \mathcal{U}' = n\right) \tag{2.28}$$

or simply:

$$\mathcal{U}\mathcal{U}' \approx \mathcal{U} + \mathcal{U}' \tag{2.29}$$

recalling that a mutation mode is simply a (non-negative) integer-valued random variable.

It is clear that in general composition is ***commutative*** (i.e. $\mathcal{U}\mathcal{U}' = \mathcal{U}'\mathcal{U}$ holds $\forall \mathcal{U}, \mathcal{U}'$) and that the trivial mutation mode acts as an identity element. The set of mutation modes under composition thus has the algebraic structure of a ***commutative semi-group***.

We now turn to mutation *operators*. If U, U' are two mutation operators we would like to define a mutation operator UU' by:

$$\mathbf{P}\left((UU')(x) = y\right) = \mathbf{P}\left(U(U'(x)) = y\right) \tag{2.30}$$

for $x, y \in \mathcal{A}^L$ to denote mutation by U' followed by mutation by U. In general, this will not yield a mutation operator in the sense that we have defined it; for mutation operators have the property that given $x \in \mathcal{A}^L$ there is an equal probability that x mutates to any sequence a given Hamming distance away. This will not necessarily be the case for the operator defined by Eq. (2.30), as we may easily convince ourselves with a simple example for $L = 2$: let U' have the property that $U'(00)$ is the constant mutation mode with rate 1 and let U have the property that $U(01)$ is constant with rate 0 and $U(10)$ is constant with rate 2. Applying U' to the sequence 00 we have thus an equal probability of $\frac{1}{2}$ of mutating to either 01 or 10. Now applying U to 01 is trivial and thus leaves us at 01, while applying U to 10 always takes us to 01. Thus the probability that applying UU' to 00 takes us to 01 is 1, while the probability that it takes us to 10 is 0 even though 01 and 10 are both Hamming distance 1 from 00. We remark, however, that if U, U' are *uniform* mutation operators - i.e. the same mutation mode applies at every sequence - then Eq. (2.30) does indeed yield a (uniform) mutation operator.

Now suppose a neutral partitioning $\mathcal{A}^L = \bigcup_{i=1}^N \Gamma_i$ given and that U, U' are mutation operators compatible with the partitioning. Then even if UU' is not a mutation operator we may still define a mutation matrix $\boldsymbol{m}(UU')$ by Eq. (2.7). We then have:

$$m_{ij}(UU') = \sum_{k=1}^{N} \mathbf{P}\left(U(U'(X_j)) \in \Gamma_i \;\middle|\; U'(X_j) \in \Gamma_k\right) \cdot m_{kj}(U') \tag{2.31}$$

where X_j is uniform on Γ_j. Now the problem is that, given $U'(X_j) \in \Gamma_k$, the random sequence $U'(X_j)$ is not necessarily uniform on Γ_k. However, if Γ_k is *Markov* with respect to U - i.e. $\mathcal{M}_k(U) = 0$ - then it is easy to see that $\mathbf{P}\left(U(U'(X_j)) \in \Gamma_i \;\middle|\; U'(X_j) \in \Gamma_k\right) = \mathbf{P}\left(U(X_k) \in \Gamma_i\right) = m_{ik}(U)$ where X_k is *uniform* on Γ_k. We thus have in particular:

Propostion 2.2.1. *If the partitioning is Markov with respect to U then:*

$$\mathbf{m}(UU') = \mathbf{m}(U) \cdot \mathbf{m}(U') \tag{2.32}$$

In general Eq. (2.32) will not hold exactly - we cannot expect that a sequence arriving in Γ_k by mutation from Γ_j will be uniformly distributed on Γ_k. However, we note that the condition for Prop. 2.2.1 to obtain - that the partitioning be Markov with respect to U - is, as we argue in the next Chapter, the same as that required for our maximum entropy approximation to hold. So if we are going to assume that a maximum entropy approximation is acceptably accurate for a given neutral partitioning and mutation operator(s) we may as well assume also that the approximation:

$$\boldsymbol{m}(UU') \approx \boldsymbol{m}(U) \cdot \boldsymbol{m}(U') \tag{2.33}$$

is acceptable, in the sense that the approximation is likely to be as good as the maximum entropy approximation itself. We shall callEq. (2.33) the ***weak multiplicative mutation approximation***.

The (global) Markov coefficient $\mathcal{M}(U)$ is thus likely to be a suitable indicator as to how well this approximation might be expected to obtain.

Now from Eq. (2.29) we have for the uniform constant mutation operators $U^{(n)}$ with $n \ll L$, that:

$$U^{(n)} \approx \left(U^{(1)}\right)^n \tag{2.34}$$

With Eq. (2.33) this gives:

$$\boldsymbol{m}^{(n)} \approx \boldsymbol{m}^n \tag{2.35}$$

where we drop the superscript and write $\boldsymbol{m} = \boldsymbol{m}^{(1)}$. Under this approximation we have from Eq. (2.9), for a general mutation operator U that:

$$m_{ij}(U) \approx \sum_{n=0}^{L} u_{j,n}\ (\boldsymbol{m}^n)_{ij} \tag{2.36}$$

where again $u_{j,n} = \mathbf{P}(\mathcal{U}_j = n)$. We shall call Eq. (2.36) the ***strong multiplicative mutation approximation***; it provides a simple expression for calculating the mutation matrix for a general (compatible) mutation operator in terms of the uniform constant 1-point mutation matrix $\boldsymbol{m}$. This approximation might be expected to be acceptable if:

1. the Markov coefficient of the neutral partitioning with respect to $\mathcal{M}(U)$ is small
2. mutation rates are low, in the sense that $\mathbf{P}(\mathcal{U}_j = n)$ is small unless $n \ll L$.

In particular, if the mutation operator is *uniform* (i.e. $\mathcal{U}_j = \mathcal{U}$, say, $\forall j$) we have simply, in matrix notation:

$$\boldsymbol{m}(U) \approx \sum_{n=0}^{L} u_n\, \boldsymbol{m}^n \tag{2.37}$$

where $u_n = \mathbf{P}(\mathcal{U} = n)$. For instance, for $\mathcal{U}$ binomial with per-locus rate μ we have:

$$\boldsymbol{m}(U) \approx [(1-\mu)\mathbf{1} + \mu \cdot \boldsymbol{m}]^L \tag{2.38}$$

where $\mathbf{1}$ is the $N \times N$ identity matrix, while Poisson mutation with per-sequence rate $\bar{u}$ gives:

$$\boldsymbol{m}(U) \approx e^{-\bar{u}(\mathbf{1}-\boldsymbol{m})} \tag{2.39}$$

We note that even if Eq. (2.29) does not hold - i.e. mutation rates may be high - we can still, under the weak multiplicative mutation assumption Eq. (2.33), calculate an approximation to $\boldsymbol{m}^{(n)}$ in terms of $\boldsymbol{m}^n$ as follows: note that by Eq. (2.33), $\boldsymbol{m}\left(\left(U^{(1)}\right)^n\right) \approx \boldsymbol{m}\left(\left(U^{(1)}\right)\right)^n = \boldsymbol{m}^n$. Now $\left(U^{(1)}\right)^n$ amounts to performing n consecutive, independent point mutations. Let us set:

$$P_{n,k} = \mathbf{P}(k \text{ actual point mutations occur in } n \text{ consecutive independent point mutations}) \tag{2.40}$$

Then we have:

$$\boldsymbol{m}^n \approx \sum_{k=0}^{n} P_{n,k}\, \boldsymbol{m}^{(k)} \tag{2.41}$$

We may derive the recursion relation:

$$P_{n,k} = \left(1 - \frac{k-1}{L}\right) P_{n-1,k-1} + (1-a)\frac{k}{L} P_{n-1,k} + a\frac{k+1}{L} P_{n-1,k+1} \tag{2.42}$$

where $a = \frac{1}{|\mathcal{A}|-1}$ is the probability that a locus that has undergone point mutation reverts to its original allele after a further point mutation. As an example, for $n = 2$ we have immediately that $\boldsymbol{m}^{(0)} = \mathbf{1}$, $\boldsymbol{m}^{(1)} = \boldsymbol{m}$ and from Eq. (2.41)

$$\boldsymbol{m}^2 \approx P_{2,0}\mathbf{1} + P_{2,1}\boldsymbol{m} + P_{2,2}\boldsymbol{m}^{(2)} \tag{2.43}$$

Using Eq. (2.42) we may calculate that $P_{2,0} = a\frac{1}{L}$, $P_{2,1} = (1-a)\frac{1}{L}$ and $P_{2,2} = 1 - \frac{1}{L}$, yielding:

$$\boldsymbol{m}^{(2)} \approx \frac{L}{L-1}\boldsymbol{m}^2 - (1-a)\frac{1}{L-1}\boldsymbol{m} - a\frac{1}{L-1}\mathbf{1} \tag{2.44}$$

Note that for the *binary* alphabet $a = 1$ and we have simply:

$$\boldsymbol{m}^{(2)} \approx \frac{L}{L-1}\boldsymbol{m}^2 - \frac{1}{L-1}\mathbf{1} \tag{2.45}$$

In general the mutation rates encountered in this thesis will be reasonably low, and we will frequently adopt the strong multiplicative mutation approximation Eq. (2.36). This has the particular advantage that the one point mutation matrix $\boldsymbol{m}$ now encapsulates *all* the mutational structure information for the landscape under the given neutral partitioning, in the sense that we may construct any mutation matrix $\boldsymbol{m}(U)$ from $\boldsymbol{m}$ via Eq. (2.36).

2.2.5 Optimal Mutation

Suppose we are given a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ and a neutral partitioning $\mathcal{A}^L = \bigcup_{i=1}^{N} \Gamma_i$. Given neutral networks Γ_i, Γ_j we may ask: is there a mutation mode/rate which maximises the probability that a (uniform randomly selected) sequence in Γ_j mutates to Γ_i? To answer this question, we note that for a mutation mode $\mathcal{U}$ and X_j uniform on Γ_j:

$$\mathbf{P}(\mathcal{U}(X_j) \in \Gamma_i) = \sum_{n=0}^{L} u_n \, \boldsymbol{m}_{ij}^{(n)} \tag{2.46}$$

where $\boldsymbol{m}^{(n)}$ is the mutation matrix for uniform constant mutation with rate n and we have set $u_n = \mathbf{P}(\mathcal{U} = n)$. We can thus ask how to choose the u_n (for given i, j) so as to *maximise* the RHS of Eq. (2.46). Now considering the $\boldsymbol{m}_{ij}^{(n)}$ as fixed constants, Eq. (2.46) is a *linear* function of the u_n over the simplex described by the constraint $\sum_{n=0}^{L} u_n = 1$. It is thus clear that, barring any "degeneracies" among the coefficients $\boldsymbol{m}_{ij}^{(n)}$, the maximum of this linear function must lie over a vertex of the simplex; i.e. a point where all the u_n are zero except for one value of n, for which $u_n = 1$. Thus we have:

Propostion 2.2.2. *Given a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$, a neutral partitioning $\mathcal{A}^L = \bigcup_{i=1}^{N} \Gamma_i$ and neutral network Γ_i, Γ_j then the mutation mode which maximises the probability that a (uniform randomly selected) sequence in Γ_j mutates to Γ_i is n-point (constant) mutation, with rate n equal to the value which maximises $\mathbf{m}_{ij}^{(n)}$ if a unique such n exists.*

We note that if there is no *unique* n maximising $\boldsymbol{m}_{ij}^{(n)}$ - if for example $n_1, n_2, \ldots, n_K$ all yield the same maximum value - then any mutation mode with $\mathbf{P}(\mathcal{U} = n) = 0$ unless $n = n_k$ for some k maximises $\mathbf{P}(\mathcal{U}(X_j) \in \Gamma_i)$. We note furthermore that if we accept the strong multiplicative mutation approximation given by Eq. (2.36), then the optimal mutation rate of Prop. 2.3.1 is that which maximises $\boldsymbol{m}_{ij}^{n}$ where $\boldsymbol{m} = \boldsymbol{m}^{(1)}$ is the 1-point constant mutation matrix.

2.2.6 Innovations, Accessibility and Percolation of Neutral Networks

In-depth analysis of RNA sequence to secondary structure mappings (Grüner et al., 1996; Huynen et al., 1996; Huynen, 1996; Reidys, Stadler, & Schuster, 1997) have demonstrated the following striking feature of the neutral networks with respect to what we have referred to as a *phenotypic neutral partitioning* - equivalence classes of sequences mapping to the same secondary structure:

> *Every "generic" neutral networks may be reached by a few point mutations from an arbitrary sequence.*

This feature is frequently described in terms of ***percolation***[5] of neutral networks through sequence space.

Constant Innovation

In (Huynen et al., 1996; Huynen, 1996) the implications of such percolation-like properties of neutral networks as regards evolutionary dynamics on fitness landscapes based on RNA secondary structure landscapes are investigated. There, *neutral walks* are performed on neutral networks of an RNA folding landscape. At each sequence encountered along such a walk, the number of hitherto unseen phenotypes within a Hamming distances of one or two of the current sequence is logged and the cumulative number of such ***innovations*** - hitherto unseen phenotypes - is plotted against the number of steps taken on the neutral walk. For "generic" neutral networks, the resulting plot remains approximately linear for lengthy walks, the implication being that in exploring a neutral network (e.g. by *neutral drift*) we may expect to encounter, at an approximately constant rate, an almost inexhaustible supply of novel (and potentially "fitter") phenotypes. The authors coin the suggestive term ***constant innovation*** to describe this characteristic property of neutral networks in RNA folding landscapes and go on to discuss the qualitative structure of population evolution on a landscape featuring such networks.

Here we take the view that any measure of percolation/innovation, if it is to be useful as an indicator of evolutionary dynamics, ought to relate to accessibility of neutral networks via *mutation*. We thus co-opt the term *innovation* to denote the discovery of a hitherto unseen neutral network (with respect to some neutral partitioning) by mutation (via some compatible mutation operator) during an instantiation of an evolutionary process (Chapter 3). To analyse the phenomenon of constant innovation, for a given neutral network we consider a stochastic process whereby at each step we select a sequence uniformly at random from our neutral network and mutate it via the extant mutation operator. We then log the cumulative number of innovations - neutral networks not previously discovered by mutation during the process - against number of time steps. Since a neutral walk of the *blind ant* variety asymptotically samples a neutral network uniformly (Hughes, 1996) (see also Chapters 3 and 4), we may consider this a (crude) approximation of the the procedure employed by Huynen *et al.*.

Thus let us suppose that given a neutral partitioning $\bigcup_{i=1}^{N} \Gamma_i$ of a landscape $\mathcal{L}$ and a compatible mutation operator U we perform the above process on neutral network Γ_j. At each time step, the probability that mutation of a sequence selected uniformly at random from Γ_j lands in Γ_i is (by

[5] Use of the term "percolation" in this context is intuitively appealing, but perhaps somewhat unfortunate in that it does *not* necessarily equate with the precise graph-theoretical definition of the term (Bollobás, 1985). We shall nevertheless follow this practice and use the term "percolation" in the looser sense intended by RNA researchers.

definition) just $m_{ij}(U)$. Let us define the r.v. $I_j(t)$ to be the (cumulative) number of innovations at time step t - i.e. after t selection/mutation events. We then have:

$$\begin{aligned}\mathbf{P}\left(I_j(t) = I_j(t-1)+1\right) &= \mathbf{P}\left(\text{innovation at time } t\right) \\ &= \sum_{i=1}^{N} \mathbf{P}\left(\text{mutation discovers } \Gamma_i \text{ for the first time at time } t\right) \\ &= \sum_{i=1}^{N} \left[1-m_{ij}(U)\right]^{t-1} m_{ij}(U)\end{aligned}$$

Since we must always have $I_j(1) = 1$, we may readily calculate in particular that:

$$\mathbf{E}\left(I_j(t)\right) = N - \sum_{i=1}^{N} \left[1-m_{ij}(U)\right]^t \tag{2.47}$$

Thus the expected number of neutral networks remaining *un*discovered decays in time as a sum of exponentials, corresponding to the times taken to discovery of each neutral network in the landscape. If $N_j(U)$ is the number of neutral networks "accessible" from Γ_j under the mutation operator U - i.e. those Γ_i for which $m_{ij}(U) > 0$ - then $\mathbf{E}\left(I_j(t)\right)$ approaches $N_j(U)$ asymptotically as $t \to \infty$ as we should expect. If, for instance, U is Poisson mutation on Γ_j then *every* neutral network may be reached from Γ_j by mutation with non-zero probability so that $N_j(U) = N$. Fig. 2.1 illustrates the idea with a minimal example of time-dependence of expected innovations from a neutral network with three accessible networks (i.e. $N_j(U) = 3$). Mutation probabilities (0.001, 0.1 and 0.899 respectively) vary in order of magnitude and the resultant differing time-scales of decay of numbers of undiscovered networks may be clearly seen.

Accessibility

From Eq. (2.47) we see that those neutral networks which are easily discovered by mutation - those Γ_i for which $m_{ij}(U)$ is large - contribute little to the expected number of innovations for large times. Now may calculate:

$$\frac{d}{dt}\mathbf{E}\left(I_j(t)\right) = -\sum_{i=1}^{N} \left[1-m_{ij}(U)\right]^t \log\left(1-m_{ij}(U)\right) \tag{2.48}$$

as a measure of "innovation rate" at time t. From this we see that in some sense easily accessible neutral networks also *lower the innovation rate*, as they are likely to be repeatedly *re*discovered (and thus not qualify as innovations). It is easily seen that, given that $N_j(U)$ neutral networks are accessible from Γ_j, then the innovation rate is always highest when the probability of discovering any particular of the $N_j(U)$ neutral networks is "evenly spread"; specifically, the innovation rate (for any time t) takes a maximum value when all the (non-zero) $m_{ij}(U)$ are equal to $1/N_j(U)$.

This strongly suggests that a useful measure of constant innovation/percolation for a neutral network might be the amount of *uncertainty* as to which network a sequence might mutate to from our network. In Section 2.2.3 we introduced the entropy $H_j(U)$ for neutral network Γ_j with respect to a mutation mode U. In fact we find it more intuitive to consider the quantity:

$$\mathcal{P}_j(U) = 2^{H_j(U)} \tag{2.49}$$

with $H_j(U)$ as given by Eq. (2.13), which we term the ***percolation index*** for neutral network Γ_j and propose as a measure for innovation/percolation. We see that if $N_j(U)$ neutral networks are

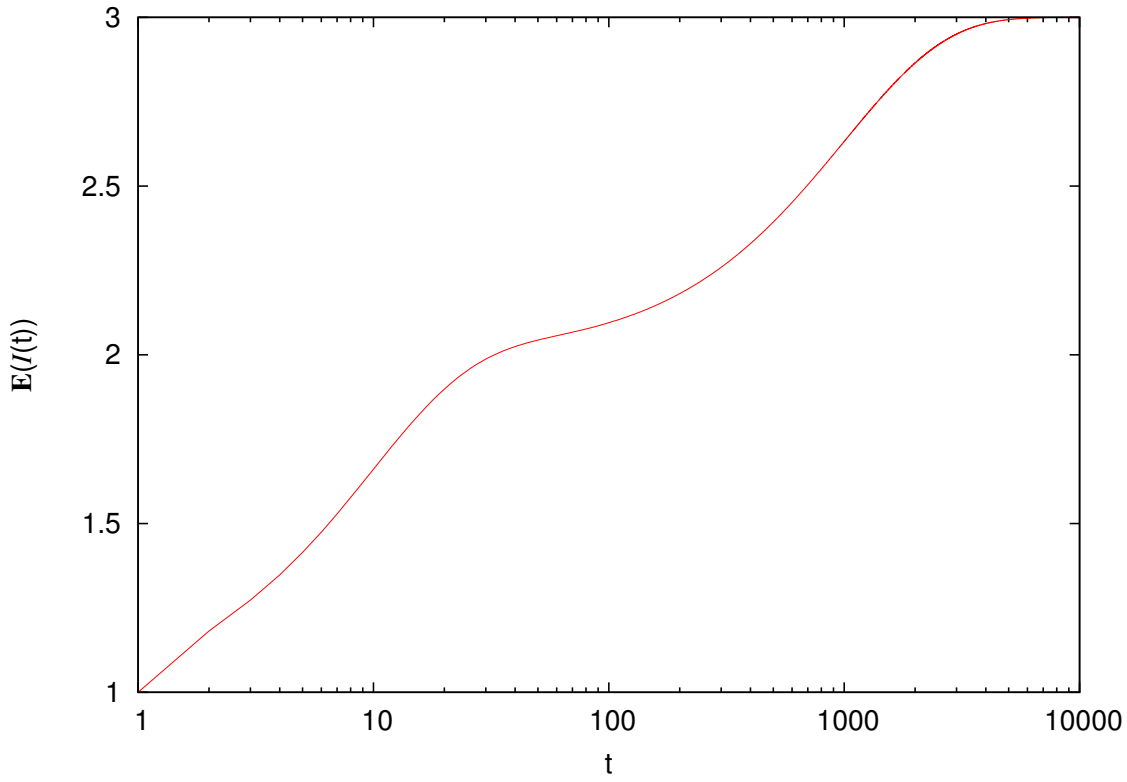

Figure 2.1: Expected (cumulative) innovations (Eq. (2.47)) plotted against time (logarithmic scale) for a neutral network with access to three networks. Mutation probabilities are 0.001, 0.1 and 0.899.

accessible from Γ_j via U then:

$$1 \leq \mathcal{P}_j(U) \leq N_j(U) \tag{2.50}$$

with $\mathcal{P}_j(U) = 1$ iff only one neutral network is accessible from Γ_j and $\mathcal{P}_j(U) = N_j(U)$ iff all $N_j(U)$ neutral networks are equally likely to be discovered by mutation from Γ_j. We might thus think of $\mathcal{P}_j(U)$ as an ***"effective number of accessible neutral networks"*** for Γ_j with respect to U. We also remark that $\mathcal{P}_j(U)$ may in some cases be approximated analytically (*cf.* Chapter 5) and is otherwise quite readily estimated in simulation (*cf.* Chapter 6).

Now if the neutrality of a network is high then (by definition) mutation will repeatedly "rediscover the network itself". If we know that $\nu = \nu_j(U)$ is the neutrality of Γ_j then we may refine the bounds on $\mathcal{P}_j(U)$ (Eq. 2.50) by:

$$1 \leq \mathcal{P}_j(U) \leq \nu^{-\nu} \left(\frac{N_j(U) - 1}{1 - \nu} \right)^{1-\nu} \tag{2.51}$$

where the percolation index achieves its maximum value iff all neutral networks *apart from* Γ_j *itself* are equally likely to be discovered. This confirms the intuitive suspicion that high neutrality might be expected to lower the innovation rate in the sense that a high proportion of discovery attempts will be "wasted" on neutral mutations[6]. For example if $\nu = \frac{1}{2}$ this yields:

$$1 \leq \mathcal{P}_j(U) \leq 2\sqrt{N_j(U) - 1} \tag{2.52}$$

To "factor out" this neutrality effect we might consider the ***neutral-adjusted percolation index***:

$$\mathcal{P}'_j(U) = 2^{H'_j(U)} \tag{2.53}$$

with $H'_j(U)$ as given by Eq. (2.15). We have then:

$$1 \leq \mathcal{P}'_j(U) \leq N_j(U) - 1 \tag{2.54}$$

with the maximum attained again iff all neutral networks apart from Γ_j itself are equally likely to be discovered.

For the example of Fig. 2.1 we find a percolation index $\mathcal{P} \approx 1.39$. If we were to take the network with highest probability of being "mutated to" to be the network itself - i.e. neutrality is $\nu = 0.899$ - then the upper bound on percolation would be ≈ 1.49. The neutral-adjusted percolation index in this case would be $\mathcal{P}' \approx 1.06$, actually a little lower (due to the small number of accessible networks) than unadjusted percolation in this (somewhat artificial) example.

Homogeneity and Drift

Another statistical feature of a neutral network is the degree to which accessibility of other networks varies from sequence to sequence across the network. This might be expected to relate to what we have termed the *Markov index* (Def. 2.2.5) of a neutral network - a measure of the degree to which knowing the precise sequence on a network disambiguates the possible networks to which that sequence might mutate. In the next Chapter we shall see that this has important consequences for the utility of *neutral drift*.

[6] At least if the number of accessible networks is reasonably large: the upper bound on $\mathcal{P}_j(U)$ from Eq. (2.51) will be $< N_j(U)$, the upper bound of Eq. (2.50), approximately when $N_j(U)^{\nu} > \nu^{-\nu}(1-\nu)^{-(1-\nu)}$. For $\nu = \frac{1}{2}$, for example, this would require $N_j(U) > 4$ while for $\nu = \frac{1}{3}$ we would need $N_j(U) > 6.75$, etc.

Consider, then, the "cumulative innovation" process introduced earlier. To approximate a neutral walk we repeatedly chose sequences uniformly at random from our network Γ_j and logged the discovery by mutation of neighbouring networks. Now suppose that instead of a neutral walk - which may be thought of as maximising drift - we instead start the process at a (uniformly selected) random sequence on our network but then at subsequent time steps we mutate *the same sequence* repeatedly. This "in-place" process may be though of as minimising drift under the scenario that the putative innovation which originally discovered Γ_j may be treated as a uniform random selection[7] from Γ_j. Let $m_i(x,U)$ be the probability that sequence $x \in \Gamma_j$ mutate under the mutation operator U to the neutral network Γ_i. Let us define the r.v.$I'_j(t)$ to be the (cumulative) number of innovations at time step t of our in-place process. We find, conditioning on the initial uniform selection:

$$\mathbf{P}\left(I'_j(t) = I'_j(t-1)+1\right) = \frac{1}{|\Gamma_j|}\sum_{x\in\Gamma_j}\sum_{i=1}^{N}\left[1-m_i(x,U)\right]^{t-1} m_i(x,U) \tag{2.55}$$

and, analogous to Eq. (2.47):

$$\mathbf{E}\left(I'_j(t)\right) = N - \frac{1}{|\Gamma_j|}\sum_{x\in\Gamma_j}\sum_{i=1}^{N}\left[1-m_i(x,U)\right]^{t} \tag{2.56}$$

As in the previous Subsection, this strongly suggests that an appropriate measure of accessibility for the in-place process might be the "mean entropy"

$$-\frac{1}{|\Gamma_j|}\sum_{x\in\Gamma_j}\sum_{i=1}^{N} m_i(x,U)\log_2\left(m_i(x,U)\right) \tag{2.57}$$

Thus we see that the "excess entropy" of the neutral work innovation process over the in-place process is precisely the mutual information measure $\mathcal{M}_j(U)$ of Def. 2.2.5 which we have termed the Markov Index of Γ_j. Note that we always have $0 \leq \mathcal{M}_j(U) \leq H_j(U)$, suggesting that, in some sense, *neighbouring networks are always "more accessible" via a neutral walk than via an in-place process* and that the less homogeneous a network - in terms of access to neighbouring networks by individual sequences - the more important drift is likely to be for discovery of innovations. Possible interpretations and implications of this statement will be discussed in Chapter 4. In the mean time, we take as a measure of network homogeneity or "utility of drift" (Chapter 4) the quantity:

$$\mathcal{D}_j^{perc}(U) = \frac{\mathcal{M}_j(U)}{H_j(U)} \tag{2.58}$$

which we shall call the the ***(percolation) drift factor*** of network Γ_j with respect to the mutation operator U. $\mathcal{D}_j^{perc}(U)$, which we may think of as ***"the fraction of network accessibility information conveyed by knowledge of the actual sequence"***, may vary between zero and one; if zero, which network a sequence in Γ_j mutates to is independent of the particular sequence, and drift will consequently be unimportant. The higher the drift factor, the more important drift is likely to be as regards accessibility of networks by mutation.

Some further remarks are in order regarding the usefulness of the accessibility measures introduced in this Section: firstly, they do not address actual fitnesses and may thus hold little relevance

[7] Whether this is likely to be a realistic assumption will be more carefully examined in Chapter 4.

for *optimisation*. For instance, a neutral network may percolate quite thoroughly but offer no access to *higher fitness* networks (*cf.* Chapter 6, Section 6.4.3). In the next Section we discuss fitness-dependent statistics. Secondly, the statistics are based on *uniform* sampling of networks. While (as noted) this may be appropriate for neutral walks (see also Chapter 4), for other population-based evolutionary processes (Chapter 3) the evolution of *mutational robustness* - or *mutational buffering* (Rendel, 1979; Huynen, Konings, & Hogeweg, 1993; Huynen & Hogeweg, 1994; A. Wagner & Stadler, 1999; Nimwegen, Crutchfield, & Huynen, 1999; Wilke, 2001) - implies that in practice neutral network sampling may be biased towards regions of the network where neutrality is higher. It is not clear to what extent this sampling bias is likely to affect our conclusions; more research would seem to be required.

2.3 Fitness-Dependent Structure

So far all the statistical properties we have looked at have depended only on some partitioning of the fitness landscape rather than on actual fitness. That is, given a landscape $\mathcal{L} = (\mathcal{A}, L, f)$ all statistics encountered up till now can be expressed solely in terms of a partitioning $\mathcal{A}^L = \bigcup_{i=1}^{N} \Gamma_i$ and a mutation operator $U(x)$. We now turn to those properties of a fitness landscape that depend in addition on actual fitness values $f(x) \in \mathbf{R}$.

2.3.1 The Mutant Fitness Distribution

In the next Chapter we shall see that by definition selection of sequences for an evolutionary search process is performed solely on the basis of fitness and that novel sequences are created by mutation. Hence, given a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ and a mutation operator U we shall be interested in the real-valued random variable $f(U(x))$ for $x \in \mathcal{A}^L$ (we recall that $U(x)$ is a random variable taking values in $\mathcal{A}^L$). This random variable represents the distribution of *fitness values*[8] of mutants of the sequence x under the mutation operator U. In particular we may take the expectation $\mathbf{E}(f(U(x)))$, variance $var(f(U(x)))$ and higher moments.

If X is a "random sequence" - i.e. a random variable taking values in $\mathcal{A}^L$ - the (real-valued) random variable $f(U(X))$ is also well-defined. Now many traditional statistical measures for fitness landscapes, such as *auto-correlation* (see below) are defined in terms of random variables of the form $f(X)$, $f(U(X))$, etc. where X is a *uniform* random variable on $\mathcal{A}^L$. As mentioned in the Introduction, however, such uniform random sampling of a fitness landscape may not be particularly useful, since the statistical properties of areas of the fitness landscape that are likely to interest us - in particular areas of high fitness - may be "swamped" by uninteresting, low fitness contributions. Traditional fitness landscape structural statistics may thus turn out to be less than useful in practice - see for instance (Smith, Husbands, & O'Shea, 2001). One way around this problem is to consider instead *fitness-conditional* sampling. Thus given a uniform random variable X on $\mathcal{A}^L$ we consider the distribution of $f(U(X))$ *conditional on* $f(X)$.

[8] In the literature, the distribution of actual offspring *sequences* (with respect to particular genetic operators) has been termed the *transmission function* (Altenberg, 1994; G. P. Wagner & Altenberg, 1996; Smith, Husbands, Layzell, & O'Shea, 2002), presumably because it mediates the transmission of genetic information from parent to offspring. In this thesis we prefer to work directly with the distribution of *fitness*, since this is all that our evolutionary processes (Chapter 3) actually "see".

To make the notation more compact, in the remainder of this Section let us define the jointly distributed r.v.'s:

$$W = f(X) \tag{2.59}$$
$$W' = f(U(X)) \tag{2.60}$$

where U is a mutation operator and X is uniform on $\mathcal{A}^L$. In words: *W is the fitness of a sequence selected uniformly at random from $\mathcal{A}^L$ and W' the fitness of a mutant (with respect to U) of the* same *sequence*. The object of study of this Section - the ***mutant fitness distribution*** - is thus the distribution of W' conditional on W.

We then define the ***mean mutant fitness of $\mathcal{L}$ with respect to*** U to be the real-valued function:

$$\mathcal{F}(U|w) = \mathbf{E}\left(W' \mid W = w\right) \tag{2.61}$$

i.e. $\mathcal{F}(U|w)$ is the expected fitness of a mutant of a uniformly sampled sequence, given that that sequence has fitness[9] w. The fitness-conditional distribution of mutant fitness is encapsulated by the mutation matrix for the *maximal* neutral partitioning $\mathcal{A}^L = \bigcup_{i=1}^{N} \Gamma_i$. Let $f(x) = w_i \in \mathbf{R}$ for $x \in \Gamma_i$ thus:

$$\mathbf{P}\left(W' = w_i \mid W = w_j\right) = m_{ij}(U) \tag{2.62}$$

which yields:

$$\mathcal{F}\left(U \mid w_j\right) = \sum_{i=1}^{N} w_i \, m_{ij}(U) \tag{2.63}$$

2.3.2 Parent-Mutant Fitness Correlation

A commonly encountered statistic in the literature is the ***auto-correlation function*** of a fitness landscape. It measures how correlated the fitness values are of sequences a given distance apart in sequence space (under the Hamming metric). In our treatment of fitness landscapes "nearness" of sequences is viewed in terms of *mutation*. We thus adopt the following definition of auto-correlation:

Definition 2.3.1. For a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ let U be a mutation operator and W, W' as defined above. The ***auto-correlation of $\mathcal{L}$ with respect to*** U is:

$$\rho(U) = corr\left(W, W'\right) \tag{2.64}$$

That is, $\rho(U)$ is the correlation between the fitnesses of a uniform randomly selected sequence and its mutant with respect to U.

The more conventional definition (Eigen, McCaskill, & Schuster, 1989; Weinberger, 1990; Stadler, 1996) of the auto-correlation function $\rho(d)$ at Hamming distance d (where $1 \leq d \leq L$) is then simply $\rho\left(U^{(d)}\right)$, where $U^{(d)}$ is uniform d-point (constant) mutation[10]. Auto-correlation is often described as a measure of *ruggedness* of a fitness landscape; if $\rho(U)$ is high we interpret this

[9] Strictly speaking $\mathcal{F}(U|w)$ is defined only if $w \in f\left(\mathcal{A}^L\right)$; i.e. if $w = f(x)$ for some $x \in \mathcal{A}^L$.

[10] In much of the literature auto-correlation is defined somewhat differently in terms of *random walks* on the sequence space - see for instance (Weinberger, 1990). We prefer the definition given here for its simplicity and also because, as remarked in (Stadler, 1996) "... it seems to be rather contrived to invoke a stochastic process in order to characterise a given function defined on a finite set.".

as saying that the fitnesses of "nearby" sequences (i.e. nearby in terms of mutation) are correlated and the landscape is therefore "smooth". If conversely auto-correlation is small then the fitnesses of nearby sequences are uncorrelated and might vary wildly - the landscape is "rugged".

We now show that auto-correlation gives us limited information about the distribution of mutant fitness. Firstly, note that the random variable (conditional expectation) $\mathcal{F}(U|W) = \mathbf{E}(W' \mid W)$ is well defined. It is straightforward to show that:

$$\mathbf{E}(W') = \mathbf{E}(\mathcal{F}(U|W)) \tag{2.65}$$

and:

$$cov(W, W') = cov(W, \mathcal{F}(U|W)) \tag{2.66}$$

If in particular our mutation operator U is *uniform*, then, as noted previously, $U(X)$ is identically (but not, of course, independently) distributed as X - both are uniform on $\mathcal{A}^L$. The marginal distributions of W and W' are thus identical, so that with Eq. (2.66) we may state:

> For a uniform mutation operator U the auto-correlation $\rho(U)$ depends on the mutant fitness distribution only via the *mean* mutant fitness

That is, if we know the fitness distribution W and the mean mutant fitness function $\mathcal{F}(U|w)$ then we may calculate $\rho(U)$. Equivalently, to calculate $\rho(U)$, in addition to the $\upsilon_i = \mathbf{P}(W = w_i)$ - i.e. the distribution of W - we need just $\mathcal{F}(U|w_j)$ as given by Eq. (2.63) rather than the full mutation matrix $m_{ij}(U)$. This suggests that auto-correlation is likely to be of limited usefulness, since it does not depend on higher moments of the mutant fitness distribution. It cannot, for instance, tell us much about the probability of *fitness-increasing* mutation (but see Section 2.3.4 below).

In general, we might expect that auto-correlation decreases with Hamming distance between parent and mutant (*cf.* Chapter 5, Chapter 6). Landscapes for which the decay of auto-correlation with Hamming distance is *exponential* have been termed ***elementary***[11]. That is, setting $\rho(d) = \rho(U^{(d)})$ where $U^{(d)}$ is the d-point constant mutation operator, we have:

$$\rho(d) = e^{d/\ell} \tag{2.67}$$

where $\ell > 0$ is the ***correlation length*** (Weinberger, 1990; Stadler, 1996; Kauffman, 1993).

2.3.3 Linear Correlation

We shall call a fitness landscape ***linearly correlated with respect to*** U iff the mean mutant fitness depends linearly on fitness; i.e. if:

$$\mathcal{F}(U|w) = aw + b \tag{2.68}$$

for some constants a, b. If U is in addition *uniform* then it follows from Eq. (2.66) that the proportionality constant a must be precisely the auto-correlation $\rho(U)$ and we may re-write the linear correlation condition Eq. (2.68) as:

$$\mathcal{F}(U|w) - \bar{w} = \rho(U)(w - \bar{w}) \tag{2.69}$$

[11] This is not quite the technical definition: see for (Stadler, 1996) for details.

where:

$$\bar{w} = \mathbf{E}(W) \tag{2.70}$$

is the ***mean fitness*** of the landscape. Intuitively, on a linearly correlated landscape mutation reduces fitness [12] (on average) by a factor equal to the auto-correlation: the higher up the landscape we go, the more severely mutation might be expected to impact fitness. Research by the author suggests that linear correlation (or its analogue for *random* fitness landscapes - see Section 2.4 below) may be, at least to some approximation, a ubiquitous and important property of many real world as well as model fitness landscapes. See also Chapter 5 and Chapter 6.

2.3.4 Evolvability

Perhaps the most relevant statistic as regards the performance of a (mutation-based) optimisation process is the probability that mutation be "beneficial" - that it produce an *improvement* in fitness. By definition, if a (mutation-based) evolutionary search process is to evolve high-fitness sequences it must do so via such fitness-increasing mutations. We should thus expect *availability* of fitness-increasing mutations to be a basic metric of how well an evolutionary search process can fare on a given landscape; in other words, on the capacity for fit sequences to evolve. We thus define the ***evolvability*** statistic (Altenberg, 1994; Altenberg, 1995; Smith et al., 2001; Smith, Husbands, et al., 2002):

$$\mathcal{E}(U|w) = \mathbf{P}\left(W' > w \;\middle|\; W = w\right) \tag{2.71}$$

(note that evolvability thus defined is a *fitness rank*-dependent statistic, in the sense that it is invariant under a rank-preserving transformation of fitness). We might expect the *decay* of evolvability with increasing fitness to be particularly significant; we shall return to this point for the fitness landscapes of Chapters 5 and 6.

It was noted above that auto-correlation depends only on the *mean* fitness of mutants (of a given fitness) but tells us nothing - in lieu of more detailed knowledge of the full distribution of mutant fitness - about the probability that mutation actually increase fitness. For instance, a correlated landscape may feature *locally sub-optimal* neutral networks (*cf.* Chapter 6) with respect to, say, one-point mutation. Then *no* mutant of a sequence from that network can, by definition, be fitness-increasing. Nevertheless we still might suppose that, "in general", with sufficient correlation, the "tail" of the mutant fitness distribution is likely to "overlap" the parent sequence fitness (Fig. 2.2). This thesis is concerned explicitly with fitness landscapes featuring high neutrality and some degree of auto-correlation. We will in particular tend to assume that *higher fitness sequences are more likely to produce higher fitness mutants than sequences of lower fitness* (*cf.* Chapter 3, Section 3.4.1). While as we have pointed out this doesn't follow of necessity from auto-correlation alone - see also (Altenberg, 1995) - to paraphrase (Nimwegen & Crutchfield, 1998):

> We believe that this assumption is consonant, by definition, with the very idea of evolutionary search for optimisation. Imagine, on the contrary, that higher fitness sequences are more likely to be close to *lower* fitness sequences. It then seems strange to have selection preferably replicate sequences of higher fitness over sequences of lower fitness. Therefore designing a search algorithm to select higher fitness sequences preferentially over lower fitness sequences implicitly assumes that [fitness-increasing] sequences tend to be found close to sequences of current best fitness.

[12] Or, rather, the difference between fitness and mean fitness.

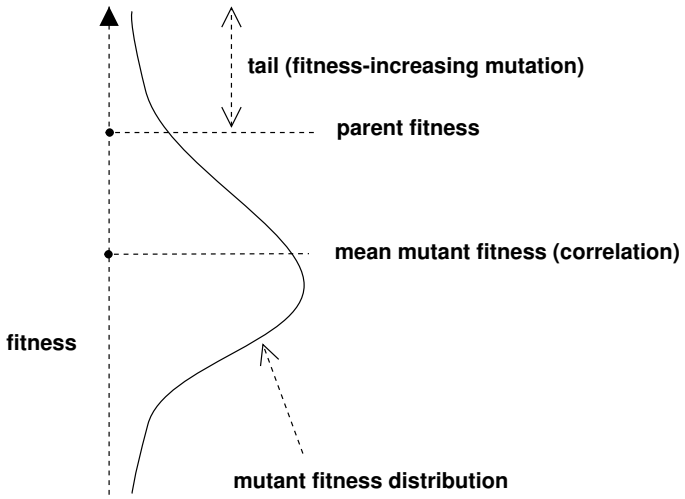

Figure 2.2: Mutant fitness distribution with "tail" of fitness-increasing mutants.

This echoes our remarks in the Introduction, that it is a matter of faith for the GA practitioner that his fitness landscape *is* in fact amenable to evolutionary optimisation! It might be said (Altenberg, 1994; Altenberg, 1995) that *evolvability* rather than correlation is precisely the quantity we would like to be present in our landscapes; we might argue, however, that correlation is: (a) likely to be present to some degree in a "well-designed" artificial evolution problem (*cf.* Section 1.1.1 in the Introduction): (b) comparatively well-understood, amenable to analysis and a *de facto* metric of fitness landscapes in the literature and: (c) will hopefully, as argued above, imply some degree of evolvability.

Evolvability and Optimal Mutation

In Section 2.2.5 we demonstrated (Prop. 2.2.2) that to maximise the probability of (a uniform randomly selected sequence) mutating from one given neutral network to another, we should use constant mutation at a rate that may be determined from the mutation matrix. It is clear that an analogous argument works equally for *fitness-increasing* mutation from a given network:

Propostion 2.3.1. *Given a neutral network* Γ *with fitness* w*, the mutation mode which maximises the probability that a mutant of a sequence selected uniformly at random from* Γ *has fitness* $> w$ *is* n*-point (constant) mutation.*

In particular, if Γ is maximal then the mutation mode of the above Proposition is given by $\mathcal{U}^{(n)}$ where n is such as to maximise the evolvability $\mathcal{E}(U|w)$ with mutation operator $U = \mathcal{U}^{(n)}$ on Γ. Again, if there is no *unique* such n - if, say, $n_1, n_2, \ldots$ all maximal the probability of fitness-increasing mutation - then any mutation mode with $\mathbf{P}(\mathcal{U} = n) = 0$ unless $n = n_k$ for some k suffices.

Prop. 2.3.1 has significant implications for the design of efficient evolutionary search processes. In particular it suggests that the common practice among GA practitioners of using what we have termed Poisson mutation - i.e. of mutating alleles per-locus with a fixed probability - may

not be the best choice of mutation operator[13,14]. For a mutation-based search process we might indeed attempt to deploy an optimal mutation *operator* that always uses the mode/rate which maximises the probability of fitness-increasing mutation in the sense of Prop. 2.3.1. Now without prior knowledge of evolvability this may appear unfeasible. As we shall see however (Chapter 5 and Chapter 6), some knowledge (or presumption) of landscape structure along with "on-the-fly" information-gathering during evolutionary search may assist us towards this end. In general auto-correlation (and hence, as we have argued, evolvability) tends to drop off with increasing Hamming distance (*cf.* Chapter 6). The optimal mutation rate for a neutral network thus involves a "balancing act" between correlation and neutrality:

- If the mutation rate is too low, mutants will tend to be neutral and thus have no chance of locating a higher-fitness network.
- If the mutation rate is too high, the mutant's fitness will tend to be uncorrelated with the fitness of its parent sequence.

The optimal mutation rate of Prop. 2.3.1, then, involves mutating "just enough" to get off the network but not stray too far from it... this will lead us (Chapters 5 and 6) to our *1/e Neutral Mutation Rule*.

Evolvability and Neutral Drift

In Chapter 4 we discuss the utility of drift for an evolutionary optimisation process. Here we note that, as mentioned in Section 2.2.6, the (percolation) drift factor $\mathcal{D}^{perc}$ for a neutral network does not take fitness into account (see also Chapter 5, Section 5.4.1). Here we remedy that situation somewhat by introducing the ***evolvability drift factor*** $\mathcal{D}^{evol}$ for a neutral network Γ. It measures, essentially, the degree to which the probability of finding fitness-increasing mutations from Γ depends on the actual sequence sampled. Thus let X be uniform random on Γ and let Z be an indicator r.v. for discovery of a higher fitness sequence by a mutant of X; i.e. $Z = 1$ if $f(U(X)) > f(X)$ and $Z = 0$ otherwise. We then define the ***(evolvability) drift factor*** for the neutral network Γ to be :

$$\mathcal{D}^{evol} = \frac{I(Z,X)}{H(Z)} \tag{2.72}$$

where $I(Z,X) = H(Z) - H(Z|X)$ is the *mutual information* between Z and X. Intuitively, $\mathcal{D}^{evol}$ is ***the fraction of fitness-increasing mutation information conveyed by knowledge of the actual sequence for network*** Γ. Thus if $\mathcal{D}^{evol} = 0$ then Z is independent of X - it makes no difference where we mutate from on Γ - and drift will be irrelevant to discovery of higher fitness networks. In general the larger the evolvability drift factor, the more important we might expect neutral drift to be as regards discovery of higher fitness networks (see also Chapter 4).

[13] It should be remarked, however, that many GA practitioners do not see the rôle of mutation as being primarily as a *search* operator; see Chapter 3, Section 3.4 for more on this point.

[14] There may occasionally be sound arguments for *not* deploying constant mutation, on the grounds that (i) there may sometimes be *no* fitness-increasing mutations for a given constant mutation rate from certain sequences and (ii) if the fitness landscape has many "locally sub-optimal" neutral networks (*cf.* Chapter 6) then optimisation may benefit from the occasional "long jump" mutation generated by eg. Poisson mutation.

2.4 Random Fitness Landscapes

As alluded to in the Introduction to this thesis, we are as likely to be presented with a *class* of optimisation problems - that is, a class or *family* of fitness landscapes - than a single landscape. Thus the GA practitioner may consider the landscape presented to him for optimisation as drawn from some "random distribution" of fitness landscapes. Indeed, many of the landscape *models* in the literature are defined in terms of random parameters; in Chapter 6 we shall encounter just such a model. We thus need some notion of a ***random fitness landscape***.

Our definition(s) of a random fitness landscape are not rigorous. For a more mathematically precise treatment we refer the interested reader to eg. (Reidys & Stadler, 2001) [Reidys & Stadler - Neutrality in Fitness Landscapes]:

Definition 2.4.1. A ***random fitness function*** (over the sequence space $\mathcal{A}^L$) is a random variable F with values in a (measurable) subset $\Omega \subseteq \left\{ f \;\middle|\; f : \mathcal{A}^L \longrightarrow \mathbf{R} \right\}$ of the set of all fitness functions on $\mathcal{A}^L$. A ***random fitness landscape*** is a triple $\mathcal{L} = (\mathcal{A}, L, F)$ where F is a random fitness function over $\mathcal{A}^L$.

We shall sometimes refer to a random fitness landscape as defined above as a *family* or *ensemble* of (random) landscapes.

Given a random fitness landscape $(\mathcal{A}, L, F)$ and a sequence $x \in \mathcal{A}^L$ we may consider $F(x)$ as a real-valued random variable, formed by *evaluating* the random fitness function F at x; we write the distribution of $F(x)$ symbolically as:

$$\mathbf{P}\left(F(x) \leq w\right) = \sum_{f \in \Omega} \mathbf{P}\left(f(x) \leq w\right) \mathbf{P}\left(F = f\right) \tag{2.73}$$

Similarly if $x_1, x_2, \ldots \in \mathcal{A}^L$ we consider the evaluations $F(x_1), F(x_2), \ldots$ to be *jointly-distributed* random variables, formed by evaluating the *same* sampled value of F at $x_1, x_2, \ldots$. Symbolically:

$$\mathbf{P}\left(F(x_1) \leq w_1, F(x_2) \leq w_2, \ldots\right) = \sum_{f \in \Omega} \mathbf{P}\left(f(x_1) \leq w_1, f(x_2) \leq w_2, \ldots\right) \mathbf{P}\left(F = f\right) \tag{2.74}$$

In this sense, we may specify a random fitness landscape by the joint distribution of the random variables $(F(x) \mid x \in \mathcal{A}^L)$. We might have defined , alternatively:

Definition 2.4.2 (alt). A ***random fitness function*** (over the sequence space $\mathcal{A}^L$) is a family $(F(x) \mid x \in \mathcal{A}^L)$ of jointly-distributed real-valued random variables indexed by $\mathcal{A}^L$. A ***random fitness landscape*** is a triple $\mathcal{L} = (\mathcal{A}, L, F)$ where F is a random fitness function over $\mathcal{A}^L$.

This "constructive" specification is perhaps more intuitive, particularly when it comes to sampling random landscapes in simulation (*cf.* Chapter 6): to sample a random landscape we need to sample the $F(x)$ for all x in the sequence space. This is perhaps best illustrated by example:

Example 2.4.1. For a real-valued random variable Z the *fully uncorrelated random fitness landscape with underlying distribution Z* is that for which the $F(x)$ are iidas Z. We may sample this landscape by assigning fitnesses to sequences independently from the distribution Z.

Many of the statistics already encountered have "ensemble" analogues. These must all now, however, be expressed strictly in terms of fitness-dependence rather than relative to a particular neutral

partitioning, since there may not be any natural neutral partitioning which will be valid across the random family. A neutral network on one "sample landscape" of a family of random landscapes may not be a neutral network on another! We may still, however, make statements about the statistical properties of a neutral network *conditional on that network having some particular fitness*. We shall calculate many such statistics in Chapter 6.

As regards mutation, we would like to allow the situation where the mutation operator depends on the "sample" of the random landscape. For instance, there may be some "optimal" mutation operator for landscapes drawn from some class, where the optimal operator depends on the particular landscape. Thus when we talk of a mutation operator for a *family* Ω of fitness landscapes we shall mean a mapping $U: \ f \mapsto U_f$ which assigns a mutation operator U_f to a fitness function $f \in \Omega$. We must, however, be careful what we mean by a "compatible" mutation operator (Def. 2.2.2). If there is some natural neutral partitioning - eg. the *maximal* partitioning - for every member $f \in \Omega$ of a family of landscapes, then by a "compatible" mutation operator we shall mean a mutation operator $f \mapsto U_f$ on the family where each U_f is compatible with respect to the partitioning on its own landscape f. A special case of such a compatible mutation operator is a *fitness-dependent mutation operator*: to each fitness value $w \in \mathbf{R}$ there corresponds a mutation mode $\mathcal{U}_w$. This defines a mutation operator for any fitness function $f \in \Omega$ whereby a sequence $x \in \mathcal{A}^L$ is mutated according to $\mathcal{U}_{f(x)}$. We shall encounter such a mutation operator in Chapter 6. Frequently we shall deal simply with mutation operators such as the $U^{(d)}$ which are uniform and identical for every member of a family; the above considerations then do not arise.

One way of forming statistics for a random family of landscapes is simply by *averaging* a (single landscape) statistic over the family; we shall use angle brackets to denote such averages. For example, given a family Ω of landscapes and a mutation operator $f \mapsto U_f$ as above, we may consider auto-correlation as mapping a fitness function $f \in \Omega$ to the real number $\rho_f(U_f) =$ auto-correlation of fitness landscape f with respect to mutation operator U_f. Given a *random* fitness landscape $\mathcal{L} = (\mathcal{A}, L, F)$ we may then view $\rho_F(U_F)$ as a (real-valued) *random variable*. We write the mean (if it exists) of this r.v. as $\langle \rho_F(U_F) \rangle_F$ or just $\langle \rho(U) \rangle$ if the random fitness function and mutation operator are clear from the context. Fitness-dependent statistics may be similarly averaged (for fixed fitness w) over an ensemble of fitness landscapes; eg. the probability $\nu(U; w)$ that a mutation of a (uniform randomly selected) sequence by U is neutral given that the sequence has fitness w, where U is a fitness-dependent mutation operator as described in the previous paragraph. A caveat: there may be alternative "ensemble" versions of some statistics - auto-correlation is one example we shall encounter below - which will *not* generally be the same as the corresponding averaged statistic[15].

Given a random fitness landscape $\mathcal{L} = (\mathcal{A}, L, F)$ and a mutation operator $f \mapsto U_f$, for $x \in \mathcal{A}^L$ the jointly-distributed random variables $F(x), F(U_F(x))$ are well-defined; they should be thought of as representing the fitnesses of sequence x and a mutant of x respectively, where *both fitnesses are evaluated with the same fitness function and mutation uses the operator corresponding to that fitness function* - i.e. mutation and fitness evaluation are on the *same landscape drawn at random from the family*. Similarly, if X is a r.v. on $\mathcal{A}^L$ then $F(X), F(U_F(X))$ are well-defined and, as for

[15] The extent to which an averaged statistic approximates its ensemble analogue might be thought of as a kind of "self-averaging".

non-random landscapes, we define the jointly-distributed r.v.'s:

$$W = F(X) \tag{2.75}$$
$$W' = F(U(X)) \tag{2.76}$$

where X is *uniform* on $\mathcal{A}^L$. Again, it is important bear in mind that W, W' represent the fitnesses of a (uniform random) sequence and a mutant of that sequence evaluated on the *same* landscape. For a fitness value $w \in \mathbf{R}$ we may consider as before the fitness of a mutant conditional on the fitness of the un-mutated sequence being equal to w; i.e. W' conditional on $W = w$. The distribution of W' conditional on W might be considered an ensemble analogue of our mutation matrix $\boldsymbol{m}(U)$.

We now define the ***ensemble mean mutant fitness of*** $\mathcal{L}$ ***with respect to*** U - *cf.* Eq. (2.61) - to be the function[16]:

$$\mathcal{F}(U|w) = \mathbf{E}\left(W' \mid W = w\right) \tag{2.77}$$

Similarly we may define the ***ensemble auto-correlation of*** $\mathcal{L}$ ***with respect to*** U - *cf.* Eq. (2.64) - to be:

$$\rho(U) = corr\left(W, W'\right) \tag{2.78}$$

Note that, as alluded to above, the ensemble auto-correlation $\rho(U)$ will *not* in general be equal to the auto-correlation $\langle \rho_F(U_F) \rangle_F$ of individual landscapes averaged of the family. As for the single landscape case we have:

$$\mathbf{E}\left(W'\right) = \mathbf{E}\left(\mathcal{F}(U|W)\right) \tag{2.79}$$

and:

$$cov\left(W, W'\right) = cov\left(W, \mathcal{F}(U|W)\right) \tag{2.80}$$

so that for a *uniform* mutation operator U (i.e. U_f is uniform, although not necessarily the same, for all fitness functions f in the family) ensemble auto-correlation $\rho(U)$ depends only on the fitness distribution W and the ensemble mean mutant fitness function $\mathcal{F}(U|w)$ rather than the full (joint) distribution of W, W'. Again we call a random fitness landscape ***linearly correlated withe respect to*** U iff:

$$\mathcal{F}(U|w) = aw + b \tag{2.81}$$

for constants a, b. Again, if U is uniform then a must be the (ensemble) auto-correlation $\rho(U)$ and we may re-write the linear correlation condition as:

$$\mathcal{F}(U|w) - \bar{w} = \rho(U)(w - \bar{w}) \tag{2.82}$$

where :

$$\bar{w} = \mathbf{E}(W) \tag{2.83}$$

is the (ensemble) mean fitness of a uniformly sampled sequence of the family.

There are also natural ensemble versions of neutrality statistics. We may define:

$$\nu(U|w) = \mathbf{P}\left(W' = w \mid W = w\right) \tag{2.84}$$

[16] Strictly speaking, this function is defined on $\{w \in \mathbf{R} \mid \phi(w) \neq 0\}$ where $\phi(w)$ is the probability density function (pdf) of W.

Note that we would not expect neutrality to "self-average"; i.e. we would not expect the average over the ensemble of the neutralities of networks of fitness w to be the same as $\nu(U|w)$. Finally, the (ensemble) definition of evolvability (Section 2.3.4) goes through formally unchanged for random fitness landscapes:

$$\mathcal{E}(U|w) = \mathbf{P}\left(W' > w \;\middle|\; W = w\right) \tag{2.85}$$

Again, we would not expect this quantity to "self-average".

Chapter 3

Evolutionary Dynamics

3.1 Populations

Firstly we introduce some mathematical preliminaries. If Z is any set and $M > 0$ an integer, then the group of permutations of $1,2,\ldots,M$ acts (on the left) as a transformation group on the set Z^M of sequences $\boldsymbol{z} = (z_1, z_2, \ldots, z_M)$ by:

$$\sigma \cdot (z_1, z_2, \ldots, z_M) = \left(z_{\sigma^{-1}(1)}, z_{\sigma^{-1}(2)}, \ldots, z_{\sigma^{-1}(M)}\right) \tag{3.1}$$

or:

$$(\sigma \cdot \boldsymbol{z})_\alpha = z_{\sigma^{-1}(\alpha)} \qquad \text{for } \alpha = 1,2,\ldots,M \tag{3.2}$$

for a permutation σ. We may check that if σ, σ' are two permutations of $1,2,\ldots,M$ then for any $\boldsymbol{z} \in Z^M$ we have:

$$\sigma \cdot (\sigma' \cdot \boldsymbol{z}) = (\sigma\sigma') \cdot \boldsymbol{z} \tag{3.3}$$

as required. The group of permutations of $1,2,\ldots,M$ thus induces an equivalence relation on Z^M, the equivalence classes being the orbits of the group action; i.e. $\boldsymbol{z} \equiv \boldsymbol{z}'$ iff $\exists$ a permutation σ such that $\boldsymbol{z}' = \sigma \cdot \boldsymbol{z}$. We shall refer to this as the ***re-ordering*** equivalence relation on Z^M. We shall use angle brackets $< \ldots >$ to denote "equivalence class of $\ldots$ under re-ordering"; i.e. if $(z_1, z_2, \ldots, z_M) \in Z^M$ we write $< z_1, z_2, \ldots, z_M >$ for the equivalence class to which $(z_1, z_2, \ldots, z_M)$ belongs.

If $\phi : \; Z^M \longrightarrow W$ is a mapping from Z^M to some set W such that for any $\boldsymbol{z} \in Z^M$ and any permutation σ of $1,2,\ldots,M$ we have: $\phi(\sigma \cdot \boldsymbol{z}) = \phi(\boldsymbol{z})$ we say that ϕ is a *symmetric* function on Z^M, or is *invariant under re-ordering*. Similarly for a mapping $\psi : \; Z^M \longrightarrow W^M$, we call ψ invariant under re-ordering iff $\psi(\sigma \cdot \boldsymbol{z}) = \sigma \cdot \psi(\boldsymbol{z}) \; \forall \boldsymbol{z}, \sigma$.

By a "population" of objects we would like to mean simply a collection of objects, some of which may be identical. For instance for a population of genotypes we would like to think of identical genotypes - *clones* - as being *indistinguishable*[1] from the evolutionary perspective. This

[1] Of course in natural evolution clones - identical twins! - are not identical on a *phenotypic* (and therefore fitness) level. This may indeed be true of artificial evolution eg. if phenotype is developed from genotype through some kind of "noisy" procedure. An example of this is where a neural network phenotype is "grown" stochastically from its genotypic specification. We specifically exclude such systems in this work. Note that this is not the same as noise on *fitness evaluation* (Section 8.2).

motivates the following:

Definition 3.1.1. A ***population of size*** M on a set Z is the set $\mathcal{P}^M(Z)$ of equivalence classes of Z^M under re-ordering. By convention we define $\mathcal{P}^0(Z)$ to be the set $\{\emptyset\}$ and we define $\mathcal{P}(Z) = \bigcup_{M=0}^{\infty} \mathcal{P}^M(Z)$ (disjoint union) to be the set of populations on Z of any size. For $\boldsymbol{z} \in \mathcal{P}(Z)$ we write $|\boldsymbol{z}|$ for the population size of $\boldsymbol{z}$; i.e. $|\boldsymbol{z}| = M \Leftrightarrow \boldsymbol{z} \in \mathcal{P}^M(Z)$.

The populations we are interested in are, of course, populations of genotypes on a sequence space $\mathcal{A}^L$. We shall use bold symbols $\boldsymbol{x},\boldsymbol{y},\ldots$ for populations in $\mathcal{P}(\mathcal{A}^L)$ and by abuse of notation, for $\boldsymbol{x} =< x_1, x_2, \ldots, x_M > \in \mathcal{P}^M(\mathcal{A}^L)$ and a sequence $y \in \mathcal{A}^L$ we write $y \in \boldsymbol{x}$ to mean that $y = x_\alpha$ for some $\alpha \in \{1, 2, \ldots, M\}$, which we interpret as saying that "*y is represented in the population* $\boldsymbol{x}$".

If $\mathcal{L} = (\mathcal{A}, L, f)$ is a fitness landscape and $\boldsymbol{x} =< x_1, x_2, \ldots, x_M > \in \mathcal{P}^M(\mathcal{A}^L)$ a population of size M on $\mathcal{A}^L$ we write $f(\boldsymbol{x}) =< f(x_1), f(x_2), \ldots, f(x_m) > \in \mathcal{P}^M(\mathbf{R})$. The ***mean fitness*** of $\boldsymbol{x}$ is defined to be:

$$\bar{f}(\boldsymbol{x}) = \frac{1}{M} \sum_{\alpha=1}^{M} f(x_\alpha) \tag{3.4}$$

Note that the RHS specifies a symmetric function so that $\bar{f}(\boldsymbol{x})$ is well-defined. We also define the ***best fitness*** in the population to be:

$$f^*(\boldsymbol{x}) = \max\{ f(x_\alpha) \mid \alpha = 1, 2, \ldots, M \} \tag{3.5}$$

3.2 Selection Operators and Evolutionary Operators

In the broadest sense an evolutionary process on a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ is a stochastic process on the set of populations $\mathcal{P}(\mathcal{A}^L)$ on the sequence space $\mathcal{A}^L$. This is, however, obviously too general a definition to describe what we would like to think of as *evolution*. In particular, we wish to pinpoint more clearly the notion of evolution as constituting *fitness-based selection* and *replication with variation*.

Our definition of an evolutionary process on a fitness landscape will inevitably be restrictive. To the GA researcher we therefore issue a warning against dissapointment if their favourite genetic algorithm does not appear to fall within our ambit; we shall at least attempt to make explicit the restrictions inherent in our approach.

A brief description and motivation for our definition is as follows: evolution is driven by *selection* and *variation* mechanisms. The first major restriction is that as regards variation, we deal solely with *mutation* as described in the previous Chapter. Our motivations for excluding *recombination* will be discussed more fully in Chapter 7 - suffice to say that recombination could be brought into our framework without too much difficulty. Our definition of mutation (Chapter 2) is also, of course, restrictive; in principle more general forms of mutation might be allowed. Again, if we were discussing natural evolution this might be necessary.

Another major restriction is that (in a sense to be made precise below) selection will depend only on the *fitness* of sequences. This will potentially exclude a fairly extensive class of genetic algorithms, such as *spatially distributed* GA's (McIlhagga et al., 1996; McIlhagga, Husbands, & Ives, 1996; Sarma & Jong, 1999) which make use of structural mechanisms additional to the sequence $\rightarrow$ fitness mapping.

A further crucial restrictive assumption we shall make is that an evolutionary process be *Markovian*. This is for two reasons: the first is purely pragmatic: Markov processes are far more amenable to analysis than stochastic processes in general. The second is empirical: many (but by no means all[2]) stochastic search processes in common use - and arguably those in particular that might be recognised as *evolutionary* search processes - are Markovian. We don't, however, insist in general that an evolutionary process be *time homogeneous*. This allows search processes such as *simulated annealing* (Kirkpatrick, Gelatt, & Vecchi, 1983; Catoni, 1996) to fall under our definition of an evolutionary process.

As regards the *time* aspect of our evolutionary processes, in this thesis we restrict ourselves to *discrete time* (Markov) processes. While it is perfectly feasible (and indeed in the biological case probably preferable) to discuss evolutionary processes in continuous time, we are principally concerned with artificial (probably computer-based) processes which occur naturally in discrete time. Suffice to say that most results presented have continuous time counterparts. As has already been remarked, throughout this thesis we take the view that as regards search/optimisation, the most time-intensive computational aspect is considered to be fitness evaluation. Therefore, in analysing and evaluating search processes, we should always use the number of fitness evaluations as a measure of time. However: *it should not be assumed that the time step of our evolutionary process - as a Markov process - is necessarily a single, or even a fixed number, of fitness evaluations*. Instead, we measure "Markovian" time in *generations*. The number of fitness evaluations per generation may vary and our generations may well *overlap* in the biological sense (Maynard Smith, 1998); that is, population members may "survive" into subsequent generations. Care must be taken to avoid confusion on this issue; in general we shall try to use the Roman "t" for time measured in fitness evaluations and the Greek "τ" for generational (Markovian) time.

3.2.1 Evolutionary Operators

Broadly, then, we need a mechanism - an ***evolutionary operator*** - to form a new population from a current population such that sequences in the new population are either copies (clones) or mutant offspring of sequences from the current population. Mutation has been dealt with in the previous Chapter. A rigourous definition of selection in the sense that we require turns out to be somewhat technical and is thus relegated to Appendix A; here we supply a non-rigourous, intuitive definition. Suppose given a fitness landscape and a mutation operator on the associated sequence space.

> To form a new population from a current population (of size M) we generate a (finite) number of mutants of sequences from the current population using the mutation operator. The fitness of each new mutant is evaluated[3]. We then select (copies of) sequences from both the original population and the new mutants to comprise a new population; this selection procedure may be *stochastic* and must depend *only on fitness* (of original and/or mutant sequences).

A prescription for fitness-based selection of the above nature (independent of the actual mutation operator deployed) defines what we shall call a ***selection operator for population size M*** (*cf.*

[2] *Tabu search* (Glover & Laguna, 1993) for example, is not a Markov process.

[3] Since mutation is the only mechanism for producing *new* sequences, we assume that fitness is evaluated when, and only when, a mutation occurs; this notwithstanding that mutation might conceivably produce an offspring sequence identical to its (already evaluated) parent.

Def. A.1.1). We then define (*cf.* Def. A.1.2):

> An ***evolutionary*** or ***generational operator for population size*** M on a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ is a pair $\mathcal{G} = \mathcal{G}(\mathcal{S}, U)$ where $\mathcal{S}$ is a selection operator for population size M (Def. A.1.1) and U a mutation operator. It defines a map from the set of populations $\mathcal{P}^M(\mathcal{A}^L)$ on the sequence space $\mathcal{A}^L$ to the set of random variables on the set of populations $\mathcal{P}(\mathcal{A}^L)$ on $\mathcal{A}^L$ (*cf.* Eq. A.6).

Thus an evolutionary operator $\mathcal{G}$ for population size M takes as its argument a population $\boldsymbol{x} \in \mathcal{P}^M(\mathcal{A}^L)$ and generates stochastically a new population $\boldsymbol{y} \in \mathcal{P}(\mathcal{A}^L)$ - a "next generation" - with probability $\mathbf{P}(\mathcal{G}(\boldsymbol{x}) = \boldsymbol{y})$, by sequentially generating mutants of current sequences and selecting the new population from the mutants and original sequences on the basis of fitness alone. A key point in our definition is that, in creating the next generation, we may only mutate sequences from the current generation; we may not create "mutants of mutants". This ties in with the usual conception of "generation" in natural evolution (Maynard Smith, 1998; Crow & Kimura, 1970). As previously stated, the chief restrictions inherent in our definition are (besides the absence of recombination) the dependence on fitness *only* for selection and our somewhat restrictive definition of mutation.

Frequently a selection operator - or, rather, a *series* of selection operators - will be defined for a *range* of population sizes (see below for some examples). In this case it is easier to think of the corresponding evolutionary operator as mapping from $\mathcal{P}(\mathcal{A}^L)$ - the set of populations of *any* size - to the set of r.v.'s on $\mathcal{P}(\mathcal{A}^L)$. We shall thus fudge the issue and generally drop the "for population size M" rider; it will be clear from context to which population size(s) our operators apply.

We have, given an evolutionary operator $\mathcal{G}$ and a population $\boldsymbol{x}$, the random variables $|\mathcal{G}(\boldsymbol{x})| =$ the *size* (Eq. A.9) of a population created from $\boldsymbol{x}$ and $\|\mathcal{G}(\boldsymbol{x})\| =$ the *number of fitness evaluations* (Eq. A.10) required to create a new population from $\boldsymbol{x}$. If $|\mathcal{G}(\boldsymbol{x})| = |\boldsymbol{x}|$ (a.s.) for any population $\boldsymbol{x}$ then the selection operator is *fixed population size*. If $\|\mathcal{G}(\boldsymbol{x})\|$ is constant (a.s.) for any population $\boldsymbol{x}$ then the selection operator is *fixed number of fitness evaluations*. Most selection operators we shall encounter are both fixed population size and fixed number of fitness evaluations (an exception is Example 3.2.3 below).

A selection operator is *generational* (or has *non-overlapping generations*) if a new generation consists entirely of mutants - i.e. none of the original population sequences are selected (un-mutated) - otherwise selection is *steady-state* (or has *overlapping generations*). A selection operator is *fitness-proportional* if it is invariant under re-scaling of fitness by a scalar factor; it is *rank-invariant* (or just *ranked*) if it is invariant under transformations leaving rank-order of fitness invariant; it is *elitist* if the population best fitness is non-decreasing from generation to generation[4]. See Appendix A.1 for more precise definitions.

We now introduce a few selection schemes which should be familiar from either population biology or the GA field. See Appendix A.2 for precise definitions in terms of our mathematical formalism of selection operators. We note that each of these selection operators defines, along with a mutation operator a (homogeneous) *evolutionary process* (see next Section 3.2.2) which may be familiar as a model in population genetics, a GA or other evolutionary search procedure.

[4] Elitism is sometimes taken to mean the survival of (at leat one copy of) a current best fitness sequence into the next generation.

Example 3.2.1. ***Birth-and-death selection:*** to form the new population some sequence "dies" - is eliminated from the population - while another sequence is "born" - i.e. arises as a mutant replica of an existing sequence. Birth and death selections are on the basis of fitness. For each $\boldsymbol{w} =< w_1,\ldots,w_M >\in \mathcal{P}^M(\mathbf{R})$ we thus define random variables $B(\boldsymbol{w})$ and $D(\boldsymbol{w})$ taking values in $1,2,\ldots,M$. Intuitively, for a population $\boldsymbol{x} =< x_1,\ldots,x_M >$ with $f(\boldsymbol{x}) = \boldsymbol{w}$ the sequence $x_{B(\boldsymbol{w})}$ replicates, while the sequence $x_{D(\boldsymbol{w})}$ dies and is replaced by the new mutant $U\left(x_{B(\boldsymbol{w})}\right)$. Note that we might have $D(\boldsymbol{w}) = B(\boldsymbol{w})$; i.e. the same sequence is chosen to replicate and die. Birth-and-death selection is steady-state, of fixed population size and fixed number of fitness evaluations ($= 1$; only the new mutant needs to be evaluated for fitness). It is fitness-proportional (resp. ranked) iff the birth and death r.v.'s $B(\boldsymbol{w})$ and $D(\boldsymbol{w})$ are invariant by scalar multiplication of $\boldsymbol{w}$ (resp. by transformations preserving the rank order of $\boldsymbol{w}$).

Example 3.2.2. ***Winner-beats-loser*** *2-**tournament*** is a ranked birth-and-death selection method defined as follows: let A_1 and A_2 be independent uniform random variables on $\{1,2,\ldots,M\}$. The birth and death r.v.'s are given by

$$\begin{aligned} B(\boldsymbol{w}) = A_1, \quad D(\boldsymbol{w}) = A_2 \quad &\text{if } w_{A_1} > w_{A_2} \\ B(\boldsymbol{w}) = A_2, \quad D(\boldsymbol{w}) = A_1 \quad &\text{otherwise} \end{aligned} \tag{3.6}$$

(it may be checked that $B(\cdot), D(\cdot)$ thus defined are, as they must be, invariant under re-ordering - see Appendix A.2). Intuitively, to form a new population we pick two sequences at random from the current population. A mutant of the fitter then replaces the less fit.

It is not quite obvious that this selection operator is *not* elitist. For suppose there is exactly one sequence x in the population that is fitter than every other sequence in the population. Then it may happen that x is selected twice (i.e. $A_1 = A_2 = \alpha$, say, and $x = x_\alpha$). Then x "beats itself" and a mutant x', say, of x replaces x. Now if x' is less fit than x the best fitness in the population has decreased! Note that if we had demanded that $A_1 \neq A_2$ - i.e. that two *distinct* sequences must be chosen for a tournament - then we would have elitism.

Example 3.2.3. ***Moran selection*** is similar to the birth-and-death selection as introduced above, but is not of fixed population size (or fixed number of fitness evaluations). It is based on a continuous-time population genetics model introduced by (Moran, 1958). In the Moran model, for a population $\boldsymbol{x} =< x_1, x_2, \ldots, x_M >$ of size M with fitnesses $f(\boldsymbol{x}) = \boldsymbol{w}$, in each small time interval δt there is a probability $\lambda_\alpha(\boldsymbol{w}) \cdot \delta t + \boldsymbol{o}(\delta t)$ that the sequence x_α replicates and a probability $\mu_\alpha(\boldsymbol{w}) \cdot \delta t + \boldsymbol{o}(\delta t)$ that it dies for some $\lambda, \mu : \mathbf{R}^M \longrightarrow \mathbf{R}^M$. We may calculate that in the event of *either a birth or a death*, the probability that the event is a birth rather than a death is given by:

$$q(\boldsymbol{w}) = \frac{\lambda(\boldsymbol{w})}{\lambda(\boldsymbol{w}) + \mu(\boldsymbol{w})} \tag{3.7}$$

where we have set $\lambda(\boldsymbol{w}) = \sum_{\alpha=1}^{M} \lambda_\alpha(\boldsymbol{w})$ and $\mu(\boldsymbol{w}) = \sum_{\alpha=1}^{M} \mu_\alpha(\boldsymbol{w})$. We may also calculate that the waiting time $T(\boldsymbol{w})$ to the next birth/death event is *exponentially* distributed with expectation:

$$\mathbf{E}\left(T(\boldsymbol{w})\right) = \frac{1}{\lambda(\boldsymbol{w}) + \mu(\boldsymbol{w})} \tag{3.8}$$

To simulate the process, then, we draw waiting times till the next event from the distribution of $T(\boldsymbol{w})$ and suppose that the event is a birth with probability $q(\boldsymbol{w})$ or a death with probability

$1-q(\boldsymbol{w})$. In the case of a birth the probability that sequence x_α replicates is $\lambda_\alpha(\boldsymbol{w})/\lambda(\boldsymbol{w})$ while the probability that x_α dies in the case of a death is given by $\mu_\alpha(\boldsymbol{w})/\mu(\boldsymbol{w})$.

Example 3.2.4. ***Multinomial selection***, also commonly known as *roulette-wheel selection* operates as follows: selection is generational. To form the new population from a population of size M we perform M independent selections on the basis of fitness ("spins of the roulette wheel") from the current population. We thus have a function $\boldsymbol{p}\colon \mathbf{R}^M \longrightarrow \mathbf{R}^M$ where $p_\alpha(f(x_1), f(x_2), \ldots, f(x_M))$ represents the probability ("roulette-wheel sector size") that x_α is chosen on each independent selection. Since the p_α are probabilities, they also satisfy:

1. $0 \leq p_\alpha(\boldsymbol{w}) \leq 1 \; \forall \boldsymbol{w}, \alpha$
2. $\sum_{\alpha=1}^{M} p_\alpha(\boldsymbol{w}) = 1 \; \forall \boldsymbol{w}$

For $\boldsymbol{w} \in \mathbf{R}^M$ we then define the jointly (multinomially) distributed non-negative integer-valued random variables $R_1(\boldsymbol{w}), R_2(\boldsymbol{w}), \ldots, R_M(\boldsymbol{w})$ by:

$$\mathbf{P}\left(R_1(\boldsymbol{w}) = r_1, \ldots, R_M(\boldsymbol{w}) = r_M\right) = \frac{M!}{r_1! \ldots r_M!} p_1(\boldsymbol{w})^{r_1} \ldots p_M(\boldsymbol{w})^{r_M} \tag{3.9}$$

Intuitively $R_\alpha(f(x_1), f(x_2), \ldots, f(x_M))$ is the number of mutant replicas of x_α in the new population. The particular case where:

$$\boldsymbol{p}(\boldsymbol{w}) = \left(\frac{1}{\overline{w}}\right)\boldsymbol{w} \tag{3.10}$$

where $\overline{w} = \sum_{\alpha=1}^{M} w_\alpha$ yields a fitness-proportional multinomial selection operator also known as *Fisher-Wright selection* (Crow & Kimura, 1970). Other choices for $\boldsymbol{p}(\boldsymbol{w})$ allow for the possibility of rank-based multinomial selection, etc. Multinomial selection is of fixed population size and fixed number of fitness evaluations M.

Example 3.2.5. A ***stochastic hill-climber*** has a fixed population size of 1; a population is thus specified by a single sequence, which we call the *current* sequence[5]. To form the new population - i.e. to specify a new current sequence - we create a single mutant replica of the current sequence and then select on the basis of fitness between the current sequence and the new mutant[6]. We thus have a Bernoulli (Boolean) random variable (i.e. a biased coin toss) $Y(w, w')$, representing the event that the new mutant replaces the current sequence given that $f(current) = w$ and $f(mutant) = w'$. Stochastic hill-climbers have a fixed number of fitness evaluations of 1. An example is the ***netcrawler*** introduced in (Barnett, 2001). Here:

$$Y(w, w') = \begin{cases} true & w' \geq w \\ false & \text{otherwise} \end{cases} \tag{3.11}$$

The netcrawler, which is ranked and elitist, may be described thus: *"at each time step, if the new mutant is fitter than or of equal fitness to the current sequence, the mutant replaces the current*

[5] In the literature hill-climbers are frequently described as having population size 2 or "$1+1$"; see e.g. (Michaelewicz, 1996). Indeed, it might be argued, we surely need more than one sequence to be able to select non-trivially! However our definition of selection allows, of course, for the creation of (possibly transient) new mutants. Defining population size 1 will, as we shall see, also simplify the mathematics required to analyse evolutionary processes based on stochastic hill-climbers.

[6] We don't, in fact, demand that our hill-climbers always climb hills! They may accept fitness-decreasing or fitness-neutral mutants.

sequence; otherwise the mutant is discarded and the current sequence retained". The netcrawler thus always accepts *neutral* mutations as well as fitness-increasing mutations.

We note here that this algorithm is almost identical to the *Random Mutation Hill Climber* (RMHC) presented in (Forrest & Mitchell, 1993; Mitchell, Holland, & Forrest, 1994), the only difference being that the RMHC only ever flips one (randomly chosen) bit at each step - our constant 1-bit mutation mode. We avoid the term "hill-climber" to emphasise that, in the presence of neutral networks, a netcrawler spends most of its time not climbing hills, but rather performing *neutral walks* (Huynen et al., 1996).

A related stochastic hill-climber, which we shall refer to as the ***in-place*** stochastic hillclimber is defined by:

$$Y(w,w') = \begin{cases} true & w' > w \\ false & \text{otherwise} \end{cases} \tag{3.12}$$

In contrast to the netcrawler, the in-place hill-climber only moves to a new mutant if the mutant is strictly fitter than the current sequence.

A ***random walk*** is described simply by:

$$Y(w,w') = true \tag{3.13}$$

always. That is, the mutant always replaces the current sequence. In particular, if the mutation mode is *completely random mutation* (Section 2.2.1) then we have ***random search***.

Finally, we define a ***nervous ant neutral walk*** with *drift constant* $0 \leq q \leq 1$ by:

$$Y(w,w') = \begin{cases} Q(q) & w' = w \\ false & \text{otherwise} \end{cases} \tag{3.14}$$

where the Boolean r.v. $Q(q)$ is a biased coin toss which is $true$ with probability q. Thus a nervous ant neutral walk only accepts *neutral* mutants, and then only with fixed probability q. It remains on the neutral network on which it finds itself. The special case $q = 1$ - *always* move to a neutral mutant - is known as a ***blind ant neutral walk***[7] (Hughes, 1996).

Example 3.2.6. ***Multiple independent stochastic hill-climbers*** are precisely that: for population size M we have M *independent* Bernoulli random variables $Y_\alpha(w,w')$ representing the event that new mutant x'_α replaces current sequence x_α given that $f(x_\alpha) = w$ and $f(x'_\alpha) = w'$. Population size is fixed and number of fitness evaluations is M.

3.2.2 Evolutionary Processes

We are now ready to define what we mean by an evolutionary process. Recall that "τ" denotes time in generations:

Definition 3.2.1. Let $\mathcal{L} = (\mathcal{A}, L, f)$ be a fitness landscape over the sequence space $\mathcal{A}^L$. An ***evolutionary process*** on $\mathcal{L}$ is a Markov process $\boldsymbol{X}(\tau)$ on $\mathcal{P}(\mathcal{A}^L)$ of the following form: there is a

[7] On the subject of hill-climbers, we note that neither a *steepest ascent hill-climber* (Forrest & Mitchell, 1993) nor a *myopic ant neutral walk* (Hughes, 1996) employ evolutionary operators as we have defined them; in neither case do they employ *random* mutation and thus (to this author at least) it seems reasonable that neither process be described as "evolutionary". The *random mutation hill-climber* (RMHC) of (Mitchell et al., 1994), on the other hand, is just our *netcrawler* selection with uniform 1-bit constant mutation.

sequence $\mathcal{S}(\tau)$ of selection operators and a sequence $U(\tau)$ of mutation operators on $\mathcal{A}^L$ such that for $\boldsymbol{x},\boldsymbol{y} \in \mathcal{P}(\mathcal{A}^L)$ the transition probabilities are given by:

$$\mathbf{P}\left(\boldsymbol{X}(\tau+1)=\boldsymbol{x} \mid \boldsymbol{X}(\tau)=\boldsymbol{y}\right)=\mathbf{P}\left(\mathcal{G}(\tau)(\boldsymbol{y})=\boldsymbol{x}\right) \tag{3.15}$$

where $\mathcal{G}(\tau)=\mathcal{G}(\mathcal{S}(\tau),U(\tau))$ is the evolutionary operator induced by the corresponding selection and mutation operators[8].

If the $\mathcal{G}(\tau)$ are the same for all τ - that is, there is a selection operator $\mathcal{S}$ and a mutation operator U such that $\mathcal{G}(\tau)=\mathcal{G}(\mathcal{S},U)\ \forall\tau$ - then we call the evolutionary process ***(time) homogeneous***. A well-known example of a *non*-homogeneous evolutionary process is the following:

Example 3.2.7. Suppose given a decreasing function $T:\mathbf{N}\longrightarrow\mathbf{R}^+$ from the non-negative integers to the (positive) real numbers and a real parameter $k>0$. Given a fitness landscape $\mathcal{L}=(\mathcal{A},L,f)$ let U be a mutation operator on $\mathcal{A}^L$ and for $\tau=0,1,\ldots$ let $\mathcal{S}(\tau)$ be the stochastic hill-climber with Bernoulli selection $Y(\tau)(w,w')$ given by:

$$\mathbf{P}\left(Y(\tau)(w,w')\right)=\begin{cases}1 & w'\geq w\\ \exp\left(\frac{w'-w}{kT(\tau)}\right) & w'<w\end{cases} \tag{3.16}$$

Thus if the new mutant is fitter (or of equal fitness to) the current sequence it is accepted unconditionally. If the mutant is less fit than the current sequence it may still be accepted, with a probability that depends exponentially on the fitness decrement $w'-w$. Note that if $w'<w$ then, as the ***temperature*** $T(\tau)$ decreases, the argument $\dfrac{w'-w}{kT(\tau)}$ becomes larger and negative so that its exponential decreases. Thus the probability of accepting a given fitness decrement decreases over time. The evolutionary process defined by the evolutionary operators $\mathcal{G}(\tau)=\mathcal{G}(\mathcal{S}(\tau),U)$ is known as ***simulated annealing*** (Kirkpatrick et al., 1983; Catoni, 1996). The dependence of $T(\tau)$ on time τ is called the ***annealing schedule***. Note that as temperature $T(\tau)$ approaches zero, the behaviour of the process approaches that of the *netcrawler* (Eq. 3.11) - i.e. it accepts only non-fitness-decreasing mutants.

In general our evolutionary processes will be initiated at $\tau=0$, with $\boldsymbol{X}(0)$ the *initial population* (or *generation*), although sometimes $\tau=1$ may be more convenient. It is common (though not invariable) for evolutionary search processes to be initiated with a *random* initial population; that is, a population comprising M independent uniform random selections from $\mathcal{A}^L$. Whatever the origin of an initial population, we always assume that fitness must be evaluated for every initial sequence: *an initial population of size M always incurs exactly M fitness evaluations.*

3.3 Statistical Dynamics

In the previous section we defined an evolutionary process as a Markov process on the space of populations on a fitness landscape. A major obstacle to the mathematical analysis of such processes is the sheer size of the state space. Indeed, the number of possible populations for alphabet $\mathcal{A}$, sequence length L and population size M is of the order of $|\mathcal{A}|^{ML}$. An approach

[8] Recall our population size fudge of Section 3.2.1; it is assumed that $\mathcal{S}(\tau+1)$ is defined for the population resulting from application of $\mathcal{S}(\tau)$.

introduced in (Nimwegen et al., 1997) takes a cue from Statistical Mechanics by *coarse-graining* the state space and making a *maximum entropy assumption* regarding the distribution of states within the coarse-grained structure. Here we outline this ***Statistical Dynamics*** approach.

3.3.1 Coarse-graining and the Maximum Entropy Approximation

How are we to coarse-grain our state space? As defined in the previous section our evolutionary processes depend, via selection, only on the fitness of sequences in a population. In particular, selection as defined cannot differentiate sequences of *equal* fitness. Any partitioning of the state space should thus respect this aspect of selection, in the sense that sequences of different fitness should not be lumped together since they would be expected to behave differently with regard to selection. This was precisely the motivation for our introduction of *neutral partitionings* of a fitness landscape in the previous Chapter. However, as intimated there, we cannot expect that sequences of equal fitness will behave similarly with respect to *mutation*.

Thus suppose we are given a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$, a neutral partitioning $\mathcal{A}^L = \bigcup_{i=1}^{N} \Gamma_i$ of $\mathcal{L}$ with $f(x) = w_i$ for $x \in \Gamma_i$ and an evolutionary process $\boldsymbol{X}(\tau)$ on $\mathcal{L}$ with evolutionary operators $\mathcal{G}(\tau) = \mathcal{G}(\mathcal{S}(\tau), U(\tau))$. We suppose further that the mutation operators $U(\tau)$ are compatible with the partitioning. Now for a population $\boldsymbol{x} =< x_1, x_2, \ldots, x_M >\in \mathcal{P}^M(\mathcal{A}^L)$ of size M we can define $\tilde{\boldsymbol{x}} =< \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_M >\in \mathcal{P}^M(\widetilde{\mathcal{A}^L})$ where $\widetilde{\mathcal{A}^L} = \{1, 2, \ldots, N\}$ is as before the index set of neutral subspaces and angle brackets again indicate equivalence classes with respect to re-ordering. Thus $\widetilde{\boldsymbol{X}(\tau)}$ defines a stochastic process on $\mathcal{P}(\widetilde{\mathcal{A}^L})$ - but it is *not necessarily a Markov process*. For, while the probability of selecting a sequence from a neutral network does not depend on the particular sequence from that neutral network (since selection probabilities depend only on fitness), the probability that a sequence from a neutral network *mutates* to another neutral network may well depend on the particular sequence under consideration. In short, if U is a mutation operator and $x, y \in \mathcal{A}^L$ we have:

$$\tilde{x} = \tilde{y} \not\Rightarrow \widetilde{U}(x) = \widetilde{U}(y) \tag{3.17}$$

But, if we are lucky, Eq. (3.17) may hold "approximately" in the sense that for $x \in \Gamma_j$:

$$\widetilde{U}(x) \approx \widetilde{U}(X_j) \tag{3.18}$$

where X_j is *uniform* on Γ_j. That is to say, the probability that a given sequence from neutral network Γ_j mutates to a particular neutral network may be approximately the same as the probability that a sequence drawn *uniformly at random* from Γ_j mutates to the same neutral network.

Definition 3.3.1. Eq. (3.18) is the ***maximum entropy approximation*** for the mutation operator U and the given neutral partitioning.

It may be shown (see Appendix A.3) that we may "lift" an evolutionary operator $\mathcal{G}$ in a unique, natural way to act on populations $\mathcal{P}^M(\widetilde{\mathcal{A}^L})$ of (indices of) neutral networks. Intuitively, to calculate selection and mutation probabilities for $\tilde{\mathcal{G}}$ we identify each neutral network index j in a population of network indices with an (independent) uniform random sequence X_j on Γ_j. We then apply $\mathcal{G}$ to the "population of random sequences" thus obtained. As noted in Appendix A.3 mutation enters into the calculation of $\tilde{\mathcal{G}}$ simply via the mutation matrix for the neutral partitioning, since by

definition $\mathbf{P}\left(\widetilde{U}(X_j)=i\right)=\boldsymbol{m}_{ij}(U)$. The maximum entropy approximation says that the following diagram "almost commutes":

$$\begin{array}{ccc} & \tilde{\mathcal{G}} & \\ \mathcal{P}^M(\widetilde{\mathcal{A}^L}) & \longrightarrow & \mathcal{R}\left(\mathcal{P}(\widetilde{\mathcal{A}^L})\right) \\ \uparrow & & \uparrow \\ \mathcal{P}^M(\mathcal{A}^L) & \longrightarrow & \mathcal{R}\left(\mathcal{P}(\mathcal{A}^L)\right) \\ & \mathcal{G} & \end{array} \tag{3.19}$$

where $\mathcal{R}(\ldots)$ denotes "set of random variables with values in …".

Definition 3.3.2. We call the mapping $\tilde{\mathcal{G}}$ defined above the ***maximum entropy approximation of*** $\mathcal{G}$ for the given partitioning. Under the maximum entropy approximation (applied at each time step τ) an evolutionary process $\boldsymbol{X}(\tau)$ on $\mathcal{P}(\mathcal{A}^L)$ now defines a Markov process $\widetilde{\boldsymbol{X}}(\tau)$ on $\mathcal{P}(\widetilde{\mathcal{A}^L})$ as in Eq. (3.15) by:

$$\mathbf{P}\left(\widetilde{\boldsymbol{X}}(\tau+1)=\boldsymbol{i}\;\middle|\;\widetilde{\boldsymbol{X}}(\tau)=\boldsymbol{j}\right)=\mathbf{P}\left(\tilde{\mathcal{G}}(\tau)(\boldsymbol{j})=\boldsymbol{i}\right) \tag{3.20}$$

for $\boldsymbol{i},\boldsymbol{j}\in\mathcal{P}(\widetilde{\mathcal{A}^L})$, which we call the maximum entropy approximation of $\boldsymbol{X}(\tau)$ for the given partitioning[9].

The naming of the approximation derives from direct analogy with the parallel procedure in statistical mechanics: given a neutral network Γ we "forget" the precise distribution of sequences in Γ and then treat them as if they were drawn from a maximally disordered distribution - i.e. a distribution with maximum entropy - within the constraint that they are in Γ. The power of the maximum entropy approximation for an evolutionary process stems from the reduction in size of the state space $\mathcal{P}^M(\mathcal{A}^L)$ to the smaller and hopefully more tractable state space $\mathcal{P}(\widetilde{\mathcal{A}^L})$, a reduction in state space size from order of $|\mathcal{A}|^{LM}$ to order of N^M where N is the number of neutral subspaces in the partitioning.

We will often talk about the "maximum entropy assumption" instead of "approximation"; the assumption we are implicitly making is that the maximum entropy approximation for an evolutionary process really *is* in some sense a reasonable approximation to the actual evolutionary process. For instance, we may examine a "macroscopic" quantity such as $\mathbf{E}\left(\bar{f}(\boldsymbol{X}(\tau))\right)$, the expected mean fitness at time τ, and ask how well it is approximated (for a given neutral partitioning) by the corresponding quantity $\mathbf{E}\left(\bar{f}(\widetilde{\boldsymbol{X}}(\tau))\right)$.

Ultimately, as in statistical mechanics proper, how good an approximation we obtain with a maximum entropy approximation will probably need to be tested empirically. This we shall frequently do for the evolutionary processes to be analysed later. We might expect a trade-off between analytic tractability (the coarser the partitioning, the smaller the state space and hence the simpler the analysis) and accuracy of the approximation. Also, we note that in the previous Chapter we introduced the quantities $\mathcal{M}_j(U)$ and $\mathcal{M}(U)$ which, in some sense, measure the extent to which Eq. (3.18) applies and hence are an indication of how well a maximum entropy approximation to an evolutionary process might be expected to hold.

[9] Note the distinction between $\widetilde{\boldsymbol{X}}(\tau)$ and $\widetilde{\boldsymbol{X}(\tau)}$. Perhaps more properly, we should say that the Markov process $\widetilde{\boldsymbol{X}}(\tau)$ is (hopefully) an approximation to the not-necessarily-Markov process $\widetilde{\boldsymbol{X}(\tau)}$.

Finally we note that, while classical statistical mechanics addresses itself almost exclusively to *equilibrium* statistics (in our case *stationary* Markov processes) the evolutionary phenomena of interest to us (e.g. first passage times to specified fitness levels) tend to be essentially *non*-equilibrium; this, of course, makes the task of analysis no easier...

3.4 Epochal Dynamics

This thesis deals explicitly with fitness landscapes featuring a high degree of neutrality and some degree of correlation. The dynamics of evolutionary processes on such landscapes typically display some characteristic features (Fig. 3.1)[10] as identified in (Huynen et al., 1996; Nimwegen et al., 1997; Reidys et al., 1998; Barnett, 1997; Barnett, 1998; Smith et al., 2001; Harvey & Thompson, 1996), *etc.*

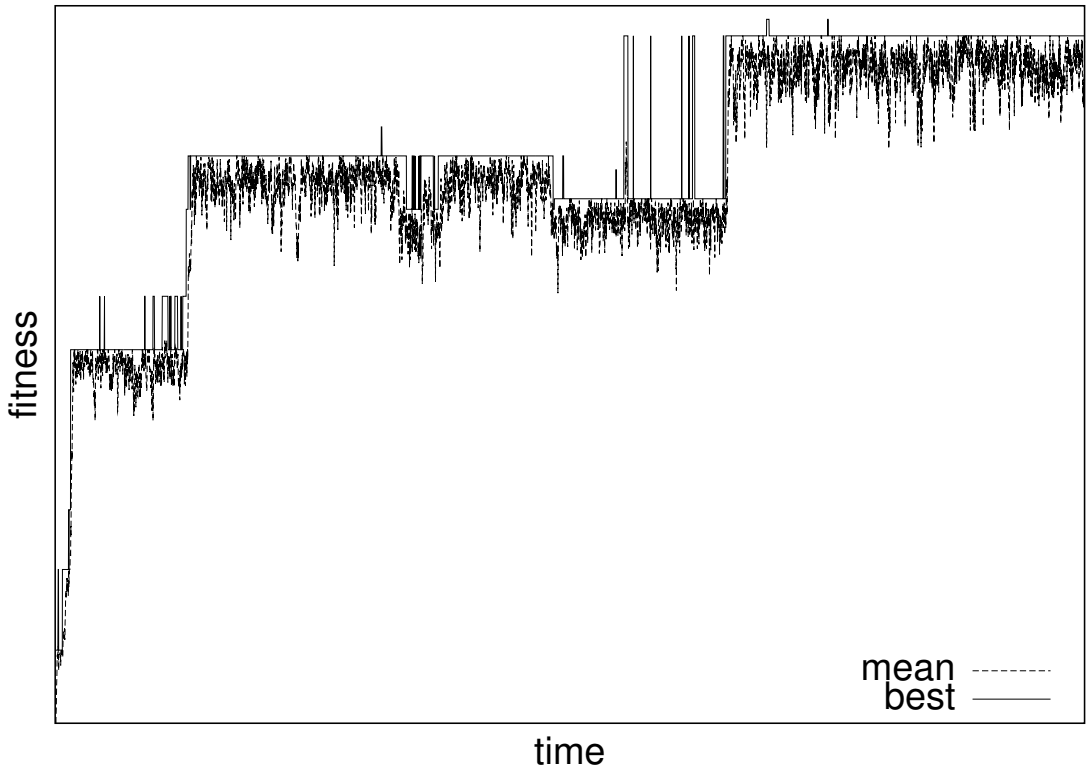

Figure 3.1: Typical evolutionary dynamics on a fitness landscape featuring neutral networks.

- Evolution proceeds by *fitness epochs* (Nimwegen et al., 1997), during which the mean fitness of the population fluctuates around a stable (quasi-)equilibrium. These mean fitness equilibria roughly track the population best fitness.
- Transitions to higher fitness epochs are preceded by the discovery of a higher fitness sequences than currently resides in the population[11].

[10] The landscape of Fig. 3.1 is an NKp landscape (Chapter 6), the evolutionary process "standard" fitness-proportional multinomial selection with Poisson mutation.

[11] The discovery of a new high fitness sequence is sometimes termed an *innovation*; however, use of this term is not consistent in the literature. In this work we have used the term "innovation" (Section 2.2.6) to denote the discovery of a *previously unseen* sequence, not necessarily of higher fitness.

- Transitions may be down to a lower fitness epoch as well as up; this is associated with loss of all current best fitness sequences.
- The discovery of a higher fitness sequence than the current best does not necessarily initiate a new epoch; the new sequence may be quickly lost before it can become established in the population[12]. This may be repeated several times before establishment of a new epoch.
- If a higher fitness sequence *does* initiate a fitness epoch, there is a transition period, brief compared to a typical epoch duration[13], during which the population mean fitness climbs to the new epoch level.

Through the work of various researchers a consistent explanation has emerged for the above characteristics:

- During a fitness epoch the population is localised in sequence space, somewhat like a classical *quasi-species* (Eigen, 1971; Eigen et al., 1989). The best fitness sequences reside on a neutral network, along which they *diffuse neutrally* (Kimura, 1983; Kimura, 1964; Huynen et al., 1996), until either...
- ... a *portal* sequence (Nimwegen & Crutchfield, 1999) to a higher fitness neutral network is discovered, or...
- ... the epoch *destabilises*; all sequences on the current highest neutral network are lost due to sampling noise; this phenomenon relates to the concept of the (finite population) *error threshold* (Eigen et al., 1989; Swetina & Schuster, 1982; Nowak & Schuster, 1989) (*cf.* Chapter 7).
- If a higher fitness portal sequence is discovered it will survive and drift to *fixation* (Maynard Smith, 1998) with a probability (Fisher, 1930; Kimura, 1962; Lande, 1985) and rate (Kimura & Ohta, 1969) dependent on the selective advantage of the portal sequence and the mutation rate of the evolutionary process.
- During the transient period when a portal sequence is fixating, the population becomes strongly converged genetically (this phenomenon is variously known in the literature as "hitch-hiking" or the "founder effect" (Mitchell, Forrest, & Holland, 1992; Forrest & Mitchell, 1993)), as the higher fitness portal sequence and its selectively neutral mutants are preferentially selected at the expense of lower fitness sequences.

Now a more "traditional" view of population evolution (with recombination, on a not-necessarily-neutral landscape) might impute a somewhat different interpretation to Fig. 3.1. It might be assumed that epochs correspond to episodes during which the population is entrapped in the vicinity of a local fitness sub-optimum, while transitions to higher fitness levels signify discovery of higher local peaks, somewhat in the manner of Sewall Wright's "shifting balance" theory (S. Wright, 1982). Broadly, the traditional GA picture promoted by John Holland (Holland, 1992) and subsequent researchers, might be characterised thus: recombination assembles fitness-enhancing "building blocks" present in the population into higher fitness sequences (the so-called *Building Block Hypothesis* (Forrest & Mitchell, 1993)); mutation is merely a "background operator" to prevent total loss of genetic diversity. This process continues as long as there is sufficient genetic diversity in the population for recombination to work with. Once genetic diversity has waned (inevitably

[12] This will not, of course, occur if selection is *elitist* (Section 3.2.1).
[13] Indeed so brief as to be virtually indiscernible on the time-scale of Fig. 3.1.

so, due to the combined effects of selection pressure and finite-population stochastic sampling) the population is deemed "converged" and no further fitness improvements are likely.

Thus it tends to be considered necessary to initiate the GA with a (usually large) randomly generated population - that is, with a "Big Bang"[14] of genetic diversity for recombination to work upon. This perception goes some way to explaining the obsession of much GA research with "premature convergence" and the multitudinous schemes prevalent in the literature for the avoidance thereof. In this author's view there are several serious flaws to this picture, particularly as regards evolution on landscapes with high neutrality; we discuss this in Chapter 7 on recombination. Our conclusions there lead us to reverse received wisdom and justify our view of *mutation* as the driving force behind evolutionary search. If recombination has a role to play we view it as secondary (and obscure!) and consequently exclude it from our analysis.

3.4.1 Analysing Epochal Dynamics - Fitness Barriers and Entropy Barriers

Referring to Fig. 3.1, during an epoch (i.e. during periods when transients associated with losing the current neutral network or moving to a higher network have subsided) the evolutionary process $\boldsymbol{X}(\tau)$ is, as a Markov process, "almost" *stationary* (Kampen, 1992); roughly speaking, the probability of finding the population in a given state does not vary over time. In (Nimwegen et al., 1997) an evolutionary process during such an episode is described as *metastable*. As an approximation we may consider a metastable evolutionary process $\boldsymbol{X}(\tau)$ as a (stationary) Markov process in its own right. In particular, the sub-population of sequences *on* the highest fitness network behave similarly to a population diffusing on a *flat* (i.e. selectively neutral) landscape; this situation is analysed to some extent in (Derrida & Peliti, 1991), where it is shown that neutrally diffusing populations exhibit a characteristic *clustered* structure, with sub-populations of sequences sharing common genealogies.

In the previous Chapter we noted that, given some correlation, higher fitness sequences are more likely to produce higher fitness mutants than sequences of lower fitness. In practice this means that in an evolving population during a metastable episode, we can expect portal sequences to be discovered, in the main, as mutants of the sub-population of sequences *diffusing on the current highest network*. Optimisation, then, is dominated by waiting times for this diffusing sub-population to search the neighbourhood of the highest network; the larger the volume of the network in sequence space (and the more sparsely distributed are portals) the longer we can expect to wait. (Nimwegen & Crutchfield, 1999) coin the term ***entropy barrier*** to describe the search obstacle presented by the volume of a neutral network under this scenario. He then contrasts this with the ***fitness barrier*** presented by entrapment of a population on a sub-optimal network, where portal discovery is dominated by the time required for a lineage of "off-network" sequences to cross a "ditch" of lower-fitness sequences. He then goes on to show that in general we can expect crossing entropy barriers to be faster by orders of magnitude than crossing fitness barriers; if portals from the highest network do indeed exist, he concludes, a population is more likely to discover these portals - cross the entropy barrier - than to cross a fitness barrier. His analysis also demonstrates that fitness barriers are most likely to be crossed where they are *narrowest* (i.e. require shorter lineages of off-network mutants) than where they are *shallowest* (i.e. where the

[14] This term appears to be due to Inman Harvey.

selective differential is smallest).

This thesis concerns itself explicitly with the case where entropy barriers dominate search; that is, where there is a high degree of neutrality (hence large neutral networks) and comparitively few locally sub-optimal networks (Chapters 5 and 6 introduce some models for such landscapes). The results of van Nimwegen *et al.* then suggest that the more efficiently we can search a network for neighbouring portals the more effective our search will be. In the next Chapter we argue that this is best achieved by maximising *neutral drift* of our population on the highest network.

As regards optimal mutation rates, in the previous Chapter we noted that for *individual* sequences on a neutral network, setting an optimal mode/rate is a balancing act between mutating off the network (where portals are to be found) but not *too* far so as to lose fitness correlation between parent and mutant offspring (Section 2.3.4). For *populations*, our picture of a drifting sub-population on the current fittest network during a metastable episode introduces a new tension, involving not just mutation rate but also *population size*:

- The *higher* the mutation rate and the *smaller* the population size, the higher the rate of drift and therefore the more thoroughly is the neighbourhood of a network searched.
- The *lower* the mutation rate and the *larger* the population size, the larger the search sub-population of sequences drifting on the highest network.

Finding an optimal mutation rate/population size must now balance both the individual sequence portal discovery probabilities with the above neutral drift effects. (Nimwegen & Crutchfield, 1998) analyse optimal parameters for populations evolving on a class of fitness landscapes featuring nested neutral networks[15]. They find a large "sweet spot" in the parameter space and calculates scaling laws for search times. These results are echoed in our analysis and experiments on similar landscapes in Chapter 5. (Nimwegen et al., 1997) also analyse the epochal dynamics for similar landscapes in some detail, using a maximum entropy approximation and a variety of statistical dynamics techniques. They are able to predict successfully various aspects of the dynamics such as epoch mean fitness and fitness variance, epoch destabilisation probabilities, portal fixation probabilities and fixation times. Predicting epoch *durations* tends to be more problematic; this seems to be due largely to inadequacies in our ability to analyse the structure of neutrally drifting populations (Derrida & Peliti, 1991).

3.5 Measuring Search Efficiency

How are we to measure the "efficiency" of an optimisation process? Firstly, as stressed earlier, it is almost inevitable for a non-trivial real-world optimisation problem that the most computationally expensive aspect of any search process is likely to be *fitness evaluation*. Therefore when comparing evolutionary processes we should always measure the *time-scale* of our search processes in terms of fitness evaluations, rather than generations[16]. If $\mathcal{G}$ is an evolutionary operator and $\boldsymbol{x}$ a

[15] The landscapes in this study are an instance of our ε-correlated landscapes introduced in Chapter 5.

[16] There is an implicit assumption here that fitness evaluation are performed *sequentially*. If we were to implement an evolutionary operator so that fitness evaluations were performed in *parallel* during the execution of a generation - quite feasible eg. for a multinomial selection operator - we should not, of course, measure time in fitness evaluations. Although our definition of a selection operator (Appendix A.1) does not dictate the synchronisation of fitness evaluation, we do not address parallel implementations in this thesis.

population of sequences we have defined the random variable $\|\mathcal{G}(\boldsymbol{x})\|$ to be the number of fitness evaluations *of new mutants* that we would have to perform in creating the next generation from $\boldsymbol{x}$ using $\mathcal{G}$. The reasoning is that mutation, as the only creator of novel sequences[17], dictates when fitness needs to be evaluated. The implicit assumption here is that evaluated fitness values may always be stored alongside a sequence for future reference[18].

What, then, is in fact the object of our optimisation quest? In much of the GA literature optimisation performance tends to be gauged, as a matter of course, on attainment of a *global* optimum of the given fitness landscape; thus search efficiency is judged on the time taken to find a global optimum, or the success rate in finding a global optimum within a pre-specified time. For real-world optimisation problems this approach appears unrealistic or even unfeasible; how, indeed, are we to *know* when we have attained a global optimum without knowing the solution to the problem in advance? Furthermore is a global optimum really what we are looking for, or would some minimum level of fitness suffice? As an example, if we wished to evolve a controller to facilitate the performance of a navigational task by a robot, the benchmark of success is likely to be *adequate* performance of the task rather than some putative *optimal* performance of the task. To this author's mind, it seems that the emphasis in the GA literature on locating global optima may be a result of the over-reliance on highly artificial "test functions" and toy problems by GA researchers; the agenda for the practitioner wishing to solve difficult practical problems may well be rather different.

Imagine, then, that we are given a succession of (unknown) fitness landscapes drawn from some class of optimisation problem. Then there may well be:

1. a *minimum acceptable fitness value*, w_c, say
2. a *maximum acceptable search time*, t_c, say (in fitness evaluations)

or (most likely) both. These quantities should be considered as *parameters* for benchmarking search performance. It may not be clear which search aspect - fitness or time - is likely to be the more critical; we might therefore consider these aspects separately. Suppose we are given an evolutionary process $\boldsymbol{X}(\tau)$ with evolutionary operators (i.e. Markov transition probabilities) $\mathcal{G}(\tau)$ (Section 3.2.2; recall that "τ" represents generational time). Suppose that the initial population $\boldsymbol{X}(0)$ has size M and that the number of fitness evaluations required to form the τ'th generation $\boldsymbol{X}(\tau)$ from the previous generation $\boldsymbol{X}(\tau-1)$ is given by the r.v. $\|\boldsymbol{X}(\tau)\|$. Note that $\|\boldsymbol{X}(0)\| = M$ and that $\|\boldsymbol{X}(\tau)\|$ is distributed as $\|\mathcal{G}(\tau-1)(\boldsymbol{X}(\tau-1))\|$ (*cf.* Eq. A.10 in Appendix A.1). The number of *fitness evaluations* after τ generations of the process is then:

$$T(\tau) = \|\boldsymbol{X}(0)\| + \|\boldsymbol{X}(1)\| + \ldots + \|\boldsymbol{X}(\tau)\| \tag{3.21}$$

Let $[\boldsymbol{X}(\tau)]$ be the best evaluated fitness in forming the population $\boldsymbol{X}(\tau)$, which is a r.v. distributed as $[\mathcal{G}(\tau-1)(\boldsymbol{X}(\tau-1))]$ (*cf.* Appendix A.1, Eq. A.15). For $t \geq 0$ we may then define the random

[17] The exception, as previously noted, is the *initial* population.

[18] Note that if fitness evaluation is *noisy* (Jakobi, Husbands, & Harvey, 1995) - that is, it is stochastic - this assumption becomes suspect. We then have several choices, dependent ultimately on exactly *what* we are trying to optimise (eg. *mean fitness*): we might, for instance, decide to re-evaluate fitness of every sequence in the population once per generation, as required by selection or (as before) only on creation.

variable $W^*(t)$ to be the *best fitness so far* of our process up to and including time t (measured in fitness evaluations):

$$W^*(t) = \max\{[\boldsymbol{X}(\tau)] \mid T(\tau) \leq t\} \tag{3.22}$$

For real w let the random variable $T^*(w)$ denote the *first passage time* of our process (again in fitness evaluations) to discovery of fitness w or higher; i.e.:

$$T^*(w) = \min\{T(\tau) \mid [\boldsymbol{X}(\tau)] \geq w\} \tag{3.23}$$

Let $P(w,t)$ denote the probability that fitness w has been achieved within t evaluations; i.e.:

$$P(w,t) = \mathbf{P}(W^*(t) \geq w) = \mathbf{P}(T^*(w) \leq t) \tag{3.24}$$

The expectations of $W^*(t)$ and $T^*(w)$ are then candidate "benchmarks" for optimisation performance:

1. $\mathbf{E}(W^*(t)) =$ mean best-so-far fitness after t evaluations
2. $\mathbf{E}(T^*(w)) =$ mean first passage time to achieve fitness w

These quantities may still not really be what we want. For instance if there really is a minimum acceptable fitness w_c, there is, perhaps, not much point in knowing $\mathbf{E}(T^*(w_c))$ if the *distribution* of $T^*(w_c)$ is unknown. Rather, we might want some measure of *how likely we are to achieve* minimum acceptable fitness within a given time. Thus we might, for instance, consider:

$$t^*(w) = \min\{t \mid P(w,t) \geq .95\} \tag{3.25}$$

which answers the question: *How long do we have to run our process to be* 95% *sure of reaching minimum acceptable fitness w ?* Conversely we might consider:

$$w^*(t) = \max\{w \mid P(w,t) \geq .95\} \tag{3.26}$$

to answer the question: *What is the maximum fitness that we can be* 95% *sure of achieving within maximum acceptable time t ?* The measures $\mathbf{E}(W^*(t))$, $w^*(t)$ might be termed ***time-critical***, the measures $\mathbf{E}(T^*(w))$, $t^*(w)$ ***fitness-critical***.

Finally, it is quite conceivable that both time and fitness are critical. So we might ask: *Given a minimum acceptable fitness w_c and a maximum acceptable number of fitness evaluations t_c what is the probability of achieving the fitness w_c within t_c evaluations?* This suggests simply using $P(w,t)$ as a benchmark measure, which we term ***time/fitness-critical***. Of course in practice the probability $P(w_c,t_c)$ may be unacceptably small!

When it comes to *direct comparison* of two search processes we might proceed as follows: suppose we have two candidate processes $\boldsymbol{X}(\tau)$ and $\boldsymbol{X}'(\tau)$. A time-critical comparison might be phrased as: *Given a maximum acceptable number of evaluations t, which process is likely to have achieved higher fitness within t evaluations?* Thus if:

$$\mathbf{P}\left(W^*(t) > W'^*(t)\right) > \mathbf{P}\left(W'^*(t) > W^*(t)\right) \tag{3.27}$$

we might consider choosing $\boldsymbol{X}(\tau)$ in preference over $\boldsymbol{X}'(\tau)$. A corresponding fitness-critical comparison might ask: *Given a minimum acceptable fitness w, which process is likely to achieve fitness*

w in the least number of fitness evaluations? Again, we might be inclined to prefer $\boldsymbol{X}(\tau)$ over $\boldsymbol{X}'(\tau)$ if:

$$\mathbf{P}\left(T^*(w) < T'^*(w)\right) > \mathbf{P}\left(T'^*(w) < T^*(w)\right) \tag{3.28}$$

Finally, if both time and fitness were critical we might prefer $\boldsymbol{X}(\tau)$ over $\boldsymbol{X}'(\tau)$ if:

$$P(w,t) > P'(w,t) \tag{3.29}$$

i.e. if $\boldsymbol{X}(\tau)$ is more likely to achieve fitness w in t evaluations than is $\boldsymbol{X}'(\tau)$.

In Chapter 5 and Chapter 6 in particular, search process comparisons arise and we shall use the above framework of time/fitness criticality of performance measurement. A slight difficulty in producing meaningful comparisons is that the number of evaluations $T(\tau)$ per τ generations may be a random variable or may, at the very least, vary from process to process (eg.with population size). This makes comparisons of processes sampled in simulation potentially somewhat complex. In practice we will generally "cheat" somewhat by treating each generation as an arbitrary (possibly variable-length) sequence of mutation/fitness evaluation events (this procedure is possible by Eq. (A.8) of Appendix A) and log best-so-far fitness as mutants are evaluated. For the processes considered (eg. fitness-proportional multinomial selection) this gives unambiguous results - i.e. the order of evaluations is not significant - since there is an equal chance of obtaining a new best-so-far fitness at each mutation/evaluation during the course of a generation.

Chapter 4

The Utility of Neutral Drift

In this Chapter we consider why neutral drift might benefit evolutionary search. Although (neutral) drift is generally associated with an evolving *population* - when a biologist speaks of neutral drift he generally means the fixation of an allele at some locus due to finite-population sampling rather than selective pressure (Kimura, 1964; Kimura, 1983; Maynard Smith, 1998) - essentially the same mechanism may apply to a population-of-1 evolutionary process with stochastic sampling, in the sense that replacement of an allele at some locus with a new allele that doesn't affect fitness may be thought of as a "neutral move" of a population-of-1 on a neutral network. In this Section we investigate neutral drift on a single neutral network using as an analytic tool a variety of *neutral walk* with "tunable" drift (*cf.* Example 3.2.5) on a single neutral network. We must, however, be careful in extrapolating our conclusions to (larger) populations - we raise some caveats at the end of this Section.

Consider an evolutionary process during an epoch (Section 3.4) where the highest neutral network thus far discovered is Γ, say. Suppose now that during the course of the process a mutant x' of some sequence $x \in \Gamma$ belonging to our population is created. Now if x' is fitter than Γ - x' is a *portal* (Section 3.4) to a higher-fitness neutral network Γ', say - it seems reasonable to *exploit* the discovery of Γ'; we might thus be inclined to design our evolutionary process so that x' makes it into the next generation, possibly at the expense of its parent sequence x. Our motivation for doing so lies in some assumption of fitness/distance correlation: we might expect the offspring of x' to be nearer in fitness - and thus of generally higher fitness - than subsequent offspring of x. If, conversely, x' is *less* fit than Γ then in general we might - again because of an assumption of correlation - be inclined to allow x to proceed into the next generation at the expense, perhaps, of its inferior offspring x'... of course we might sometimes be inclined - if for instance we had particular reason to believe that our current highest network Γ was sub-optimal - to accept lower-fitness mutants, but in general this would be exceptional; we are, after all, explicitly concerned with the situation where entropy rather than fitness barriers present the principal challenge to optimisation.

What, however, when the mutant x' is also on Γ - that is, it is *neutral*? Do we then have any reason either to accept the neutral mutant - possibly at the expense of its parent - or simply, rather, to consider it a wasted move and retain the parent? In other words should we encourage our population to *drift* on Γ in the interests of finding portals to higher fitness networks? Intuition seems

to suggest that we should indeed encourage drift. For if we always rejected neutral mutants at the expense of their parents our process would, it might seem, be in danger of becoming trapped in a neighbourhood of Γ where the probability of finding portals happens to be negligible or even zero - whereas there are in fact other regions of Γ, accessible via neutral drift, where the probabilities of finding portals are greater. We might envisage two extreme situations (real optimisation problems might be expected to fall somewhere between the two):

[I] portals from Γ to higher fitness networks are evenly spread throughout Γ; that is, the probability of finding a portal by mutation from $x \in \Gamma$ doesn't vary much with x

[II] some sequences in Γ offer a negligible chance of discovering a portal by mutation whilst other sequences offer a reasonable chance

In case [I] it clearly won't make much difference to our chances of finding a portal whether we drift - i.e. sample a variety of sequences from Γ - or not. As regards case [II], however, consider, the following scenario: our neutral network comprises just two sequences, x_1 and x_2 and is *connected* in the sense that the probability of mutating from one sequence to the other is non-zero. Suppose now that the probability of finding a portal - that is, discovering a higher fitness sequence by mutation - from x_1 is non-zero, while the probability of finding a portal from x_2 is zero. Now if we *knew* that our sequence x was in fact x_1 - or even that it was *likely to be* x_1 - then we should evidently *not*, should the opportunity arise - i.e. if $x' = x_2$ - encourage drift to x_2, since our chances of finding a portal from x_2 are zero. Conversely, if we knew that our sequence x was (or was likely to be) x_2 then we *should*, in the event of a neutral mutation, encourage drift.

But in a real-world optimisation problem chances are that we would have no idea of portal discovery probabilities for particular members of our population. In such circumstances we might be inclined to adopt a *maximum entropy*-like assumption, assuming minimal prior knowledge of portal discovery probabilities. In the above example, for instance, we would assume equal probabilities for $x = x_1$ and $x = x_2$. In this Section, then, we argue that, *under a maximum entropy assumption*, our original intuition was indeed correct - it is *always* better (in a precise if somewhat restrictive sense) to drift.

The reason why this might be so is hinted at by the following construction: let Π be the set of portals (i.e. higher fitness sequences) for our network Γ. Suppose now that we select a sequence X, say, uniformly at random from Γ and generate independently two mutants X_1 and X_2 of sequence X. Let us set $P =$ the probability that (at least) one of the mutants finds a portal $= \mathbf{P}(X_1 \in \Pi \cup X_2 \in \Pi)$ (inclusive "or"). Now let us choose independently *two* sequences Y and Y', say, uniformly at random from Γ. Let Y_1 be a mutant of Y and Y_2 an (independent) mutant of Y' (see Fig. 4.1). Let $P' =$ the probability that (at least) one of *these* mutants finds a portal $= \mathbf{P}(Y_1 \in \Pi \cup Y_2 \in \Pi)$.

Propostion 4.0.1. *$P' \geq P$ always.*

Proof. The proof is purely algebraic: suppose that Γ consists of M sequences $x_1, x_2, \ldots, x_M$ and let $p_\alpha = \mathbf{P}(U(x_\alpha) \notin \Pi)$ be the probability that a mutant of x_α does *not* find a portal, where U is the mutation operator. Then:

$$P = \mathbf{P}(X_1 \in \Pi \cup X_2 \in \Pi)$$

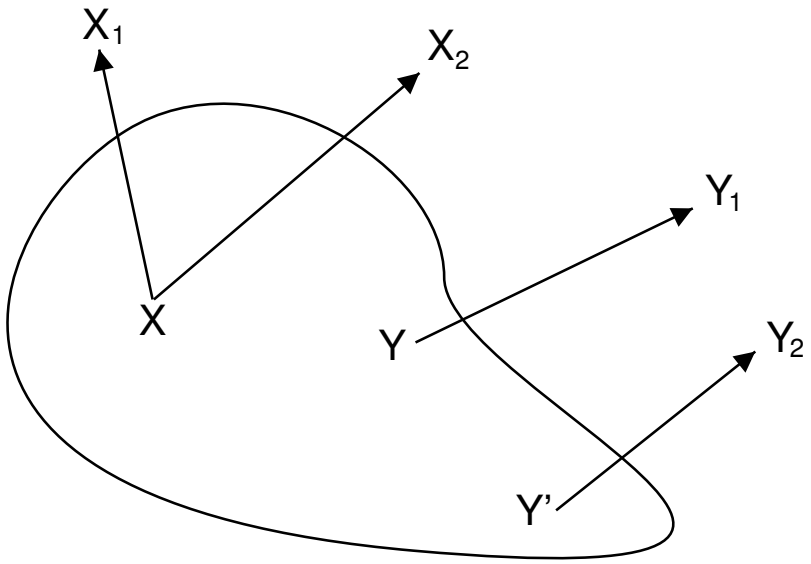

Figure 4.1: Independent mutants X_1, X_2 of the uniform random sequence X and independent mutants Y_1, Y_2 of the independent uniform random sequences Y, Y' respectively, on a neutral network Γ as in Prop. 4.0.1.

$$\begin{aligned}
&= \frac{1}{M}\sum_{\alpha=1}^{M} \mathbf{P}\left(X_1 \in \Pi \cup X_2 \ \in \Pi \mid X = x_\alpha\right) \quad \text{since } X \text{ is uniform on } \Gamma \\
&= 1 - \frac{1}{M}\sum_{\alpha=1}^{M} \mathbf{P}\left(X_1 \notin \Pi,\ X_2 \ \notin \Pi \mid X = x_\alpha\right) \\
&= 1 - \frac{1}{M}\sum_{\alpha=1}^{M} p_\alpha^2 \quad \text{since } X_1, X_2 \text{ are independent mutants}
\end{aligned}$$

On the other hand:

$$\begin{aligned}
P' &= \mathbf{P}\left(Y_1 \in \Pi \cup Y_2 \ \in \Pi\right) \\
&= 1 - \mathbf{P}\left(Y_1 \notin \Pi,\ Y_2 \ \notin \Pi\right) \\
&= 1 - \mathbf{P}\left(Y_1 \notin \Pi\right)^2 \quad \text{since } Y_1, Y_2 \text{ are completely independent} \\
&= 1 - \left(\frac{1}{M}\sum_{\alpha=1}^{M} p_\alpha\right)^2
\end{aligned}$$

Thus it suffices to show that:

$$\sum_{\alpha=1}^{M} p_\alpha^2 \geq \frac{1}{M}\left(\sum_{\alpha=1}^{M} p_\alpha\right)^2 \tag{4.1}$$

for any $p_\alpha \geq 0$ and $M = 1, 2, \ldots$. This we do by induction on M. Eq. (4.1) is certainly true for $M = 1$. Let us assume that it is true up to M. Some algebra gives:

$$\sum_{\alpha=1}^{M+1} p_\alpha^2 - \frac{1}{M+1}\left(\sum_{\alpha=1}^{M+1} p_\alpha\right)^2 = \sum_{\alpha=1}^{M} p_\alpha^2 - \frac{1}{M}\left(\sum_{\alpha=1}^{M} p_\alpha\right)^2 + \frac{M}{M+1}\left(\frac{1}{M}\sum_{\alpha=1}^{M} p_\alpha - p_{M+1}\right)^2$$

$$\begin{aligned} &\geq \sum_{\alpha=1}^{M} p_\alpha^2 - \frac{1}{M}\left(\sum_{\alpha=1}^{M} p_\alpha\right)^2 \\ &\geq 0 \quad \text{by the inductive hypothesis} \end{aligned} \tag{4.2}$$

so that Eq. (4.1) holds for $M+1$ and thence for all $M \geq 1$. □

From the construction of the proof we also see that *equality* holds (i.e. $P' = P$) iff the p_α are all equal, so that it does not matter "where we mutate from" - this is precisely situation [I] above. As a measure of the degree to it *does* matter where we mutate from, we may take the *evolvability drift factor* (Section 2.3.4, Eq. 2.72). In terms of the portal discovery probabilities $\pi_\alpha = 1 - p_\alpha$ it is given by:

$$\mathcal{D}^{evol} == 1 - \frac{\frac{1}{M}\sum_{\alpha=1}^{M} h(\pi_\alpha)}{h\left(\frac{1}{M}\sum_{\alpha=1}^{M} \pi_\alpha\right)} \tag{4.3}$$

where $h(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ is the entropy of a Bernoulli trial (biased coin-toss) with probability x. Thus if $\mathcal{D}^{evol} = 0$ then the π_α are all equal and it makes no difference where we mutate from.

Returning to Prop. 4.0.1, we would like to claim that in general the *less "related"* (by mutation) are a set of sequences, the *more likely* that (independent) mutants of those sequences are to find a portal. The above construction corresponds to the extreme case where the parents of X_1, X_2 are as related as can be - they are the same sequence! - while the parents of Y_1, Y_2 are as *un*-related as possible - they are independent. The above construction is, of course, naive. In the course of a realistic evolutionary process we would expect varying degrees of relatedness amongst sequences in (subsequent generations of) a population. We expand on this theme in the following Section.

4.1 The Nervous Ant

We now, as an analytic tool, introduce a neutral walk related to the so-called "blind ant" walk (Eq. 3.13). The current sequence resides on a neutral network Γ. At each time step an (independent) mutant is created of the current sequence. If the mutant is not on the network - i.e. the mutation is non-neutral - we keep the current sequence. If the mutant is on the network we move to the mutant with fixed probability q, where $0 \leq q \leq 1$. The *drift parameter* q thus tunes the degree to which the process diffuses on Γ: if $q = 1$ we have a blind ant neutral walk (maximum drift), while if $q = 0$ we have an "*in-place*" neutral walk. We term this a ***nervous ant*** walk (Eq. 3.14).

Thus suppose $\Gamma = \{x_\alpha \mid \alpha = 1, 2, \ldots, M\}$ comprises M sequences and we have a (compatible) mutation mode $\mathcal{U}$. Let the random variable $X(t)$ represent the current sequence on Γ and $Y(t) = \mathcal{U}(X(t))$ - not necessarily on Γ - the mutant, at time step $t = 1, 2, \ldots$. Thus for each t, $Y(t)$ depends only on $X(t)$ while $X(t+1)$ depends only on $X(t)$ and $Y(t)$. Note that $X(t)$ is a Markov process but $Y(t)$ is not. We suppose that:

$$\mathbf{P}\left(Y(t) = x_\alpha \mid X(t) = x_\beta\right) = \mathbf{P}\left(\mathcal{U}(x_\beta) = x_\alpha\right) = \mu_{\alpha\beta} \tag{4.4}$$

gives the probability of mutating (neutrally) to a specific sequence on Γ. The probability of a *neutral* mutation from sequence $x_\beta \in \Gamma$ is thus:

$$\mathbf{P}\left(Y(t) \in \Gamma \mid X(t) = x_\beta\right) = \nu_\beta = \sum_{\alpha=1}^{M} \mu_{\alpha\beta} \tag{4.5}$$

Note that by symmetry of mutation (Eq. 2.5) we have $\mu_{\beta\alpha} = \mu_{\alpha\beta}$ for all $\alpha, \beta = 1, 2, \ldots, M$. We also have $\mu_{11} = \mu_{22} = \ldots = \mu_{MM}$. The nervous ant walk is then defined by Eq. (4.4), Eq. (4.5) and:

$$\mathbf{P}\left(X(t+1) = x_\alpha \mid X(t) = x_\beta,\ Y(t) \notin \Gamma\right) = \delta_{\alpha\beta} \tag{4.6}$$

$$\mathbf{P}\left(X(t+1) = x_\alpha \mid X(t) = x_\beta,\ Y(t) = x_\gamma\right) = q\delta_{\alpha\gamma} + (1-q)\delta_{\alpha\beta} \tag{4.7}$$

Conditioning on $Y(t)$, the transition probabilities for $X(t)$ (considered as a Markov process) are easily calculated to be:

$$P_{\alpha\beta}(q) = \mathbf{P}\left(X(t+1) = x_\alpha \mid X(t) = x_\beta\right) = q\mu_{\alpha\beta} + (1 - q\nu_\alpha)\delta_{\alpha\beta} \tag{4.8}$$

or:

$$P(q) = q\Delta + I \tag{4.9}$$

where $\Delta = (\mu_{\alpha\beta} - \nu_\alpha\delta_{\alpha\beta})$[1] and I is the $M \times M$ identity matrix. Note that by mutational symmetry $P(q)$ is *bi-stochastic* (Seneta, 1973).

We now consider the following scenario: suppose that the set $\Pi \subseteq \mathcal{A}^L - \Gamma$ represents the portals from Γ to higher fitness neutral networks. Let us define the *portal discovery probabilities* to be:

$$\pi_\alpha = \mathbf{P}\left(\mathcal{U}(x_\alpha) \in \Pi\right) \tag{4.10}$$

(some of the π_α may be 0) and note that we must have:

$$\pi_\alpha + \nu_\alpha \leq 1 \quad \forall\alpha \tag{4.11}$$

Let us define the *first passage time to portal discovery* for a nervous ant walk with drift parameter q to be the random variable $T(q)$ defined by:

$$T(q) = \min\{t \mid Y(t) \in \Pi\} \tag{4.12}$$

We now calculate the distribution of $T(q)$ conditional on a particular initial sequence. Let us set:

$$f_\beta(t) = \mathbf{P}\left(T(q) > t \mid X(1) = x_\beta\right) = \mathbf{P}\left(Y(1) \notin \Pi, \ldots, Y(t) \notin \Pi \mid X(1) = x_\beta\right) \tag{4.13}$$

Conditioning on the (mutually exclusive) probabilities that $Y(1) = x_\alpha$ $(\alpha = 1, \ldots, M)$, $Y(1) \in \Pi$ or $Y(1) \in \mathcal{A}^L - \Gamma - \Pi$ we may derive the recursion relation:

$$f_\alpha(t+1) = \sum_{\beta=1}^{M} P_{\alpha\beta}(q)\, f_\beta(t) - \pi_\alpha\, f_\alpha(t) \tag{4.14}$$

where the $P_{\alpha\beta}(q)$ are given by Eq. (4.8). Let $Q(q) = P(q) - diag(\pi_1, \pi_2, \ldots, \pi_M)$ and let $\boldsymbol{f}(t)$ be the (column) vector $(f_\beta(t))$. We have (in vector/matrix notation):

$$\boldsymbol{f}(t+1) = Q(t) \cdot \boldsymbol{f}(t) \quad \text{for } t = 1, 2, \ldots \tag{4.15}$$

so that:

$$\boldsymbol{f}(t) = Q(q)^{t-1} \cdot \boldsymbol{f}(1) \tag{4.16}$$

[1] For the case where the mutation mode $\mathcal{U}$ is *fixed* - that is, mutation flips a fixed number of loci - we may consider the neutral network Γ as a graph where two vertices (i.e. sequences) are incident iff there is a non-zero probability of mutating from one to the other. Then $-\Delta$ is (up to a constant) the ***graph Laplacian*** (Stadler, 1996) of Γ - hence the notation.

Noting that $\sum_{\alpha=1}^{M} Q(q)_{\alpha\beta} = 1 - \pi_\beta = f_\beta(1)$, we may readily derive:

$$\mathbf{P}\left(T(q) > t \mid X(1) = x_\beta\right) = f_\beta(t) = \sum_{\alpha=1}^{M} \left[Q(q)^t\right]_{\alpha\beta} \tag{4.17}$$

for $t = 1, 2, \ldots$. If initial placement of the nervous ant is specified by:

$$\xi_\beta = \mathbf{P}\left(X(1) = x_\beta\right) \tag{4.18}$$

then we have:

$$\mathbf{P}\left(T(q) > t\right) = \sum_{\alpha=1}^{M} \left[Q(q)^t \cdot \xi\right]_\alpha \tag{4.19}$$

where $\xi = (\xi_\beta)$.

Now as has already been pointed out in the introduction to this Section (and as we shall see in more details below) portal discovery probabilities - specifically the (non-)discovery probability $\mathbf{P}\left(T(q) > t\right)$ - depend crucially on the initial placement probabilities ξ_β. What, however, can we know of the ξ_β? We are ultimately interested in the role of drift during an evolutionary search process. Imagine then, that at some stage during such a process a portal to the (previously unseen, higher fitness) network Γ is discovered. It might seem reasonable to suppose that we have *no* specific knowledge as to *which particular* sequence in Γ is likely to be discovered. Indeed the actual probability that a particular sequence be discovered may well depend on the neutral network structure of the landscape, the mutation operator, the evolutionary process under consideration, etc. In the spirit of our statistical dynamics approach then, we might (as intimated in the introduction) make a "maximum entropy"-like assumption that in the absence of *a priori* knowledge the initial sequence probabilities $\mathbf{P}\left(X(1) = x_\beta\right)$ should be taken to be equal.

There is, however, a reasonable alternative assumption we might make. (Nimwegen et al., 1999) state, regarding discovery of a portal sequence to a hitherto-unseen network[2]: "To a rough approximation, one can assume that the probability of a genotype ... being discovered first is proportional to the number of neighbours ... that [that genotype] has *off* the network". In our (comparable) scenario we might reasonably replace "number of off-network neighbours" with "probability of mutating to $\mathcal{A}^L - \Gamma - \Pi$" - on the grounds that the latter set is precisely that from which our "pioneer" sequence must have mutated; i.e. we might take:

$$\xi_\beta \propto 1 - \nu_\beta - \pi_\beta \tag{4.20}$$

Unfortunately it turns out to be difficult to draw any general conclusions under this assumption; indeed, the situation may be complex and counter-intuitive, as evidenced by the following example:

Example 4.1.1. Consider the 4-sequence neutral network $\Gamma = \{x_1 = 011,\ x_2 = 001,\ x_3 = 010,\ x_4 = 111\}$ in the sequence space $\{0,1\}^3$ under fixed 1-bit mutation and suppose that the portals comprise the single sequence $\Pi = \{110\}$ (see Fig. 4.2). We then have:

$$\mu = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \end{pmatrix} \tag{4.21}$$

[2] In this study mutation is fixed-probability 1-point.

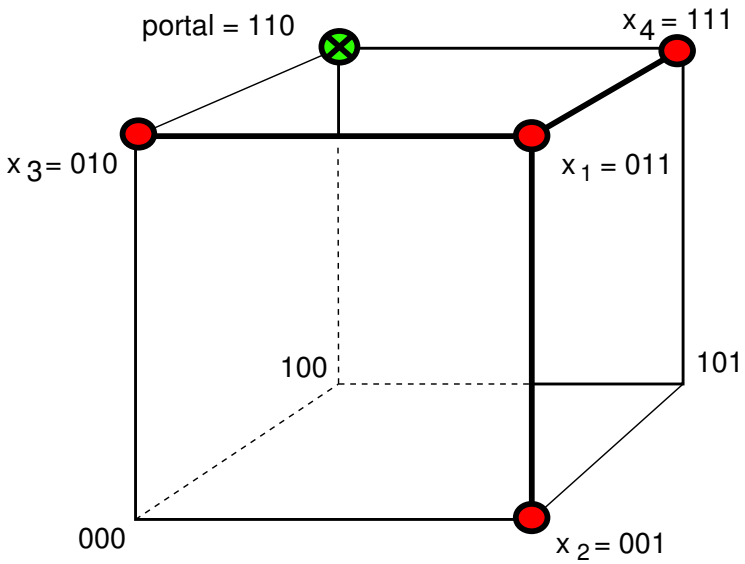

Figure 4.2: The 4-sequence neutral network plus portal (for fixed 1-bit mutation) of Example 4.1.1 on the cube $\{0,1\}^3$. Red nodes represent sequences on Γ, the green node represents the (single) portal in Π.

$$\nu = \left(1, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \tag{4.22}$$

$$\pi = \left(0, 0, \frac{1}{3}, \frac{1}{3}\right) \tag{4.23}$$

and Eq. (4.20) yields:

$$\xi = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{4} \end{pmatrix} \tag{4.24}$$

Fig. 4.3 plots $\mathbf{P}(T(q) > t)$ from Eq. (4.19) against q for this Γ, Π, for a range of t values. The result is somewhat unintuitive. Note that if we arrive on Γ at $x_3 = 101$ or $x_4 = 111$ then drift is not helpful; we are better off staying where we are. If we arrive on Γ at $x_2 = 001$ on the other hand, we are obliged to drift if we are to have any chance of finding the portal. The most striking conclusion to be drawn from this example is that whether drift is desirable or not *may depend on how long we are prepared to wait*. Thus for $t = 2$, the probability that we have still not found the portal after two evaluations increases (linearly) with drift q - drift is counter-productive; presumably the chance that we actually landed at x_3 or x_4 (and thence discover the portal) outweighs the probability that we landed at x_2 (and thus had no chance of finding the portal). As t increases the balance shifts: already by $t = 5$ it appears that we can no longer risk becoming "trapped" at x_2 and drift becomes desirable. For $t \geq 5$ we still have the somewhat curious situation that, even though $q = 1$ - maximum drift - is the best strategy for discovering a portal within t evaluations, increasing drift slightly from $q = 0$ is actually *counter*-productive. Portal discovery probabilities are not, as one might have suspected, monotonic with respect to drift. By contrast, consider the case (Fig. 4.4)

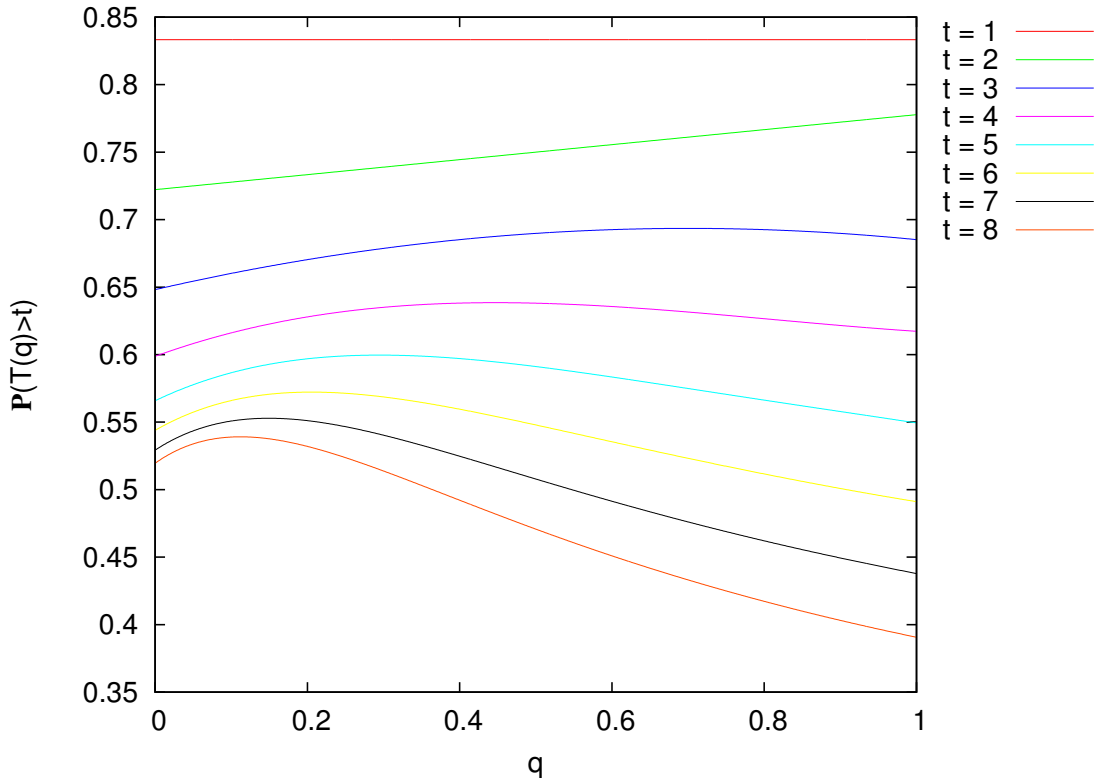

Figure 4.3: Portal (non-)discovery probability $\mathbf{P}(T(q) > t)$ of Eq. (4.19), plotted against drift parameter q for the Γ, Π of Example 4.1.1 with fixed 1-point mutation and off-network-proportional initial placement probabilities, for $t = 1, 2, \ldots, 8$. Note that, as regards discovery of portals, the *smaller* the probability $\mathbf{P}(T(q) > t)$ the *better*.

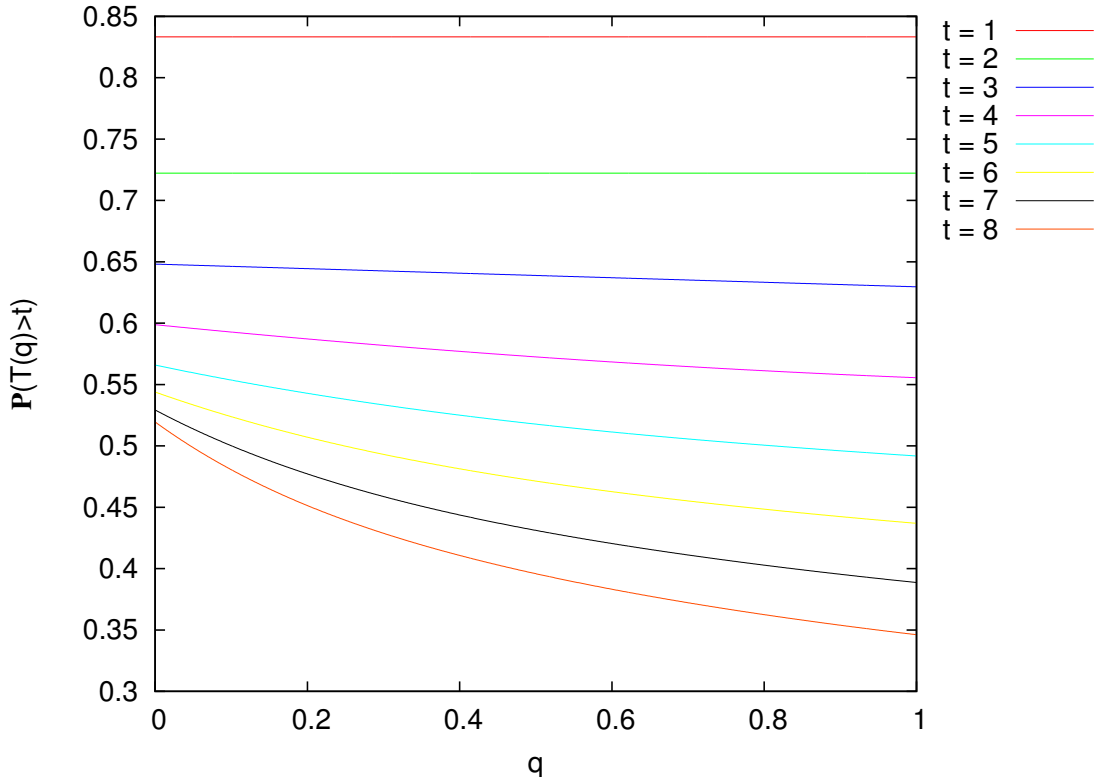

Figure 4.4: $\mathbf{P}(T(q) > t)$ plotted against q as in Fig. 4.4, except that here initial placement probabilities are equal ("maximum entropy" condition).

where we make a "maximum entropy" assumption of equal placement probabilities: i.e.:

$$\xi = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{pmatrix} \tag{4.25}$$

rather than Eq. (4.24). Here $\mathbf{P}(T(q) > t)$ *is* always monotonic (non-increasing) with respect to q: drift is *always desirable*.

Ultimately, we take the view here that in general there is insufficient evidence to justify the assumption Eq. (4.20) - more research is required as to if and when it might be warranted - and assume in general the "maximum entropy" initial placement condition:

$$\xi_\beta = \frac{1}{M} \quad \forall \beta \tag{4.26}$$

In this case it turns out that we can draw more concrete conclusions[3]. We shall, in fact, argue that the situation of Fig. 4.4 is *generic* - that drift is always desirable. We thus (unless otherwise stated) assume Eq. (4.26) in the remainder of this Section.

With Eq. (4.26), Eq. (4.19) becomes:

$$\mathbf{P}(T(q) > t) = \frac{1}{M} \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \left[Q(q)^t \right]_{\alpha\beta} \tag{4.27}$$

From Eq. (4.27) we may calculate, for $t = 1, 2, 3$:

$$\mathbf{P}(T(q) > 1) = \frac{1}{M} \sum_{\alpha=1}^{M} (1 - \pi_\alpha) \tag{4.28}$$

$$\mathbf{P}(T(q) > 2) = \frac{1}{M} \sum_{\alpha=1}^{M} (1 - \pi_\alpha)^2 \tag{4.29}$$

$$\mathbf{P}(T(q) > 3) = \frac{1}{M} \sum_{\alpha=1}^{M} (1 - \pi_\alpha)^3 - q \frac{1}{2M} \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \mu_{\alpha\beta} (\pi_\alpha - \pi_\beta)^2 \tag{4.30}$$

For the first two time steps, then, drift makes no difference to the probability that no portal is as yet discovered. By the third time step we see that increasing drift *always decreases* the probability that no portal is yet discovered; if we only had a maximum of three time steps at our disposal, we would thus certainly choose to maximise drift.

For the case $M = 2$ - a neutral net comprising just two sequences - we may calculate portal discovery probabilities explicitly for all t:

$$Q(q) = \begin{pmatrix} 1 - \pi_1 - u & u \\ u & 1 - \pi_2 - u \end{pmatrix} \tag{4.31}$$

where we assume Γ to be connected - i.e. $\mu_{12} > 0$ - and we have set $u = \mu_{12} q$. For convenience we set:

$$\frac{1}{2}(\pi_1 + \pi_2) = \phi \tag{4.32}$$

$$\frac{1}{2}(\pi_2 - \pi_1) = \psi \tag{4.33}$$

[3] It would, of course, be churlish to suggest that this might be reason in itself to concentrate on this case...

The eigenvalues of $Q(q)$ may then be calculated to be $K \pm L$ where:

$$K = 1-\phi-u \tag{4.34}$$
$$L = +\sqrt{\psi^2+u^2} \tag{4.35}$$

Note that by Eq. (4.11) we have $0 \leq \pi_1+\mu_{12} \leq 1,\ 0 \leq \pi_2+\mu_{12} \leq 1$ so that $0 \leq u \leq \min(1-\pi_1, 1-\pi_2)$ and thus $K \geq 0$ for all u. Diagonalising $Q(q)$ we may calculate:

$$f_1(t) = \frac{1}{2}\left[\left(1+\frac{u+\psi}{L}\right)(K+L)^t + \left(1-\frac{u+\psi}{L}\right)(K-L)^t\right] \tag{4.36}$$
$$f_2(t) = \frac{1}{2}\left[\left(1+\frac{u-\psi}{L}\right)(K+L)^t + \left(1-\frac{u-\psi}{L}\right)(K-L)^t\right] \tag{4.37}$$

And, as a function of $u = \mu_{12}q$:

$$\mathbf{P}(T(q) > t) = F(t,u) = \frac{1}{2}\left[\left(1+\frac{u}{L}\right)(K+L)^t + \left(1-\frac{u}{L}\right)(K-L)^t\right] \tag{4.38}$$

Propostion 4.1.1. *For fixed $t = 1,2,\ldots$ the $F(t,u)$ of Eq. (4.38) is monotone decreasing as a function of $u = \mu_{12}q$ for $0 \leq u \leq \min(1-\pi_1, 1-\pi_2)$.*

Proof. We know that for fixed t, $F(t,u)$ is a polynomial (of order $\leq t$) in u. We shall show that $\frac{\partial F(t,u)}{\partial u} \leq 0$ for u in the stated range, thus proving the proposition. We have:

$$\frac{\partial K}{\partial u} = -1$$
$$\frac{\partial L}{\partial u} = \frac{u}{L}$$

For $t = 1$ the result is trivial. For larger t we may calculate:

$$\begin{aligned}
\frac{\partial F(t+1,u)}{\partial u} &= -\frac{L^2-u^2}{2L^3}\left[(tL-K)(K+L)^t(tL+K)(K-L)^t\right] \\
&= -\frac{\psi^2}{2L^3}\sum_s\left[(tL-K)+(-1)^s(tL+K)\right]\binom{t}{s}K^{t-s}L^s \\
&= -\frac{\psi^2}{L^2}\left\{\sum_{s\ even} t\binom{t}{s}K^{t-s}L^s - \sum_{s\ odd}\binom{t}{s}K^{t-(s-1)}L^{s-1}\right\} \\
&= -\frac{\psi^2}{L^2}\sum_{s\ even}\left[t\binom{t}{s}-\binom{t}{s+1}\right]K^{t-s}L^s \\
&= -\frac{\psi^2}{L^2}\sum_{s\ even} s\binom{t+1}{s+1}K^{t-s}L^s \\
&\leq 0
\end{aligned}$$

since $K, L \geq 0$. □

Thus we have, for $M = 2$ that $\mathbf{P}(T(q) > t)$ is always monotone decreasing as a function of q. We interpret this to say: *for any network of size 2, for any time t, the probability that no portal is discovered within t fitness evaluations decreases with increasing drift q*. We conjecture that this is the case for *any* neutral network - that it *always* pays to maximise drift. We thus state:

Conjecture 4.1.1 (The [Strong] Neutral Drift Conjecture). *For any M and for $t = 1,2,\ldots$, $\mathbf{P}(T(q) > t)$ is monotone decreasing as a function of q for $0 \leq q \leq 1$.*

Note, again, that the uniform random initial condition is essential to the conjecture (recall the example in the introduction to this Section, where $M = 2$, $\pi_1 > 0$ and $\pi_2 = 0$). In fact, numerical calculations indicate that if initial conditions are *not* uniform random then for given t the value $q = q^*(t)$ which minimises $\mathbf{P}(T(q) > t)$ depends on t and may be 0, 1 or - somewhat counter-intuitively - we may even have $0 < q^*(t) < 1$ so that there is some intermediate degree of drift that optimises the chances of discovering a portal. Even for $M = 2$ the situation is complex; it may be calculated that, in the non-uniform random case, Eq. (4.38) becomes:

$$\mathbf{P}(T(q) > t) = F(t,u) = \frac{1}{2}\left[\left(1 + \frac{u+\psi\theta}{L}\right)(K+L)^t + \left(1 - \frac{u+\psi\theta}{L}\right)(K-L)^t\right] \tag{4.39}$$

where:

$$\theta = \xi_1 - \xi_2 \tag{4.40}$$

For given t we may calculate $q^*(t)$ (at least numerically) by solving $\frac{\partial F(t,u)}{\partial u} = 0$ for $u = \mu_{12}q$ with $F(t,u)$ given by Eq. (4.39).

Returning to the case of uniform random initial conditions, we have already demonstrated that Conjecture 4.1.1 holds for $t \leq 3$ and arbitrary M, while Prop. 4.1.1 proves the conjecture for the case $M = 2$ and arbitrary t. We have not managed to prove the general case; we do, however, prove the somewhat weaker:

Theorem 4.1.1 (The [Weak] Neutral Drift Theorem). *For any M and for $t = 1, 2, \ldots$, there is a $q_1 > 0$ such that $\mathbf{P}(T(q) > t)$ is monotone decreasing as a function of q for $0 \leq q \leq q_1$.*

We first establish the following:

Lemma 4.1.1. *For any integer $t \geq 2$ and any real u, v:*

$$(t+1)u^t - 2\sum_{s=0}^{t} u^s v^{t-s} + (t+1)v^t = (u-v)^2 \sum_{s=1}^{t-1} s(t-s)u^{s-1}v^{t-s-1} \tag{4.41}$$

Proof. Expand $(u-v)^2$ and gather like powers of u, v. □

Proof of Theorem 4.1.1. We have already demonstrated that the result (in fact the Strong Neutral Drift Conjecture) holds for $t = 1, 2, 3$. Now note that for fixed t, $\mathbf{P}(T(q) > t)$ as given by Eq. (4.27) is a polynomial of order $\leq t$ in q; in particular it is *analytic* in q. Note also that since $Q(0) = I - D$, $\mathbf{P}(T(0) > t) = \frac{1}{M}\sum_{\alpha=1}^{M}(1-\pi_\alpha)^t \geq 0$ for any t. To establish the Weak Neutral Drift Theorem it thus suffices to show that for $t > 3$ we have $\left[\frac{\partial}{\partial q}\mathbf{P}(T(q) > t)\right]_{q=0} \leq 0$. For convenience, let us set $p_\alpha = 1 - \pi_\alpha$. We then have $Q(q) = qU + diag(p_1, \ldots, p_M)$ where $U_{\alpha\beta} = \mu_{\alpha\beta} - \nu_\alpha\delta_{\alpha\beta}$ so that $\left[\frac{\partial}{\partial q}Q(t)\right]_{q=0} = U$. For $t = 2, 3, \ldots$ we then have:

$$\begin{aligned}
\left[\frac{\partial}{\partial q}\mathbf{P}(T(q) > t)\right]_{q=0} &= \frac{1}{M}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\left[\frac{\partial}{\partial q}\left(Q(q)^t\right)_{\alpha\beta}\right]_{q=0} \\
&= \frac{1}{M}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} U_{\alpha\beta}\left(p_\alpha^{t-1} + p_\alpha^{t-2}p_\beta + \ldots + p_\beta^{t-1}\right) \\
&= \frac{1}{M}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} \mu_{\alpha\beta}\left(p_\alpha^{t-1} + p_\alpha^{t-2}p_\beta + \ldots + p_\beta^{t-1} - tp_\beta^{t-1}\right)
\end{aligned}$$

after some re-arrangement of terms. Noting the symmetry of the $\mu_{\alpha\beta}$, we swap indices and add the resulting summations, to derive (for $t = 3, 4, \ldots$):

$$\begin{aligned}\left[\frac{\partial}{\partial q}\mathbf{P}(T(q) > t)\right]_{q=0} &= -\frac{1}{2M}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\mu_{\alpha\beta}\left(t p_\alpha^{t-1} - 2\sum_{s=0}^{t-1} p_\alpha^s p_\beta^{t-1-s} + t p_\beta^{t-1}\right) \\ &= -\frac{1}{2M}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\mu_{\alpha\beta}(p_\alpha - p_\beta)^2\sum_{s=1}^{t-2} s(t-1-s)p_\alpha^{s-1}p_\beta^{t-s-2}\end{aligned}$$

by Lemma 4.1.1. Since the terms in the summation over s are all ≥ 0, we have thus shown that $\left[\frac{\partial}{\partial q}\mathbf{P}(T(q) > t)\right]_{q=0} \leq 0$ for $t \geq 3$. □

The generality of this result is quite striking; it is purely algebraic, holding regardless of the structure of the network Γ, mutation mode and portal discovery probabilities. We remark furthermore that, as regards the Strong Neutral Drift Conjecture, extensive numerical simulation by the author failed to discover any exceptions. In summary, we propose:

> *For any neutral network, from a uniform random start, increasing neutral drift always improves our chances of finding a portal within any given number of fitness evaluations.*

4.1.1 The independent mutant approximation

If our neutral network Γ is *connected* with respect to the extant mutation operator (Section 2.1.1) then "in the long run" our nervous ant process (provided $q > 0$) spends asymptotically equal amounts of time - that is, makes an equal number of attempts at finding a portal - at every sequence on the network (Hughes, 1996). More specifically:

Propostion 4.1.2. *If Γ is connected and $q > 0$ then as $t \to \infty$, $\mathbf{P}(X(t) = x_\alpha) \to \frac{1}{M}$ for all $x_\alpha \in \Gamma$.*

Proof. If Γ is connected then the matrix $(\mu_{\alpha\beta})$ is *irreducible* (Gantmacher, 1960); so, thus, is the matrix $(P_{\alpha\beta}(q))$ of transition probabilities (Eq. (4.8)) for the nervous ant process $X(t)$. The result then follows immediately from stochasticity of $(P_{\alpha\beta}(q))$. □

In the long run then, mutation has a "mixing" effect on the selection of sequences. We may thus, for large times, approximate the nervous ant process by a process that at each time step picks a sequence *independently* and uniformly at random from Γ, then evaluates a mutant of that sequence. Provided that the waiting time $T(q)$ to portal discovery is reasonably large - a portal is not discovered too quickly - we may thus approximate the distribution of waiting time to portal discovery by the *geometric* distribution:

$$\mathbf{P}(T(q) > t) \approx (1 - \bar{\pi})^t \tag{4.42}$$

independently of q, where:

$$\bar{\pi} = \frac{1}{M}\sum_{\alpha=1}^{M}\pi_\alpha \tag{4.43}$$

is the mean of the portal discovery probabilities. We might expect the approximation Eq. (4.42) to be better if the portal discovery probabilities are small (so that waiting times are long), if the neutral

mutation probabilities ν_α do not vary much over the network - i.e. Γ is reasonably "homogeneous" - and the drift parameter q is large.

Note that if Γ is connected and $q > 0$ then from irreducibility of $(P_{\alpha\beta}(q))$ it may be deduced that the *mean* of $T(q)$ exists and, provided the conditions outlined in the previous paragraph obtain, may be approximated (independently of q and the network structure of Γ) by:

$$\mathbf{E}(T(q)) \approx \frac{1}{\bar{\pi}} \tag{4.44}$$

Diffusion coefficient

We may define the *diffusion coefficient cf.* (Huynen et al., 1996; Barnett, 1997) for the nervous ant by:

$$D_0 = \mathbf{E}\left(\left\langle h(X(t+1),X(t))^2 \right\rangle\right) \tag{4.45}$$

where $h(\cdot,\cdot)$ is Hamming distance and angle brackets denote time average. Setting $h_{\alpha\beta} = h(x_\alpha, x_\beta)$ we find[4]:

$$\begin{aligned} D_0 &= \mathbf{E}\left(\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T} h(X(t+1),X(t))^2\right) \\ &= \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} h_{\alpha\beta}^2 \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T}\mathbf{P}\left(X(t+1)=x_\alpha, X(t)=x_\beta\right) \\ &= \sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} h_{\alpha\beta}^2 P_{\alpha\beta}(q) \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T}\mathbf{P}\left(X(t)=x_\beta\right) \quad \text{from Eq. (4.8)} \end{aligned}$$

But by Prop. 4.1.2 $\mathbf{P}\left(X(t)=x_\beta\right) \to \frac{1}{M}$, so that, from Eq. (4.8):

$$D_0 = \frac{1}{M}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} h_{\alpha\beta}^2 P_{\alpha\beta}(q) = q\cdot\frac{1}{M}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M} h_{\alpha\beta}^2 \mu_{\alpha\beta} \tag{4.46}$$

since the contribution of the $\delta_{\alpha\beta}$ terms vanishes. We see immediately that the diffusion coefficient is proportional to the drift parameter q. The proportionality factor depends on the topology of the network and the mutation mode. If the mutation mode is given by $\mathbf{P}(\mathcal{U}=n) = u_n$, then we have (*cf.* Eq. 2.4):

$$\mu_{\alpha\beta} = \sum_{n=0}^{L}\delta(n,h_{\alpha\beta})\binom{L}{n}^{-1} u_n \tag{4.47}$$

and we can re-write Eq. (4.46) as:

$$D_0 = q\cdot\sum_{n=0}^{L} n^2 \nu^{(n)} u_n \tag{4.48}$$

where:

$$\nu^{(n)} = \frac{1}{M}\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\delta(n,h_{\alpha\beta})\binom{L}{n}^{-1} \tag{4.49}$$

is just the *n-flip neutrality* of the network. In particular, if the mutation mode is constant n-bit then we have simply:

$$D_0 = q\nu\bar{u}^2 \tag{4.50}$$

[4] Assuming we may exchange the order of taking the limit and summation...

where $\nu = \nu^{(n)}$ is the observed neutrality - i.e. the neutrality for constant n-bit mutation - and $\bar{u} = n$ is the per-sequence mutation rate. For Poisson mutation in the long sequence length limit, we may approximate $\nu^{(n)} \approx \left(\nu^{(1)}\right)^n$ where $\nu^{(1)}$ is the 1-bit neutrality - essentially we are neglecting *back mutation* to the network - and we may calculate that:

$$D_0 \approx q\nu(\bar{u} + \log \nu)(\bar{u} + \log \nu + 1) \tag{4.51}$$

where $\bar{u}$ is the per-sequence mutation rate and $\nu \approx e^{-\left(1-\nu^{(1)}\right)\bar{u}}$ is the observed neutrality.

4.2 Neutral Drift in Populations

We have concentrated so far on population-of-1 hill-climbers. In this case, whether or not a mutant survives at the expense of its parent is an "either/or" issue. This is not, of course, the case for evolutionary processes featuring larger (or even variable-sized) populations; the situation then is less clear-cut (*cf.* Section 3.4.1 in the previous Chapter). A proposal comparable to Conjecture 4.1.1 might be that *whatever* the evolutionary process, we should never re-mutate, or indeed retain for subsequent mutation, the *parent* of a neutral mutant; we should rather retain the neutral mutant for subsequent mutation (in the next generation). In the next Chapter we shall (with due care) invoke Conjecture 4.1.1 to argue along these lines for a specific class of fitness landscape.

We note also that for large populations there is no analog of Prop. 4.1.2: to the contrary, for large populations (at reasonably high mutation rates) the phenomenon of *mutational buffering*, or the *evolution of mutational robustness* (A. Wagner & Stadler, 1999; Wilke, 2001) implies that sequences in a population drifting on a neutral network will be found preferentially at regions of *higher neutrality* of the network. (Nimwegen et al., 1999) have shown that in the infinite population limit (for Poisson mutation and multinomial selection) the *population neutrality* (i.e. the mean neutral degree of sequences in the population) approaches the *spectral radius* of the network, considered as a (connected) graph. This raises some intriguing issues as to where portals may be found in relation to the local neutrality in a neutral network. Some new research (Bullock, 2002) suggests that for the related question of discovery of *innovations*, under certain conditions (eg. some neutral networks on RNA folding landscapes) mutational buffering of a population drifting under multinomial selection appears actually to improve the innovation rate (Section 2.2.6) compared with a (drifting) population of hill-climbers which sample a neutral network uniformly according to Prop. 4.1.2. There is much scope for future research on these issues.

Chapter 5

ε-Correlated Landscapes

5.1 Introduction

In this Chapter we make rather more specific structural assumptions about our fitness landscape. A common feature of fitness landscapes in artificial evolution is that *fitness-increasing* mutations are rare compared to neutral or fitness-decreasing mutations; indeed, if this were not the case, then optimisation would be a trivial matter. Since we are dealing with *correlated* landscapes, it also seems reasonable to suppose (and indeed appears to be the case for many artificial evolution landscapes) that mutations leading to a *large* fitness increase will be rarer than those (already rare) mutations leading to a *small* fitness increase. We are, in addition, interested specifically in fitness landscapes where entropy rather than fitness barriers (Section 3.4.1) are the chief obstacle to discovery of fitness-increasing mutation. The ε-correlated landscapes of this Chapter, first introduced in (Barnett, 2001), formalise these properties: fitness-increasing mutation probabilities - the *evolvability* statistics (Section 2.3.4) - are specified by an order parameter ε, assumed to be $\ll$ neutral/fitness-decreasing mutation probabilities. We assume further that if the neutral networks in our landscape are ordered by fitness, then the probability of mutation from a given neutral network to the "next-highest" network is non-zero (there are thus no fitness barriers) and of order ε, while the probability of mutation to any yet higher network is negligible by comparison; i.e. is $\boldsymbol{o}(\varepsilon)$. We note that ε-correlation[1] is a rather stringent condition. The next Chapter addresses a more general family of landscapes where the interplay of correlation and neutrality is more explicit; they may also feature sub-optimal networks.

ε-correlated landscapes then, present a "ladder" to evolutionary optimisation, which proceeds via epochs spent searching the successive neutral networks for portals to the "rung" above; ε-correlated landscapes are, in this sense, generalisations of the Royal Road family of landscapes introduced (albeit with very different motivation) in (Mitchell et al., 1992). In particular, the statistical dynamics techniques applied in (Nimwegen et al., 1997) to the analysis of population evo-

[1] The appearance of the term "correlation" in the naming of these landscapes (Barnett, 2001) may strike the reader as somewhat peculiar. We might, however, think of ε-correlation as relating to the degree of genotype-fitness correlation for *fitness-increasing* mutation; i.e. the degree to which a sequence nearby in sequence space to a given sequence is likely to be of similar fitness *given that the nearby sequence increases fitness*. Thus there is a small but non-zero probability that a point-mutation from any neutral network leads to a (probably small) increase in fitness, while the probability of a larger fitness increase is of a smaller order of magnitude.

lution on the Royal Road (and related) landscapes transfer wholesale to ε-correlated landscapes.

We now state the condition more precisely as follows[2]:

Definition 5.1.1. Let $\mathcal{L} = (\mathcal{A}, L, f)$ be a fitness landscape and let $\mathcal{A}^L = \bigcup_{i=1}^{N} \Gamma_i$ be the *maximal* neutral partitioning. Suppose that $f(x) = w_i$ for $x \in \Gamma_i$ and that the neutral networks are listed in order of increasing fitness: i.e. $w_1 < w_2 < \ldots < w_N$. We say that $\mathcal{L}$ is ε-***correlated*** iff there exists an ε with $0 < \varepsilon \ll 1$ and ε_j with $0 < \varepsilon_j \leq \varepsilon$ for $j = 1, 2, \ldots, N-1$ such that the 1-point mutation matrix $\boldsymbol{m} = \boldsymbol{m}^{(1)}$ takes the form[3]:

$$\boldsymbol{m} = \begin{pmatrix} \nu_1 & & & & \\ \varepsilon_1 & \nu_2 & & * & \\ & \varepsilon_2 & \nu_3 & & \\ & \boldsymbol{o}(\varepsilon) & \ddots & \ddots & \\ & & & \varepsilon_{N-1} & \nu_N \end{pmatrix} \tag{5.1}$$

The neutralities ν_i and the $*$ terms (i.e. the mutation probabilities from higher to lower fitness networks) in the above are *not* taken to be necessarily $\ll 1$ (i.e. of $\boldsymbol{o}(1)$ in ε). ■

The portal discovery probability ε_j is the probability that a point mutation takes a sequence selected uniformly at random from Γ_j to the "next neutral network up", Γ_{j+1}. Since by assumption the ε_j are all positive, there are no locally suboptimal neutral networks for 1-point mutation on an ε-correlated landscape, in the sense that for any neutral network there is always a portal to the next-highest network.

From Eq. (5.1) and Eq. (2.11) of Section 2.2.2 we may verify that the *relative volumes* $\upsilon_j = |\mathcal{A}|^{-L}|\Gamma_j|$ of the neutral networks satisfy:

$$(1 - \nu_j)\upsilon_j = [\varepsilon_{j-1} + \boldsymbol{o}(\varepsilon)]\,\upsilon_{j-1} \tag{5.2}$$

for $j = 2, 3, \ldots, N$, so that $\upsilon_j = \boldsymbol{O}(\varepsilon_1 \varepsilon_2 \ldots \varepsilon_{j-1}) = \boldsymbol{o}\left(\varepsilon^{j-2}\right)$ and neutral network size scales similarly to the networks in, eg., (Nimwegen & Crutchfield, 1998). We remark that ε-correlated landscapes thus exhibit another typical feature of (non-trivial) artificial evolutionary fitness landscapes: *the proportion of sequences of a given fitness diminishes rapidly with increasing fitness*. It is interesting to note that this follows directly from the scaling of portal discovery probabilities.

For $n \ll L$ we now adopt the (strong) multiplicative mutation approximation (Eq. 2.36) of Section 2.2.4. Then if $\mathcal{L}$ is ε-correlated it is easy to show that for $n = 2, 3, \ldots$ the n-point mutation matrix is given by:

$$\boldsymbol{m}^{(n)} \approx \boldsymbol{m}^n = \begin{pmatrix} \nu_1^{(n)} & & & & \\ \varepsilon_1^{(n)} & \nu_2^{(n)} & & * & \\ & \varepsilon_2^{(n)} & \nu_3^{(n)} & & \\ & \boldsymbol{o}(\varepsilon) & \ddots & \ddots & \\ & & & \varepsilon_{N-1}^{(n)} & \nu_N^{(n)} \end{pmatrix} \tag{5.3}$$

[2] Note that the definition given here differs slightly from that given in (Barnett, 2001).

[3] More properly, we should say that there is a parametrised family $\mathcal{L}(\varepsilon)$ of fitness landscapes such that quantities written as $\boldsymbol{o}(\varepsilon)$ are understood to be $\boldsymbol{o}(\varepsilon)$ as $\varepsilon \to 0^+$. Nevertheless, we shall continue to talk of "*an* ε-correlated landscape" and treat ε simply as a "small quantity" $\ll 1$. We shall frequently then work to leading order in ε.

where $\nu_j^{(n)} = \nu_j^n + \boldsymbol{o}(1)$ and the (approximate) n-point mutation portal discovery probabilities $\varepsilon_j^{(n)}$ are given by:

$$\varepsilon_j^{(n)} = \begin{cases} \frac{\nu_j^n - \nu_{j+1}^n}{\nu_j - \nu_{j+1}} & \varepsilon_j + \boldsymbol{o}(\varepsilon) & \nu_j \neq \nu_{j+1} \\ n\nu_j^{n-1} & \varepsilon_j + \boldsymbol{o}(\varepsilon) & \nu_j = \nu_{j+1} \end{cases} \tag{5.4}$$

Thus we have e.g. for Poisson mutation U with per-sequence mutation rate $\bar{u} \ll L$ the portal discovery probabilities:

$$\varepsilon_j(U) = \begin{cases} \frac{\exp\left(-\left(1-\nu_j\right)\bar{u}\right) - \exp\left(-\left(1-\nu_{j+1}\right)\bar{u}\right)}{\nu_j - \nu_{j+1}} & \varepsilon_j + \boldsymbol{o}(\varepsilon) & \nu_j \neq \nu_{j+1} \\ \nu_j \bar{u} \exp\left(-\left(1-\nu_j\right)\bar{u}\right) & \varepsilon_j + \boldsymbol{o}(\varepsilon) & \nu_j = \nu_{j+1} \end{cases} \tag{5.5}$$

For a general (compatible) mutation operator U, the $\varepsilon_j(U)$ and hence also the *evolvability* $\mathcal{E}\left(U \middle| w_j\right)$ (Section 2.3.4) at fitness w_j is evidently $\boldsymbol{O}(\varepsilon_j)$.

5.2 Optimal Mutation Rate

From Eq. (5.4) we may calculate immediately the *optimal (constant) mutation rate* of Section 2.3.4, Prop. 2.3.1 (Barnett, 2001), by treating n as a continuous variable, differentiating the expression Eq. (5.4) for $\varepsilon_j^{(n)}$ with respect to n and setting the derivative to zero. We may similarly calculate the optimal per-sequence mutation rate for Poisson mutation by differentiating Eq. (5.5) with respect to $\bar{u}$. We have:

Propostion 5.2.1. *Let us define:*

$$u_j^* = \begin{cases} \frac{\log\lambda_j - \log\lambda_{j+1}}{\lambda_j - \lambda_{j+1}} & \nu_j \neq \nu_{j+1} \\ \frac{1}{\lambda_j} & \nu_j = \nu_{j+1} \end{cases} \tag{5.6}$$

for $j = 2,3,\ldots,N-1$, where we have set $\lambda_j = -\log\nu_j$. Then the (constant) mutation rate n_j^* which maximises the probability of mutating from Γ_j to Γ_{j+1} is given[4] *by either $\lfloor u_j^* \rfloor$ or $\lceil u_j^* \rceil$, whichever maximises $\varepsilon_j^{(n)}$. "Usually" we will simply have:*

$$n_j^* = [u_j^*] \tag{5.7}$$

for $j = 2,3,\ldots,N-1$. Similarly, for Poisson mutation the optimal per-sequence mutation rate $\bar{u}_j^*$ for Γ_j is given by:

$$\bar{u}_j^* = \begin{cases} \frac{-\log\left(1-\nu_j\right) + \log\left(1-\nu_{j+1}\right)}{\nu_j - \nu_{j+1}} & \nu_j \neq \nu_{j+1} \\ \frac{1}{1-\nu_j} & \nu_j = \nu_{j+1} \end{cases} \tag{5.8}$$

for $j = 2,3,\ldots,N-1$. ∎

For Γ_1, the lowest-fitness network, our method breaks down since, because $\nu_1 = 1 + \boldsymbol{o}(1)$ in ε, the expression for $\varepsilon_1^{(n)}$ seems to imply that the optimal n should be as large as possible. In fact our strong multiplicative mutation assumption is untenable here; we cannot assume $n \ll L$. Now

[4] Recall that for real x, $\lfloor x \rfloor$ is the largest integer smaller than or equal to x, $\lceil x \rceil$ is the smallest integer greater than or equal to x and $[x]$ is the nearest integer to x.

according to Prop. 2.3.1 there *is* an optimal (constant) mutation rate, but we do not have enough structural information to determine it. Since, however, the relative volume of the lowest network Γ_1 is $\boldsymbol{O}(1)$ in ε, we can (in lieu of more detailed structural information) do no better than *random search* - i.e. completely random Poisson mutation (Section 2.2.1) - for sequences in Γ_1.

For the *highest* network Γ_N, of course there are no fitness-increasing mutations! We thus (formally) take the "optimum" mutation mode to be trivial (i.e. null) mutation.

Corollary 5.2.1.1. *The optimum mutation operator U adapted to the maximal neutral partitioning on an ε-correlated fitness landscape is that with $\mathcal{U}_j$ = constant mutation with rate n_j^* given by Prop. 5.2.1 for $j = 2,3,\ldots,N-1$, random search for $j = 1$ and null mutation for $j = N$.* ∎

Note that (somewhat surprisingly) the expressions for u_j^* and $\bar{u}_j^*$ are *symmetric* in the neutralities ν_j, ν_{j+1}. Fig. 5.1 plots the n_j^* of Prop. 5.2.1 against a range of neutralities. In Section 5.4 below,

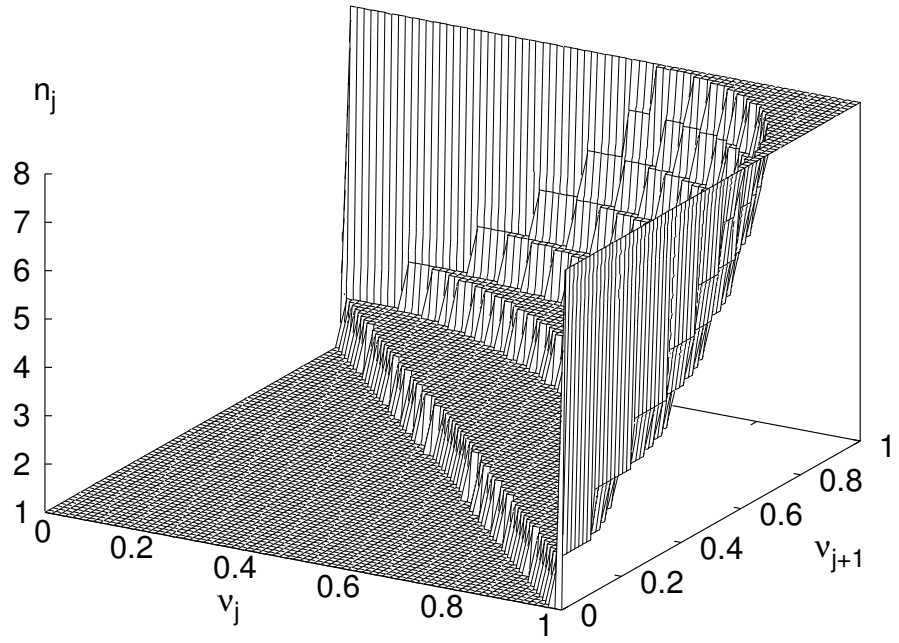

Figure 5.1: The optimum constant mutation rate $n_j^* = [u_j^*]$ of Prop. 5.2.1 plotted against ν_j, ν_{j+1}

we present results of simulations on a specific example of ε-correlated landscapes which strongly support Prop. 5.2.1

5.3 Optimal Evolutionary Search

Our object in this Section is to determine an evolutionary process that minimises the (expected) number of fitness evaluations to reach any given epoch. The argument, which proceeds in several

stages, is not mathematically rigorous - we hope that it is, at least, convincing. Consider thus an evolutionary process (Def. 3.2.1) on $\mathcal{L}$, currently in epoch j:

1. Firstly, it is clear that mutation should always maximise the probability that a fitter sequence than the parent is discovered. This implies that (at every time step) the mutation mode for a sequence in Γ_j should be $\mathcal{U}_j =$ constant mutation with rate n_j given by Prop. 5.2.1; i.e. the (compatible) mutation operator U should be that of Corollary 5.2.1.1. It remains to find an optimal selection procedure.

2. When in epoch j, the probability that mutation finds a portal to a neutral network *higher* than Γ_{j+1} is $\boldsymbol{o}(\varepsilon)$ and therefore negligible.

3. When in epoch j, the probability that *more than one* mutant is a portal to Γ_{j+1} is $\boldsymbol{o}(\varepsilon)$ and therefore negligible.

4. During the (sequential) evaluation of sequences for a generation of epoch j, if mutation discovers a portal to Γ_{j+1}, then *only that portal sequence* should be selected to the new population *and the next generation should commence immediately*. For having found a portal, there is no point in evaluating mutants of sequences of *lower* fitness than that portal sequence; such evaluations will be wasted - we would be better off evaluating mutants of the new-found portal. Thus, for every $j > 1$, epoch j is initiated with a population comprising a *single* sequence in Γ_j. (We return later to the case of epoch $j = 1$ - the "initialisation" of the process.)

5. Thus consider the situation at the *onset* of epoch j, where the population comprises a single sequence (our new-found "pioneer" portal sequence) in Γ_j. During execution of the evolutionary operator for the first generation in epoch j we are going to begin by evaluating a mutant of our single portal sequence. There are then three possibilities:

   I. If the mutant is in Γ_{j+1} we have found a portal sequence in one evaluation. As in the previous step of our argument there is no point then in evaluating further mutants of the original sequence during the same generation, so we select the mutant, disregard the parent sequence and initiate a new generation (with the single newly-discovered portal sequence) in epoch $j+1$.

   II. If the mutant is of *lower* fitness, it should not, by the arguments given above, be selected and there is, furthermore, no point in evaluating any of *its* mutants. We thus disregard the low-fitness mutant and initiate a new generation with a population comprising just our original "pioneer" sequence. We are then back in the same situation as before.

   III. If the mutant is *neutral* - i.e. also on Γ_j - we are faced with a choice: we can either create further mutants of the original portal sequence or we can choose to create mutants of the new neutral mutant. We argue that in this circumstance *we should always select the neutral mutant and disregard the original portal sequence*.

   Our arguments in support of this last claim are those of the previous Chapter, that we should maximise neutral drift; in particular Prop. 4.0.1 that the "more independent" are two mutants the more likely is portal discovery and the (Strong and Weak) Neutral Drift propositions (Conjecture 4.1.1 and Theorem 4.1.1) for the nervous ant process. These arguments are, it must be conceded, not rigorous since the scenarios of Chapter 4 do not correspond entirely with our present situation. Nonetheless, as we shall see, the arguments appear to stand up well in practice. Note also that, unlike the prior steps in our argument, we are obliged to appeal beyond any maximum entropy assumption to establish this point; indeed, under a maximum entropy assumption it should make no difference which (neutral) mutant we choose to evaluate...

6. By extrapolation of the above argument we see that every time a neutral mutant is created we should proceed by choosing to mutate that neutral mutant, rather than any of its (neutral) ancestors. This implies that having created a neutral mutant we may as well forget about selecting any of its neutral ancestors - we may, in other words, simply select the neutral mutant and begin a new generation with a population comprising *just* the newly created neutral mutant.

7. Putting the last three steps together, we find that we are left precisely with our *netcrawler* selection operator of Chapter 3, Example 3.2.5,

Returning to the initialisation stage, we note that, since the relative volume of the lowest network is $1+\boldsymbol{o}(1)$ in ε, then in the absence of more detailed information about our landscape *any* choice of initial sequence (prior to discovery of higher fitness sequences) is likely to be in Γ_1 with probability ≈ 1; we can do no better than random search. Noting that a netcrawler with completely random mutation performs random search, we thus state the following:

Conjecture 5.3.1. *The optimum evolutionary search process on an ε-correlated fitness landscape is a netcrawler with mutation operator given by Corollary 5.2.1.1 of Prop. 5.2.1* ∎

It is curious to note that the smaller the Markov coefficient $\mathcal{M}_j(U)$ for the neutral network Γ_j and optimal mutation operator - and thus the better the maximum entropy assumption is likely to obtain - the less compelling is the crucial argument in step 5 above, in the sense that it becomes less important which of several neutral mutants we choose to mutate and evaluate. Intuitively, it becomes less important to "explore" such a neutral network (by neutral drift) since we are as likely to find portals in one region of the network as another... nonetheless there is nothing to be lost by choosing an (optimal) netcrawler, which has the added advantage of being extremely simple and efficient to implement.

5.3.1 Adaptive netcrawling and the $1/e$ Neutral Mutation Rule

The astute reader will no doubt have realised that, given a fitness landscape and told only that it is (or may be) ε-correlated, *we cannot actually run* the optimal netcrawler of Conjecture 5.3.1 on our landscape - for the simple reason that we don't know the neutralities ν_j, or indeed anything about the neutral partitioning! The situation is not irretrievable, however. While we don't know neutralities *a priori*, we can, during the running of a netcrawler, *estimate* neutralities by treating the neutral/non-neutral mutations that we see as a statistical sample of the current neutral network during any given epoch. A problem still remains: Prop. 5.2.1 (somewhat curiously, it may seem) implies that to calculate the optimal (constant) mutation rate for our current (best-fitness-so-far) neutral network we need to know the neutrality not just of that network, but also of the next, higher, *as yet undiscovered* - and hence un-sampled - neutral network! We are thus forced to make further assumptions as to the structure of our fitness landscape. A conservative assumption might be that the next network up from wherever we are has *zero* neutrality. This would lead us to set a mutation rate at $n = 1$ bits always (giving us precisely Forrest and Mitchell's Random Mutation Hill-climber).

Another option might be to suppose that there is some correlation between fitness and neutrality, in the sense that neutral networks of similar fitness are likely to have similar neutrality. In

particular we might take the view that, given a neutral network, then (for lack of better knowledge) the next network up might be assumed to have the *same* neutrality as our current network. (Of course once we have discovered the higher network we can begin to sample it and revise our estimate of its neutrality.) This assumption has some interesting consequences. Suppose that we mutate sequences on a neutral network Γ at constant mutation rate n, where (unknown to us) Γ has actual neutrality ν. Suppose that after a good many mutations we have observed a fraction p to be neutral. By Eq. (5.3) we may then assume that $\nu \approx p^{1/n}$. If we are assuming that the neutrality of the next network up from Γ is *also* ν then according to Eq. (5.6), if we wished to optimise our chances of discovering a portal, we should re-adjust our mutation rate to $-1/\log\nu \approx -n/\log p$. Curiously this implies that, *whatever* the neutrality of Γ, we should, if our mutation rate is optimal, ultimately see a fraction $p = 1/e \approx 0.368$ of neutral mutations. It easy to show that the same argument works too for optimal Poisson mutation. We thus state:

Propostion 5.3.1 (The $1/e$ Neutral Mutation Rule (Barnett, 2001)). *The optimal (constant or Poisson) mutation rate for sequences evolving on an ε-correlated landscape is that for which the observed neutrality is $\approx 1/e$.* ■

If anything we should expect the rule to *overestimate* the optimal rate somewhat, since neutrality might be expected to decrease with increasing fitness (*cf.* Section 5.4.1 below and also Chapter 6).

The reader might note echoes of Rechenberg's "1/5 success rule" for *Evolution Strategies* (Back, Hoffmeister, & Schwefel, 1991) (to which our netcrawler might be considered a discrete cousin). In Section 5.4 below we provide evidence for the effectiveness of the rule for a specific family of ε-correlated landscapes. We in fact conjecture that the $1/e$ Neutrality Rule is a useful general heuristic for setting mutation rates for evolution on fitness landscapes with neutral networks. In the next Chapter we present another argument for the rule and find good evidence for its effectiveness on a more general class of landscapes.

We thus propose an ***adaptive netcrawler*** (Barnett, 2001) for landscapes which we know (or at least suspect) to be ε-correlated and for which the assumption that fitness correlates with neutrality turns out to be reasonable: we either use the $1/e$ rule to adjust the mutation rate - if we observe the fraction of neutral mutations to be less than $1/e$ we increase the rate, if more than $1/e$ we decrease it - or by using Eq. (5.6) to calculate a mutation rate based on a sampled estimate of neutrality. Later in this Chapter we shall test these schemes on some specific ε-correlated landscapes.

5.3.2 Random search on ε-correlated fitness landscapes

In the next sub-section we shall analyse the netcrawler on ε-correlated landscapes in more detail. In the following section we analyse some specific examples of ε-correlated fitness landscapes which we shall use to test Conjecture 5.3.1. Firstly, as a basis for comparison, we analyse *random search* on ε-correlated landscapes. Thus at each time step $t = 1, 2, \ldots$ (where a time step represents a single fitness evaluation) we select a sequence uniformly at random from the sequence space. Let $Z(t)$ be the network of the t-th random selection let $X(t)$ be the fittest-network-so-far in t time steps. Note that the $Z(t)$ are iid as Z where:

$$\mathbf{P}(Z = i) = \upsilon_i \tag{5.9}$$

for $i = 1, 2, \ldots, N$ and that:

$$X(t) = \max(Z(1), Z(2), \ldots, Z(t)) \tag{5.10}$$

We also define T_i to be the first passage time (in fitness evaluations) of the process $X(t)$ to network i or higher, and set:

$$P_i(t) = \mathbf{P}(T_i > t) = \mathbf{P}(X(t) < i) \tag{5.11}$$

We then have:

$$P_i(t) = \mathbf{P}(Z(1) < i, Z(2) < i, \ldots, Z(t) < i) \tag{5.12}$$

for $t = 1, 2, \ldots$ and we have immediately:

$$P_i(t) = (1 - \lambda_i)^t \tag{5.13}$$

where:

$$\lambda_i = \sum_{j \geq i} \upsilon_j = 1 - \sum_{j < i} \upsilon_j \tag{5.14}$$

Thus T_i is *geometrically distributed* with decay parameter λ_i and:

$$\mathbf{E}(T_i) = \frac{1}{\lambda_i} \tag{5.15}$$

We may also calculate the *mean best-so-far fitness* at time t to be:

$$\mathbf{E}\left(w_{X(t)}\right) = \sum_{i=1}^{N} w_i [(1 - \lambda_{i+1})^t - (1 - \lambda_i)^t] \tag{5.16}$$

5.3.3 The netcrawler on ε-correlated fitness landscapes

For a netcrawler with (possibly variable) mutation operator U, initiated with a uniform random selection at $t = 1$, let $X(t)$ again be the fittest-network-so-far in t time steps - note that since a netcrawler is *elitist*, $X(t)$ may be identified with the netcrawler process itself. Again, let T_i be the first passage time of the process $X(t)$ to i or higher so that $\mathbf{P}(T_i > t) = \mathbf{P}(X(t) < i)$. For $j < i$, let us also define $T_{i,j}$ to be the first passage time of the process to i *given that* $X(1) = j$. We then have:

$$\mathbf{P}(T_i = t + 1) = \begin{cases} \sum_{j<i} \mathbf{P}(T_{i,j} = t)\, \upsilon_j & t > 0 \\ \lambda_i & t = 0 \end{cases} \tag{5.17}$$

where again $\lambda_i = \sum_{j \geq i} \upsilon_j$. Note that the $t + 1$ appears on the LHS because we count the initial sequence $X(1)$ as a single fitness evaluation. Disregarding as $\boldsymbol{o}(\varepsilon)$ the probability that if $X(t) = i$ mutation finds a portal to a network *higher* than $i + 1$, we have:

$$T_{i,j} = T'_j + T'_{j+1} + \ldots + T'_{i-1} \tag{5.18}$$

for $j < i$, where T'_k denotes the number of fitness evaluations taken to discover a portal from Γ_k to Γ_{k+1}. Note that the T'_k in Eq. (5.18) are *mutually independent* and that to a first approximation[5]:

$$\mathbf{P}\left(T'_k > t\right) = (1 - \varepsilon_k(U))^t \tag{5.19}$$

[5]This approximation assumes that the netcrawler has at each time step an equal probability $\varepsilon_k(U)$ of finding a portal - essentially our maximum entropy assumption. In reality the probabilities of portal discovery at subsequent time steps will not be independent.

We thus have immediately:

$$\mathbf{E}(T_i) = 1 + \sum_{j<i} \mathbf{E}(T_{i,j})\upsilon_j = 1 + \frac{\upsilon_1}{\varepsilon_1(U)} + \frac{\upsilon_1+\upsilon_2}{\varepsilon_2(U)} + \ldots + \frac{\upsilon_1+\upsilon_2+\ldots+\upsilon_{i-1}}{\varepsilon_{i-1}(U)} \tag{5.20}$$

T'_k has generating function:

$$\sum_{t=1}^{\infty} \mathbf{P}\left(T'_k = t\right) z^t = \frac{\varepsilon_k(U)z}{1-[1-\varepsilon_k(U)]z} \tag{5.21}$$

so that from Eq. (5.17) the full distribution of T_i may be obtained from the generating function:

$$G_i(z) = \sum_{t=1}^{\infty} \mathbf{P}(T_i = t) z^t = z\left\{\lambda_i + \sum_{j<i} \upsilon_j \prod_{k=j}^{i-1} \frac{\varepsilon_k(U)z}{1-[1-\varepsilon_k(U)]z}\right\} \tag{5.22}$$

From:

$$\mathbf{P}(X(t) = i) = \mathbf{P}(X(t) < i+1) - \mathbf{P}(X(t) < i) = \mathbf{P}(T_{i+1} > t) - \mathbf{P}(T_i > t) \tag{5.23}$$

we derive:

$$\sum_{t=1}^{\infty} \mathbf{P}(X(t) = i) z^t = \frac{1}{1-z} \cdot \begin{cases} G_i(z) - G_{i+1}(z) & i < N \\ G_N(z) & i = N \end{cases} \tag{5.24}$$

so that we may calculate the mean best-so-far fitness at time t from the generating function:

$$\sum_{t=1}^{\infty} \mathbf{E}\left(w_{X(t)}\right) z^t = \frac{1}{1-z} \cdot \left\{w_1 G_1(z) + \sum_{i=2}^{N} (w_i - w_{i-1}) G_i(z)\right\} \tag{5.25}$$

5.4 Royal Road Fitness Landscapes

In this section we introduce the Royal Road (Mitchell et al., 1992; Forrest & Mitchell, 1993) fitness landscapes, of which, in a sense that should become clear, ε-correlated landscapes are a generalisation. Throughout this section it is convenient to label the neutral networks with indices $i, j, \ldots$ from 0 to N rather than from 1 to N, so that there are $N+1$ neutral networks. For simplicity we restrict ourselves to the binary alphabet $\mathcal{A} = \{0,1\}$, although all results carry through straightforwardly to higher cardinality alphabets.

5.4.1 Definitions and statistics

A Royal Road landscape depends on two integer parameters $N = 1,2,\ldots$ and $K = 1,2,\ldots$. A sequence then comprises N contiguous blocks of K contiguous loci - so that the sequence length is $L = NK$. We shall say that a block of K binary bits is ***set*** if all K bits have the value binary 1; otherwise we shall call it ***unset***.

Definition 5.4.1. The ***Royal Road*** fitness landscape with N blocks of length K is the fitness landscape of sequence length $L = NK$ defined by the fitness function:

$$f(x) = i \Leftrightarrow \text{exactly } i \text{ of the } N \text{ blocks of } x \text{ are set} \tag{5.26}$$

for $i = 0,1,2,\ldots,N$; i.e. to calculate fitness we simply count the number of set blocks[6].

[6] In (Nimwegen & Crutchfield, 1998) a related family of fitness landscapes, the *Royal Staircase* landscapes were introduced - here fitness is calculated by counting the number of set blocks starting from one end of a sequence, until an unset block is encountered. For the purposes of this study Royal Staircase landscapes are qualitatively similar to Royal Road landscapes, so we restrict ourselves to the latter.

Maximal neutral partitioning

The integers $i = 0, 1, 2, \ldots, N$ also label the $N+1$ (maximal) neutral networks:

$$\Gamma_i = \left\{ x \in \mathcal{A}^L \mid f(x) = i \right\} \tag{5.27}$$

We remark that for Royal Road landscapes the maximal neutral networks are not in general connected[7]. Throughout this Section the neutral partitioning is understood to be the maximal one. Relative volumes of neutral networks are easily calculated. Setting:

$$\kappa = 2^{-K} = \mathbf{P}\left(\text{a single block of length } K \text{ chosen uniformly at random is set}\right) \tag{5.28}$$

we find that:

$$\upsilon_i = \binom{N}{i} \kappa^i (1-\kappa)^{N-i} \tag{5.29}$$

Mutation probabilities and ε-correlation

Let m_{ij} be the one-point mutation probabilities. Let us set:

$$\begin{aligned} \varepsilon &= \frac{1}{2^K - 1} \\ &= \frac{\kappa}{1-\kappa} \\ &= \mathbf{P}\left(\text{a point mutation sets a uniformly randomly chosen unset block of length } K\right) \end{aligned}$$

We then have, for Royal Road landscapes:

$$m_{ij} = \frac{j}{N}\delta_{i,j-1} + \left(1 - \frac{j}{N}\right)(1-\varepsilon)\delta_{ij} + \left(1 - \frac{j}{N}\right)\varepsilon\delta_{i,j+1} \tag{5.30}$$

so that (as suggested preemptively- by the notation) ε functions as our order parameter for fitness-increasing mutation; Royal Road landscapes are indeed ε-correlated for large K (i.e. $\varepsilon \ll 1$) with:

$$\nu_j = \left(1 - \frac{j}{N}\right)(1-\varepsilon) \tag{5.31}$$

$$\varepsilon_j = \left(1 - \frac{j}{N}\right)\varepsilon \tag{5.32}$$

We see immediately that (for 1-bit mutation) *evolvability* decays *linearly* with fitness (*cf.* Section 2.3.4). We will also need to calculate (at least for small n and to $\boldsymbol{o}(1)$ in ε) the n-point mutation matrix $\boldsymbol{m}^{(n)}$. We proceed as follows: let $\boldsymbol{m}(u)$ be the mutation matrix for binomial (i.e. per-locus) mutation with per-locus rate $0 \leq u \leq 1$. We then have:

$$\boldsymbol{m}(u) = \sum_{n=0}^{L} \binom{L}{n} u^n (1-u)^{L-n} \boldsymbol{m}^{(n)} \tag{5.33}$$

Given a sequence in Γ_j - i.e. with j set and $N-j$ unset blocks. The probability that exactly k of the j set blocks undergo at least one point mutation is $\binom{j}{k}\theta^k(1-\theta)^{j-k}$ where:

$$\theta = \theta(u) = 1 - (1-u)^K = \mathbf{P}\left(\text{a single block undergoes at least one point mutation}\right) \tag{5.34}$$

[7] It may easily be seen that for Royal Staircase landscapes the maximal neutral networks *are* connected.

Similarly the probability that exactly l of the $N-j$ unset blocks undergo at least one point mutation is $\binom{N-j}{l}\theta^l(1-\theta)^{N-j-l}$. For every such block the probability that mutation *sets* it is just ε and the probability that r unset blocks become set is thus $\binom{l}{r}\varepsilon^r(1-\varepsilon)^{l-r}$. Putting this together, we find:

$$m_{ij}(u) = \sum_{k=0}^{j}\sum_{l=0}^{N-j}\sum_{r=0}^{l}\delta_{i,j-k+r}\binom{j}{k}\binom{N-j}{l}\theta^{k+l}(1-\theta)^{N-(k+l)}\binom{l}{r}\varepsilon^r(1-\varepsilon)^{l-r} \tag{5.35}$$

It is now convenient to define the generating function:

$$G_j(z,u) = \sum_{i=0}^{N} z^i m_{ij}(u) \tag{5.36}$$

and a calculation give:

$$G_j(z,u) = [z+(1-z)\theta(u)]^j\,[1-\varepsilon(1-z)\theta(u)]^{N-j} \tag{5.37}$$

The generating function:

$$G_j^{(n)}(z) = \sum_{i=0}^{N} z^i m_{ij}^{(n)} \tag{5.38}$$

for the $m_{ij}^{(n)}$ then satisfies:

$$\sum_{n=0}^{L}\binom{L}{n}u^n(1-u)^{L-n}G_j^{(n)}(z) = G_j(z,u) \tag{5.39}$$

where $G_j(z,u)$ is given by Eq. (5.37). Thus Eq. (5.37) and Eq. (5.39) may be used to calculate the Poisson and n-point mutation statistics respectively. We shall in general work to two approximations:

1. block length K is reasonably large, so that $\varepsilon \ll 1$ as required
2. the mutation rate is small, in the sense that the (mean) number of mutations per sequence is $\ll$ the number of blocks N

Under approximation 1 we thus work generally to leading order in ε. As regards approximation 2 we note in particular that the optimum mutation rates of Prop. 5.2.1 are likely to be small (roughly, of order N/j) provided j is not too small (*cf.* Fig. 5.1).

Entropy and Markov indices of neutral networks

Due to computational complexity, we calculate entropy and Markov indices analytically only for 1-point mutation (i.e. $n=1$); as n increases we might expect the former to increase and the latter to decrease. From Eq. (5.30) we have:

$$\begin{aligned} H_j &= -\sum_{i=0}^{N} m_{ij}\log_2 m_{ij} \\ &= -\frac{j}{N}\log_2\frac{j}{N} - \left(1-\frac{j}{N}\right)\log_2\left(1-\frac{j}{N}\right) - \left(1-\frac{j}{N}\right)[\varepsilon\log_2\varepsilon + (1-\varepsilon)\log_2(1-\varepsilon)] \\ &= h\left(\frac{j}{N}\right) + \left(1-\frac{j}{N}\right)h(\varepsilon) \end{aligned}$$

for $\varepsilon \ll 1$, where $h(p) = -p\log_2 p - (1-p)\log_2(1-p)$ is the entropy of a Bernoulli trial (biased coin-toss) with probability p. The first term represents the entropy of a neutral/non-neutral trial,

while the second term is a small contribution from the probability that a portal is discovered. For reasonably large K we have $h(\varepsilon) \approx 2^{-K}\left(K + \frac{1}{\log 2}\right)$. We may use the network entropies to calculate the *percolation indices* $\mathcal{P}_j = 2^{H_j}$ (Section 2.2.6) - recall that $\mathcal{P}_j$ can be interpreted intuitively as the number of networks "effectively accessible" from Γ_j. Fig. 5.2 below plots the percolation index (for 1-point mutation) against neutral network number for Royal Road landscapes with $N = 16$ and several values of K. We see that (as might be expected) neutral networks do not percolate

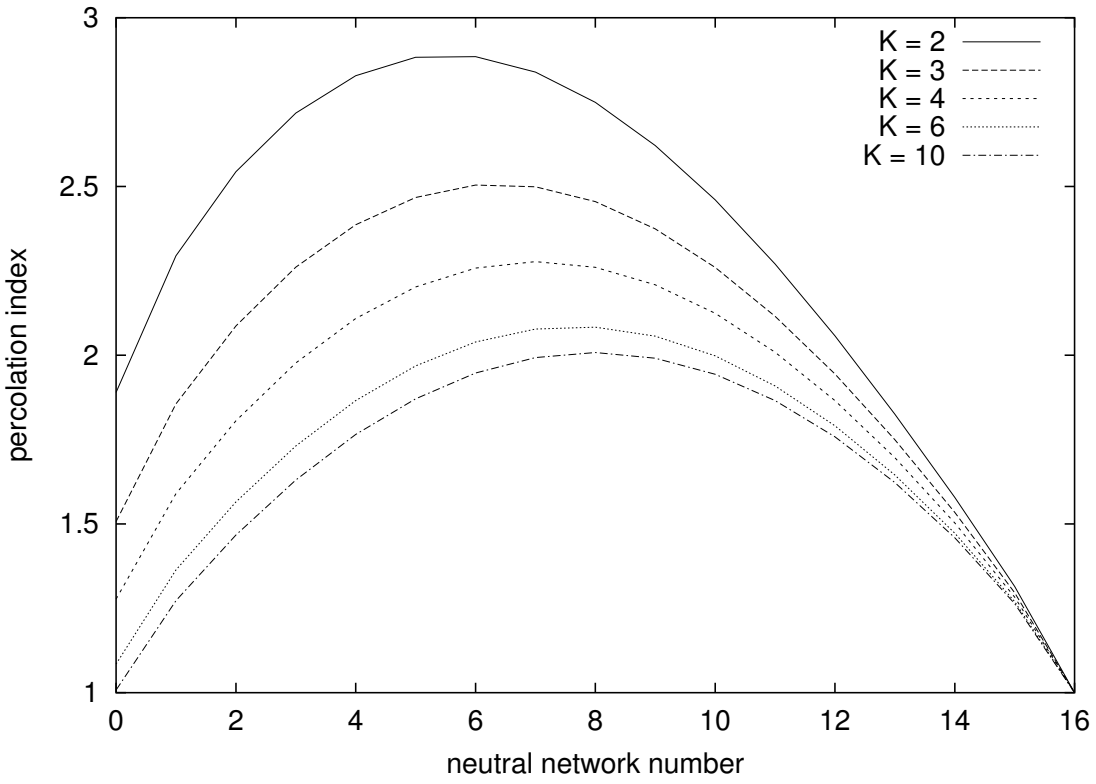

Figure 5.2: Royal Road percolation indices $\mathcal{P}_j$ for 1-point mutation, plotted for $N = 16$ and a few values of K.

much for Royal Road landscapes; indeed, for large K the only "effectively accessible" networks are the network itself and the next-lowest-fitness network, giving $\mathcal{P}_j \approx 2$.

For the Markov indices we proceed as follows: for $x \in \Gamma_j$ let $m_i(x)$ be the probability that x mutates to Γ_i under a single-point mutation and let $H(x) = -\sum_{i=0}^{N} m_i(x)\log_2 m_i(x)$. Then by Eq. (2.21) of Section 2.2.3 we have:

$$\mathcal{M}_j = H_j - \frac{1}{|\Gamma_j|}\sum_{x\in\Gamma_j} H(x) \tag{5.40}$$

Let us set:

$$\Gamma_{j,k} = \left\{x \in \Gamma_j \mid \text{exactly } k \text{ of the } N-j \text{ unset blocks have exactly } K-1 \text{ bits set}\right\} \tag{5.41}$$

for $k = 0, 1, \ldots, N-j$. Now an unset block can only become set under a single-point mutation if it already has $K-1$ bits set and an unset block with less than $K-1$ bits set remains unset under

a single-point mutation. Thus, given i, j, k, we see that $m_i(x)$ is *the same* for all $x \in \Gamma_{j,k}$, as is the entropy $H_{j,k} = H(x)$. Thus:

$$\mathcal{M}_j = H_j - \sum_{k=0}^{N-j} \frac{|\Gamma_{j,k}|}{|\Gamma_j|} H_{j,k} \tag{5.42}$$

A simple calculation gives:

$$m_i(x) = \frac{j}{N}\delta_{i,j-1} + \left(1 - \frac{j}{N} - \frac{k}{L}\right)\delta_{ij} + \frac{k}{L}\delta_{i,j+1} \tag{5.43}$$

for $x \in \Gamma_{j,k}$ and:

$$\frac{|\Gamma_{j,k}|}{|\Gamma_j|} = \binom{N-j}{k}(K\varepsilon)^k(1-K\varepsilon)^{N-j-k} \tag{5.44}$$

A calculation yields:

$$\mathcal{M}_j = \left(1 - \frac{j}{N}\right)\left\{h(\varepsilon) - \sum_{k=0}^{N-j}\binom{N-j}{k}(K\varepsilon)^k(1-K\varepsilon)^{N-j-k} \cdot h\left(\frac{k}{L}\left(1-\frac{j}{N}\right)^{-1}\right)\right\} \tag{5.45}$$

In particular, we see that:

$$\mathcal{M}_j \leq \left(1 - \frac{j}{N}\right)h(\varepsilon) \tag{5.46}$$

so that (for fixed j) $\mathcal{M}_j = \boldsymbol{O}(\varepsilon \log \varepsilon) = \boldsymbol{O}\left(K2^{-K}\right)$ as $K \to \infty$. Thus we might expect a maximum entropy approximation to work well so long as K is large enough.

We may also use the Markov indices to calculate the *percolation drift factors* $\mathcal{D}_j^{perc} = \mathcal{M}_j / H_j$ (Section 2.2.6): the higher the factor, the more important drift becomes as regards accessibility of neighbouring networks. We see that drift appears to be relatively unimportant - the percolation drift factors are ≈ 1 - but only for accessibility of other networks "in general". However, as noted in Section 2.2.6, this is somewhat misleading as regards the importance of drift for *optimisation*; specifically, in this case the percolation drift factors tell us very little about the effect of drift on accessibility of *higher fitness* networks. We thus calculate the *evolvability* drift factors $\mathcal{D}_j^{evol}$ (Eq. 2.72). Eq. (4.3) then yields:

$$\mathcal{D}_j^{evol} = 1 - h\left(\left(1-\frac{j}{N}\right)\varepsilon\right)^{-1}\sum_{k=0}^{N-j}\binom{N-j}{k}(K\varepsilon)^k(1-K\varepsilon)^{N-j-k} \cdot h\left(\frac{k}{L}\right) \tag{5.47}$$

Recall that $\mathcal{D}_j^{evol}$ represents the fraction of information about the probability of finding a portal from network Γ_j conveyed by knowledge of the actual sequence in Γ_j. Fig. 5.3 plots evolvability drift factor (for 1-point mutation) against neutral network number for Royal Road landscapes with $N = 16$ and several values of K. We see that, particularly for larger block size K, a substantial quantity of portal discovery information is conveyed by the knowledge of the actual location of a sequence in a given network - for large K (more specifically, for $N \ll 2^K/K$) we have roughly:

$$\mathcal{D}_j^{evol} \approx 1 - \frac{\log_2 L}{K} \tag{5.48}$$

In fact, for a given neutral network on a Royal Road landscape, portals are accessible (for 1-bit mutation) from only a sparse subset of the network. Neutral drift is thus likely to be essential for the discovery of portals (*cf.* Chapter 4).

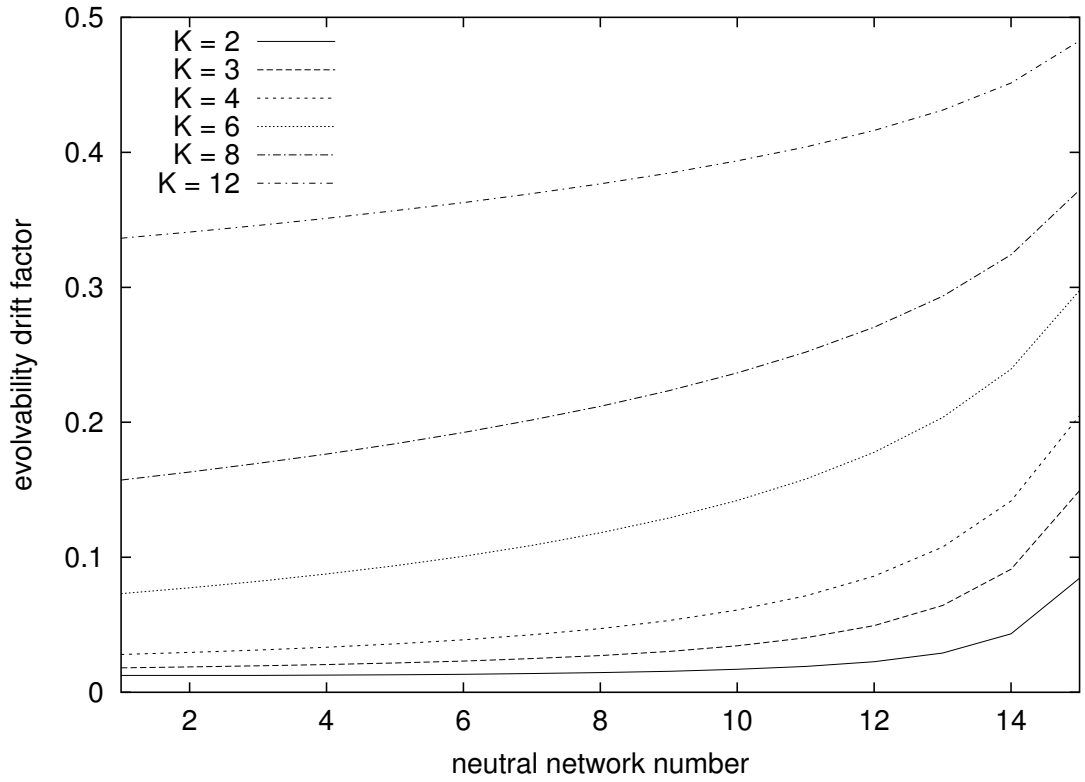

Figure 5.3: Royal Road evolvability drift factors $\mathcal{D}_j^{evol}$ for 1-point mutation, plotted for $N = 16$ and few values of K.

Correlation properties

For Royal Road landscapes the mean fitness $\mathcal{F}(d\,|\,j)$ of an d-point mutant of a sequence selected uniformly at random from Γ_j is given by $\left.\frac{dG_j^{(d)}(z)}{dz}\right|_{z=1}$ where $G_j^{(d)}(z)$ is given by Eq. (5.37) and Eq. (5.39). The local correlation properties are contained in the quantity $\mathcal{F}(d\,|\,j)$ (*cf.* Section 2.3.1). Now:

$$\left.\frac{\partial G_j(z,u)}{\partial z}\right|_{z=1} = j - [j-(N-j)\varepsilon]\theta(u) \tag{5.49}$$

Setting $w = \frac{u}{1-u}$ in Eq. (5.39) and using Eq. (5.34) we find:

$$\sum_{d=0}^{L}\binom{L}{d} w^d \mathcal{F}(d\,|\,j) = (N-j)\varepsilon(1+w)^L + [j-(N-j)\varepsilon](1+w)^{L-K} \tag{5.50}$$

which yields:

$$\mathcal{F}(d\,|\,j) = \begin{cases} (N-j)\varepsilon + [j-(N-j)\varepsilon]\left\{\binom{L-K}{d}/\binom{L}{d}\right\} & d \le L-K \\ (N-j)\varepsilon & d > L-K \end{cases} \tag{5.51}$$

We thus see that Royal Road landscapes are *linearly correlated* (Section 2.3.3). In particular, if $d \ll L$ and (as assumed) $\varepsilon \ll 1$ we have the approximation:

$$\mathcal{F}(d\,|\,j) \approx \left(1-\frac{1}{N}\right)^d j \tag{5.52}$$

so that (to a first approximation) correlation does not depend on the block length K and Royal Road landscapes are approximately *elementary* (Section 2.3.2) with auto-correlation function:

$$\rho(d) \approx \left(1-\frac{1}{N}\right)^d \tag{5.53}$$

and hence correlation length:

$$\ell = N \tag{5.54}$$

Optimal mutation mode

To test the accuracy of Prop. 5.2.1 for Royal Road landscapes, we ran a series of Monte Carlo simulations as follows: for each value of N, K and network number j and for a series of per-sequence mutation rates, we generated a sample of S uniform random sequences in Γ_j. Each sequence was mutated according to the current mutation mode and the number of sequences which found a portal to Γ_{j+1} totalled. 95% confidence limits were calculated according to Student's T-test (Feller, 1966). For each set of parameters the experiment was performed first for n-point (constant) and then per-locus (Poisson) mutation. Results were plotted against the analytic approximations Eq. (5.4) and Eq. (5.5) for portal discovery probabilities for n-point and Poisson mutation respectively - see Fig. 5.4 and Fig. 5.5.

We see that for the larger block size $K = 8$ (Fig. 5.5) and particularly for smaller mutation rates, the analytic results are in good agreement with experiment; less so for the smaller block size $K = 4$ (Fig. 5.4) and larger mutation rates. This is to be expected as, firstly Eq. (5.4) and Eq. (5.5) are only to $\boldsymbol{o}(\varepsilon) = \boldsymbol{o}\left(2^{-K}\right)$ and secondly the analysis relies on the (strong) multiplicative

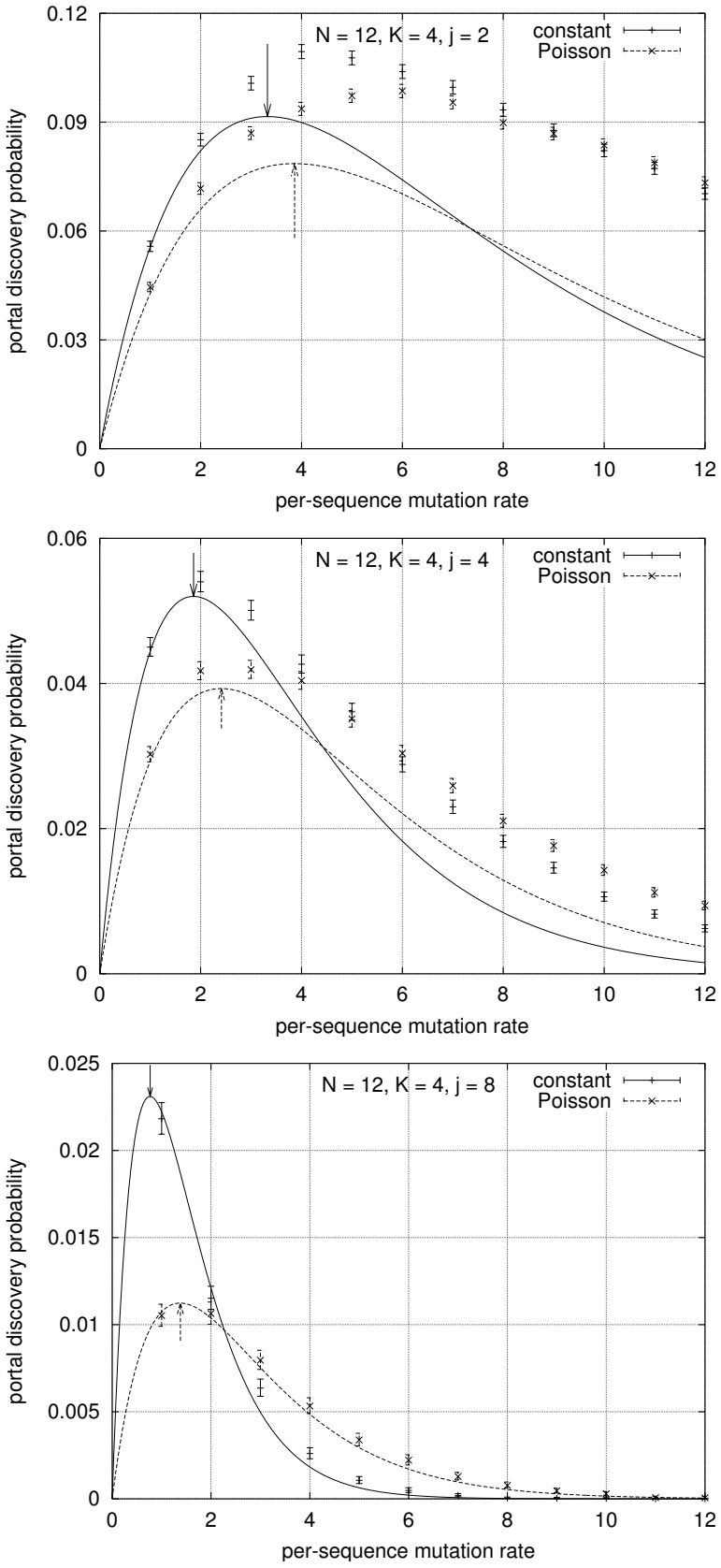

Figure 5.4: Portal discovery probability plotted against per-sequence mutation rate for a Royal Road landscape with parameters $N = 12,\ K = 4$ and several values of $j =$ network number, for n-point (constant) and per-locus (Poisson) mutation modes. Solid lines give the analytic values from Eq. (5.4) and Eq. (5.5); the vertical arrows indicate the optimum mutation rates of Prop. 5.2.1. Points are values from a simulation with a sample size of $S = 100,000$. Error bars give 95% confidence limits.

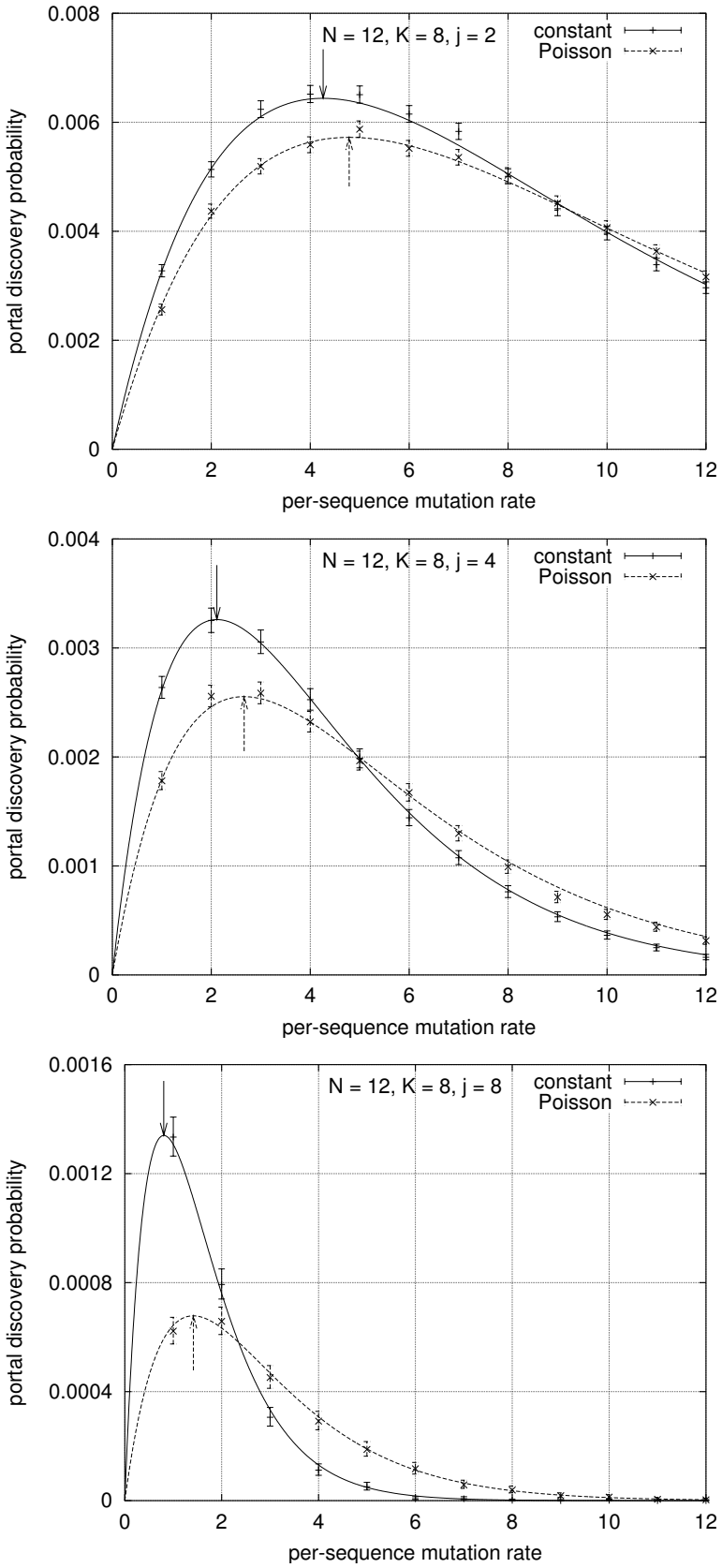

Figure 5.5: Portal discovery probability plotted against per-sequence mutation rate for a Royal Road landscape with parameters $N = 12,\ K = 8$ and several values of $j =$ network number, for n-point (constant) and per-locus (Poisson) mutation modes. Solid lines give the analytic values from Eq. (5.4) and Eq. (5.5); the vertical arrows indicate the optimum mutation rates of Prop. 5.2.1. Points are values from a simulation with a sample size of $S = 1000,000$. Error bars give 95% confidence limits.

mutation approximation (Section 2.2.4) which, as we have seen, is likely to apply if the Markov index Eq. (5.45) is small. For the larger block size at least then, our results support Prop. 5.2.1 for Royal Road landscapes, insofar as:

- Optimum mutation rates are correctly predicted
- The optimum portal discovery probability is higher for constant than for Poisson mutation

We remark that there is a fairly significant difference between the optimum portal discovery probabilities for constant and Poisson mutation (the optimum *rate* for the latter is also generally higher). As suggested in Chapter 2 (Section 2.3.4), there may be sound reasons, Prop. 2.3.1 and Prop. 5.2.1 notwithstanding, to use Poisson mutation rather than a constant mutation mode on occasion.

The adaptive netcrawler on Royal Road landscapes

To test the adaptive netcrawler suggested in Section 5.3.1 on Royal Road landscapes, we tested two schemes. They are both based, as suggested in Section 5.3.1, on a working assumption that the next-highest neutral network is of *equal neutrality* to the current network. The first scheme is based on the observation that under the above assumption the *observed* neutrality should approach $1/e$ at the optimum mutation rate. Thus an estimate ν_{obs} of observed neutrality (i.e. proportion of neutral mutations) is accumulated over the previous t_{lag} fitness evaluations. At the end of each t_{lag} evaluations, ν_{obs} is compared with $1/e$. If it is smaller, the current mutation rate is incremented; if larger the current mutation rate is decremented. This scheme, however, proved to be somewhat unstable, in that the mutation rate tended to fluctuate wildly. We thus settled on the following alternative scheme: a sliding estimate ν_{obs} of observed neutrality (i.e. proportion of neutral mutations) is maintained over a "window" of the previous t_{lag} fitness evaluations. After each fitness evaluation the current mutation rate n is updated according to:

$$n \leftarrow \left[-\frac{n_{ave}}{\log \nu_{obs}} \right] \tag{5.55}$$

where n_{ave} is the average (arithmetic mean) of the actual mutation rate used over the previous t_{lag} fitness evaluations[8]. The idea is that n tracks the optimum mutation rate, based on a sliding estimate of actual neutrality, for a neutral network where the next-highest network is of equal neutrality. We note that there are essentially two parameters involved: the "window" size t_{lag} and the initial mutation rate. It proved judicious also to specify a *maximum* mutation rate (which could usually be taken to be the same as the initial mutation rate). In practice both parameters would be tuned for best performance[9]. This latter scheme proved more stable and successful in tracking the (known) optimum mutation rate. Fig. 5.6 illustrates a typical run. As might be expected (since for Royal Road landscapes the next-highest network has somewhat smaller neutrality than the current), there is a tendency to overestimate the optimum mutation rate - this may be seen clearly

[8] We have glossed over a subtlety here: the mutation rate is likely to be changing over the "window". If the mutation rates over the window are $n_1, n_2, \ldots, n_{t_{lag}}$ and t_{neut} of the t_{lag} evaluations are neutral, then the best estimate for the *actual* neutrality ν is given by: $\nu^{n_1} + \nu^{n_2} + \ldots + \nu^{n_{t_{lag}}} = t_{neut}$. This equation is not (efficiently) solvable for ν. However, since n will not generally change too rapidly, using instead the arithmetic mean n_{ave} proved sufficient and efficient to implement.

[9] In the case of Royal Road landscapes we in fact know that the (optimum) initial mutation rate, assuming a random start lands on the lowest network, is simply the block size N, so that in actual experiments we might allow ourselves to cheat a little...

for the first few epochs in Fig. 5.6. For general (unknown) landscapes for which one suspected some relationship between fitness and neutrality (see e.g. the next Chapter), one might attempt to correct this effect by the addition of a (possibly fitness-dependent) parameter to reduce slightly the estimate of optimum mutation rate.

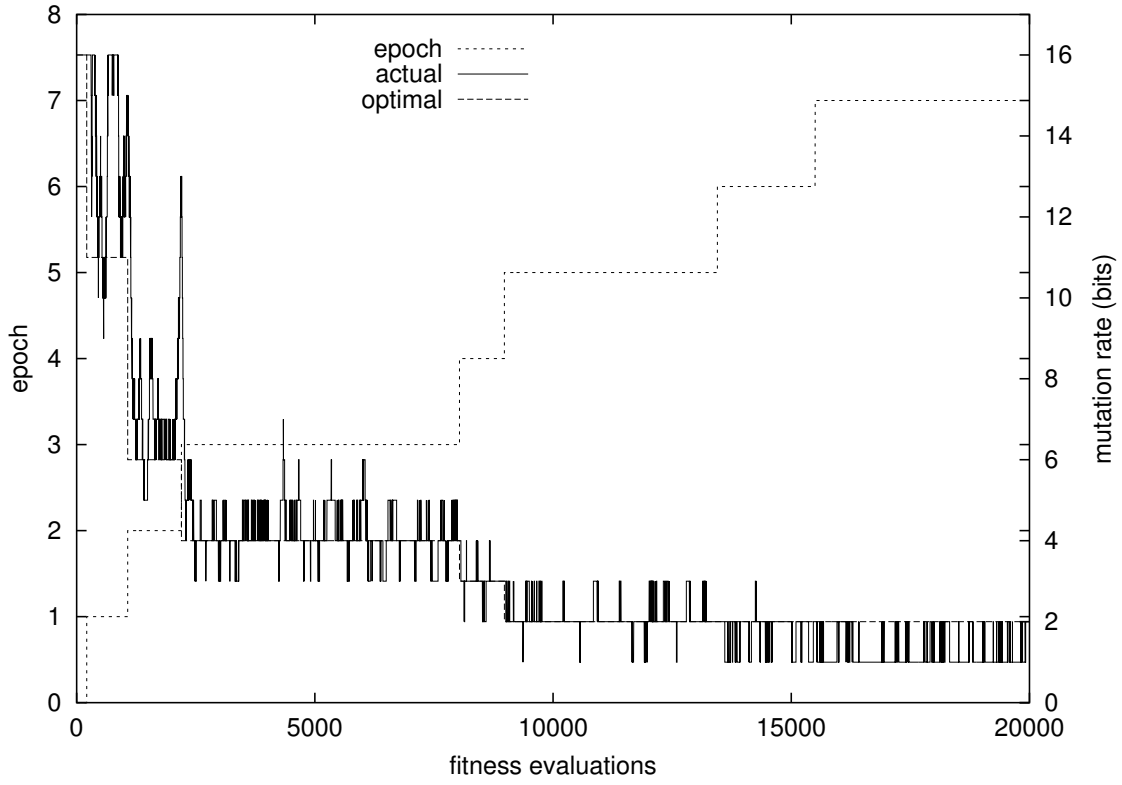

Figure 5.6: A typical run of an adaptive netcrawler (Section 5.3.1) on a Royal Road fitness landscape. The horizontal axis measures fitness evaluations. The current epoch of the netcrawler (i.e. current neutral network) is plotted against the left-hand vertical axis. Actual and optimal mutation rates (in bits) are plotted against the right-hand vertical axis. Parameters are: $N = 16$, $K = 12$, "window" size $= 100$ fitness evaluations, initial/maximum mutation rate $= 16\ (= N)$.

With the above proviso, we note that neither scheme depends on any knowledge of the underlying fitness landscape; an adaptive netcrawler may be run on an arbitrary fitness landscape, even if not known to be formally ε-correlated (but which may, in some sense, behave "locally" as though it were). We return to this issue in the next Chapter.

5.4.2 Search performance on Royal Road landscapes

The principal contention of this Chapter is contained in Conjecture 5.3.1 - that a netcrawler with the optimal (constant) mutation mode is the most effective evolutionary search process on an ε-correlated landscape. Here we put Conjecture 5.3.1 to the test on some Royal Road landscapes, where we pit a netcrawler against some commonly encountered GA's.

Performance comparison

Firstly we establish a basis for comparison of search performance on Royal Road landscapes (*cf.* Section 3.5). For a full time/fitness-critical comparison we should have to compare the full distribution of T_i, the first passage time to network i or higher, for each network i (or equivalently of $X(t)$, the best-so-far network at time t, for all t within a reasonable range), for every set of parameter values. This is unfeasible on grounds of time and space; we limit ourselves, then, to the fitness-critical measures $\mathbf{E}(T_i)$ and the time-critical measures $\mathbf{E}(X(t))$, remarking that simulations (not illustrated) measuring the full distributions of T_i and $X(t)$ do not change any of our conclusions.

We are still, however, faced with the inevitability that our simulations can only run for a finite, albeit large, maximum number of fitness evaluations, t^*, say. This is in particular a problem when it comes to estimating $\mathbf{E}(T_i)$ for the various search processes we shall be examining, since in any given instantiation of a search process and for a given i, our process may not have attained network i within the maximum time t^*. Simply ignoring runs which do not reach i will not do; apart from reducing the sample size we would also thus bias our estimate of $\mathbf{E}(T_i)$. To address this problem we make the following observation, borne out through many simulations and indeed for all search processes tested: given a network i, for some suitably large t^*, the distribution of T_i *given that* $T_i > t^*$ is similar to a *geometric* distribution; i.e. for $t > t^*$ we have $\mathbf{P}(T_i > t) \approx (1-\lambda_i)^t$ for some decay factor λ_i.

Consider, then, an experiment where we wish to estimate by sampling the mean $\mathbf{E}(T)$ of a random variable T representing the first passage time of a process $X(t)$ to a given state i. We assume that there is some large t^* and a decay factor λ such that for $t > t^*$ we have $\mathbf{P}(T > t) \approx (1-\lambda)^t$. Suppose, now, that we run S instantiations of our process $X(t)$, where S is assumed large, terminating each run at $t = t^*$. Suppose that of the S runs, some S' of them reach state i within time t^*. We may then estimate:

$$p = \mathbf{P}(T > t^*) \approx 1 - \frac{S'}{S} \approx (1-\lambda)^{t^*} \tag{5.56}$$

We then have:

$$\begin{aligned}
\mathbf{P}(T=t) &= \mathbf{P}(T=t \mid T \le t^*)\mathbf{P}(T \le t^*) + \mathbf{P}(T=t \mid T > t^*)\mathbf{P}(T > t^*) && (5.57)\\
&= \mathbf{P}(T=t \mid T \le t^*)\mathbf{P}(T \le t^*) + \mathbf{P}(T > t^* \mid T = t)\mathbf{P}(T=t) && (5.58)\\
&\approx \mathbf{P}(T=t \mid T \le t^*)\mathbf{P}(T \le t^*) + \mathbf{P}(t > t^*)\lambda(1-\lambda)^{t-1} && (5.59)
\end{aligned}$$

We thus find:

$$\mathbf{E}(T) \approx \mathbf{E}(T \mid T \le t^*)\mathbf{P}(T \le t^*) + p\left(t^* + \frac{1}{\lambda}\right) \tag{5.60}$$

Now suppose that of the S' samples which achieve state i within time t^*, the first passage times to i are $t_1, t_2, \ldots, t_{S'}$. We then have:

$$\mathbf{E}(T \mid T \le t^*) \approx \frac{1}{S'}(t_1 + t_2 + \ldots + t_{S'}) \tag{5.61}$$

which yields:

$$\mathbf{E}(T) \approx \frac{1}{S}(t_1 + t_2 + \ldots + t_{S'}) + p\left(t^* + \frac{1}{\lambda}\right) \tag{5.62}$$

A similar argument yields

$$\mathbf{E}\left(T^2\right) \approx \frac{1}{S}\left(t_1^2 + t_2^2 + \ldots + t_{S'}^2\right) + p\left[\left(t^* + \frac{1}{\lambda}\right)^2 + \frac{1}{\lambda^2} - \frac{1}{\lambda}\right] \tag{5.63}$$

where we now *define*:

$$p = 1 - \frac{S'}{S} \tag{5.64}$$

and:

$$\lambda = 1 - p^{1/t^*} \tag{5.65}$$

We may then estimate $\mathbf{E}(T)$ and $var(T)$ from Eq. (5.62) and Eq. (5.63); the results will be reasonably accurate provided the sample size S is large, and p is small. In practice we found that good estimates were obtained provided $p \leq 0.1$. If *no* processes in the sample failed to reach the state i it was found to be acceptable to take $p = 0$ provided the sample size was large enough. In all cases we used this technique to estimate means and standard deviations for the T_i. If the resulting p was found to be > 0.1 we rejected the result and, if possible, repeated the experiment with a larger value for t^*.

The netcrawler on Royal Road landscapes

To test the analytic results for the netcrawler on Royal Road landscapes, we ran simulations of netcrawlers (with optimal constant mutation) for several ranges of parameters and checked the results against the predictions of Section 5.3.3. For each trial a sample of 1000 netcrawlers (each initialised to a random starting sequence) were run to a maximum of $t^* = 1,000,000$ fitness evaluations. Results were generally found to be in reasonably good agreement with theory for the distribution of the first passage times T_i (Eq. (5.22)), the mean first passage times $\mathbf{E}(T_i)$ (Eq. (5.20)) and the mean fitness $\mathbf{E}\left(w_{X(t)}\right)$ (Eq. (5.25)). Fig. 5.7 illustrates simulation results for $\mathbf{E}(T_i)$ (points with error bars) plotted against the analytic approximation of Eq. (5.20) (solid lines) for a range of N and K values. Means and standard deviations for T_i were calculated as described in the previous sub-section; the value of t^* used ensured $p \approx \mathbf{P}(T_i > t^*)$ was < 0.1 in all cases. We remark again that the analysis of Section 5.3.3 is based on our maximum entropy approximation. For Royal Road landscapes a Markov analysis (Kampen, 1992) may be carried through for the precise distribution of the first passage times T_k' from epoch k to $k+1$. The results yield values for $\mathbf{E}\left(T_k'\right)$ slightly higher than our estimate of $1/\varepsilon_k(U)$, but approaching it asymptotically for large block size K (this is in line with our analysis of the Markov indices for Royal Road networks). Nevertheless, we see from Fig. 5.7 that we are still able to predict $\mathbf{E}(T_i)$ quite accurately, even for K as small as 4.

Comparison with GA's

Given the huge proliferation of GA's in the literature we choose a suite of GA's which we hope may be seen as representative of at least some of the major aspects of more "conventional" GA's.

As has been remarked earlier, a common perception of the functioning of GA's places the onus of search on *recombination*, whereas in the current work we have explicitly rejected recombination for reasons to be discussed more fully in a later Chapter. We thus divide our test suite into GA's with and without recombination. For GA's *without* recombination mutation is the (unique) search operator. To level the playing field, we thus deploy the same optimum (i.e. constant)

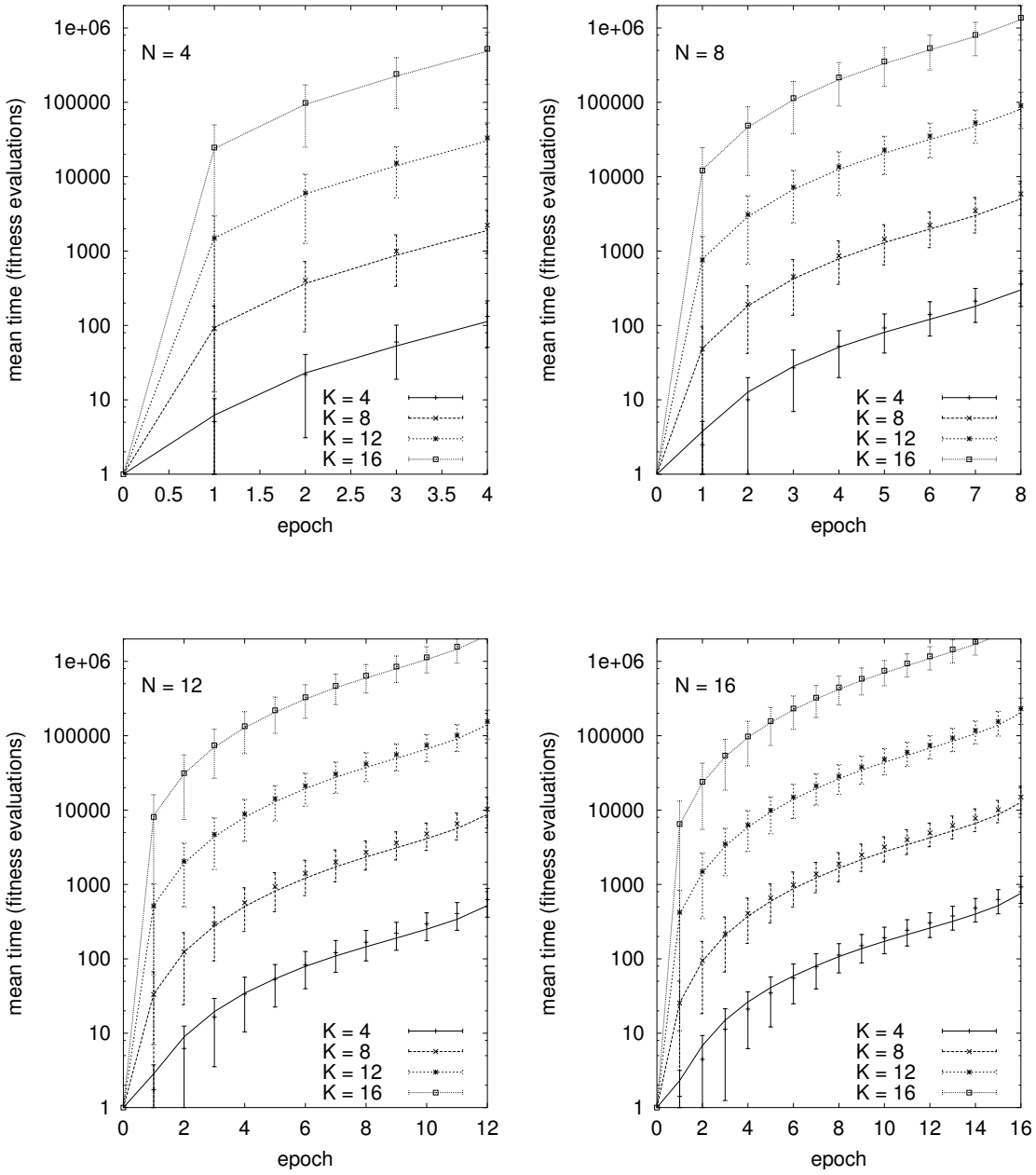

Figure 5.7: Sample estimates of expected first passage times T_i for an (optimal) netcrawler on Royal Road landscapes for a range of N and K values. Vertical axes measure times in fitness evaluations, on a logarithmic scale. The horizontal axes specify the epoch (i.e. the network number i). Means (points) and standard deviations (error bars) were estimated using Eq. (5.62) and Eq. (5.63); in all cases sample size was 1000 and the netcrawlers were run for $1,000,000$ fitness evaluations (which proved sufficient to ensure that $p < 0.1$). Solid lines plot the theoretical estimate Eq. (5.20) for $\mathbf{E}(T_i)$

mutation mode and rate for all such GA's as for the netcrawler. For GA's *with* recombination we adopt the conventional view of recombination as the principal search operator, with mutation as a "background" operator to maintain genetic diversity; we thus deploy a more conventional (per-locus) mutation operator. Since it might be deemed unfair to match such a GA against an optimal netcrawler (where mutation rate/mode depends on knowledge of the landscape) we compare performance in this case to an *adaptive* netcrawler, as described in the previous sub-section.

In either scenario, where there are "tunable" parameters for our GA's (including the adaptive netcrawler) we take care to attempt, as far as possible, to tune parameters for optimal performance on the given landscape and for the given performance criterion.

Key features of GA's that we attempt to cover are:

- fitness-proportional vs. fitness rank selection
- multinomial sampling vs. stochastic universal sampling
- generational vs. "steady-state"
- one-point vs. uniform recombination
- enforced elitism
- fitness scaling

Of course we cannot be exhaustive; some commonly encountered schemes which we explicitly exclude (on the grounds that they would lead us too far afield) include:

- distributed GA's
- associative (and other) mating schemes
- nicheing/fitness sharing

The final list of population-based GA's to be matched against a netcrawler was as follows:

FP	Fitness-proportional selection
RANK	Fitness rank selection
2TWRL	2-Tournament winner-replaces-loser

The "generational" GA's FP and RANK were run with either multinomial (roulette-wheel) sampling (MNS) or stochastic universal (roulette-wheel) sampling (SUS) and also both with and without enforced elitism. FP and RANK were also run with either linear (LIN), power law (POW) or exponential (EXP) scaling[10] - for FP, the actual fitness is scaled, while for RANK the fitness *rank* is scaled (see below). With recombination, 1-point, 2-point and *uniform* crossover were trialled.

We firstly describe the generational GA's FP and RANK. Scaling was performed as follows: a *scale factor* parameter $s > 0$ controls the selection pressure. For linear scaling (LIN) the scaled "roulette-wheel sector size" is given by:

$$x_{scaled} = x + s \tag{5.66}$$

[10] We did not implement the so-called "sigma-scaling" (Mitchell et al., 1992; Goldberg, 1989) often encountered in the literature.

For power law scaling (POW) the scaled size is given by:

$$x_{scaled} = (x+1)^s \tag{5.67}$$

an for exponential scaling (EXP) the scaled size is given by:

$$x_{scaled} = e^{sx} \tag{5.68}$$

where x represents the fitness ($x = 0,1,2,\ldots,N$) for fitness-proportional selection (FP) and the fitness rank number in the population, minus one ($x = 0,1,2,\ldots,M-1$) for rank selection (RANK). Note that for linear scaling increasing s *reduces* selection pressure, while for power law and exponential scaling increasing s *increases* selection pressure. If we take as a measure of selection pressure the ratio $\frac{x_{scaled}}{(x-1)_{scaled}}$ of a sector to the next-smallest (possible) sector, we find, for a sector of size x, selection pressures of $1+\frac{1}{x+s-1}$, $\left(1+\frac{1}{x}\right)^s$ and e^s for LIN, POW and EXP respectively. Note that for LIN and POW, for a given scale factor s, selection pressure decreases (approaching 1 from above) with increasing sector size x, while for EXP it remains constant.

Both FP and RANK selection utilise a simulated a roulette-wheel. For FP each population member is allocated a sector of size equal to its scaled fitness. For RANK the population is sorted[11] into (ascending) rank order by fitness and each population member is allocated a sector of size equal to its scaled rank. For multinomial sampling (MNS) selections are performed so that for any selection event the *probability* of selecting a given population member is proportional to its sector size (thus so too is its *expected* number of selections). For stochastic universal sampling (SUS) (Baker, 1987) selections are performed such that, given a fixed number of selections (generally equal to the population size), the *expected* number of selections of a population member is proportional to its sector size and the *variance* in its selection probability is minimal.

In the limit of very high scale factors (for power law and exponential scaling), the interplay with selection is as follows: for FP selection the entire roulette-wheel is allocated equally among the *highest fitness* sequences in the population. Thus MNS sampling amounts to uniform random selections from the highest fitness sequences, while SUS sampling yields (approximately) equal numbers of selections for *each* of the highest fitness sequences. For RANK selection, *one* of the highest fitness sequences is chosen uniformly at random (*cf.* the previous footnote regarding shuffling of the population prior to sorting); the entire roulette-wheel is then allocated to this single sequence, which consequently receives *all* samples, for both MNS and SUS sampling. Frequently in seeking optimal GA settings, it became clear that performance increased toward the maximum scale factor limit. Due to floating-point storage limitations, if a scale factor was > 30 (POW) or > 10 (EXP) the limiting case was assumed (*cf.* Table 5.1).

With no recombination, the generational GA's operate as follows. Population size is M. For the initial generation M sequences are created at random. For each subsequent generation M selections are performed according to the operant selection/sampling method. Each selected sequence is mutated (according to the theoretical optimal mode/rate for the selected sequence; i.e. according to its neutral network) and added to a new population. After M such selections the new population replaces the old to become the current population.

[11] Implementation note: before sorting the population is *shuffled*. This avoids the possibility that the sorting algorithm - *qsort* in our case - always rank *equally fit* sequences in the same order - which would, of course, violate the Markov property of the evolutionary process!

With recombination there is an additional *recombination rate* parameter $0 \leq r \leq 1$ and a *mutation rate* parameter $\bar{u}$ equal to the expected number of mutations per sequence, although (for reasons given above) mutation is now on a *per-locus basis*[12]. Again an initial random population of size M is created. For each generation we again perform M selections. This time, after each selection, with probability r a recombination event occurs and an additional "parent" sequence is selected. This parent selection is somewhat different for MNS and SUS sampling: in the former case another roulette-wheel selection is performed, while in the latter case a uniform random selection is made from the current population[13]. The selected parents are crossed over according to the crossover mode (1-point, 2-point or uniform), mutated[14] and added to the new population as before.

For the "steady-state" tournament GA 2TWRL (Chapter 3, Example 3.2.2) again an initial population of size M is created at random. The following procedure is then iterated: two sequences are selected uniformly at random and without replacement from the population for a "tournament". The tournament winner is declared to be the fitter (or, if the selections are of equal fitness, the first) of the two selections. If there is no recombination a copy of the winner is then mutated and the mutant copy replaces the tournament loser. If there is recombination then with probability r the winner crosses over with the loser and the offspring mutates and again replaces the loser. Note that 2TWRL is "almost" *elitist* insofar as, if there is just one maximally fit sequence in the population, then the only way that sequence might be eliminated at the expense of a less fit sequence is if it is selected twice for a tournament (so it becomes both winner and loser...) and then mutates (and/or mates) to produce an inferior offspring which replaces itself! We note that if selections for the tournament were performed *without* replacement then the process would be strictly elitist.

In the case of recombination, it was found that 2-point and (more markedly so) uniform crossover were invariably inferior in performance compared to the corresponding process with 1-point crossover. This is reasonable, if we consider that uniform recombination (and to a lesser extent 2-point crossover) is more likely to destroy already set blocks. We therefore present results for 1-point crossover only.

As regards elitism, we found that enforcing any form of "pure" elitism - i.e. ensuring that we never lose all of the current fittest sequences - merely reinforced the main thesis of this Chapter rather strongly, in that performance generally favoured smaller population sizes (and smaller recombination rates). In particular, best results were always obtained with a population size of 1 and a recombination rate of 0 - in which case the corresponding GA simply became a netcrawler! Results involving explicitly imposed elitism are thus omitted.

Simulations were performed for a Royal Road landscapes with $N = 8$ and $K = 8$ as follows: for each GA the process was simulated $1,000$ times (with different random seeds[15]) from random initial populations up to $10,000$ fitness evaluations. For each time (number of fitness evaluations) the

[12] Although we mutate on a per-locus basis, we still prefer - contrary to common practice in the GA literature - to quote mutation rates as an (expected) per-sequence rate as, in the author's view, this is a more meaningful figure. The reader irritated by this practice may feel free to avail them self of a pocket calculator...

[13] The reason for this is that SUS sampling presupposes a *fixed* number of selections per generation. Other schemes are of course feasible, but for simplicity we choose random "other parent" selection.

[14] We note that some GA practitioners (Michaelewicz, 1996) prefer to separate recombination and mutation; i.e. a sequence *either* mates *or* mutates. The practical difference proved minimal in our case.

[15] For pseudo-random number generation we used the "Mersenne Twister" (Matsumoto & Nishimura, 1998) RNG.

sample mean best-so-far fitness over the $1,000$ runs was calculated[16]. Standard deviations in best-so-far fitness (not shown) were also calculated (the relative deviation = standard deviation/actual fitness tended to range between $5\% - 15\%$). Great care was taken to tune the various parameters (population size, scale factors, recombination rates and mutation rates) for best performance over the alloted $10,000$ evaluations, where the mean best-so-far fitness at $10,000$ evaluations was taken as the performance indicator. Results are displayed in Fig. 5.8 and Fig. 5.9 with optimised parameters as in Table 5.1 and Table 5.2 respectively. For comparison, (theoretical) random search results and (simulated) optimised/adaptive netcrawler results are shown alongside.

We remark that for reasons of time and space we have presented results only for a Royal Road landscape with $N = 8$ and $K = 8$. This proved quite representative - other Royal Road landscape parameters did not spring any surprises, provided the block size K was reasonably large.

selection method	sampling method	scaling method	*scale factor*	*pop. size*	*performance indicator*
ONC	–	–	–	1	7.911
RSH	–	–	–	1	2.018
FP	MNS	LIN	0.00	100	4.632
FP	MNS	POW	*max*	45	7.673
FP	MNS	EXP	*max*	35	7.713
FP	SUS	LIN	0.00	100	4.809
FP	SUS	POW	*max*	50	7.694
FP	SUS	EXP	*max*	50	7.756
RANK	MNS	LIN	0.00	80	3.025
RANK	MNS	POW	15.00	55	7.339
RANK	MNS	EXP	0.16	80	7.207
RANK	SUS	LIN	0.00	80	3.133
RANK	SUS	POW	16.00	50	7.420
RANK	SUS	EXP	0.19	70	7.355
2TWRL	–	–	–	20	7.052

Table 5.1: Optimised GA parameters and results (no recombination), on a Royal Road landscape with $N = 8$, $K = 8$. ONC = optimal netcrawler, RSH = random search - see text for other abbreviations. The final column *performance indicator* = mean best-so-far fitness at $10,000$ fitness evaluations. For scale factors marked "*max*" see text.

We give a brief discussion of the results, beyond pointing out that they generally support our contention well that the (optimised/adaptive) netcrawler yields optimum performance on Royal Road (and, we would claim, ε-correlated) landscapes. We remark in particular that it is clear from the results that, while some GA's came close in performance under the time-critical "mean best-so-far fitness at t^* evaluations" indicator for $t^* = 10,000$, those GA's had to be finely tuned to achieve good performance *for that particular time scale*; the netcrawler frequently outperformed the *same*

[16]Note that even for the generational GA's we count fitness evaluations strictly sequentially, in the sense that at each actual fitness evaluation - i.e. after a mutation, a recombination or (in the initialisation phase) a random creation - we check if we have discovered a new best-so-far fitness.

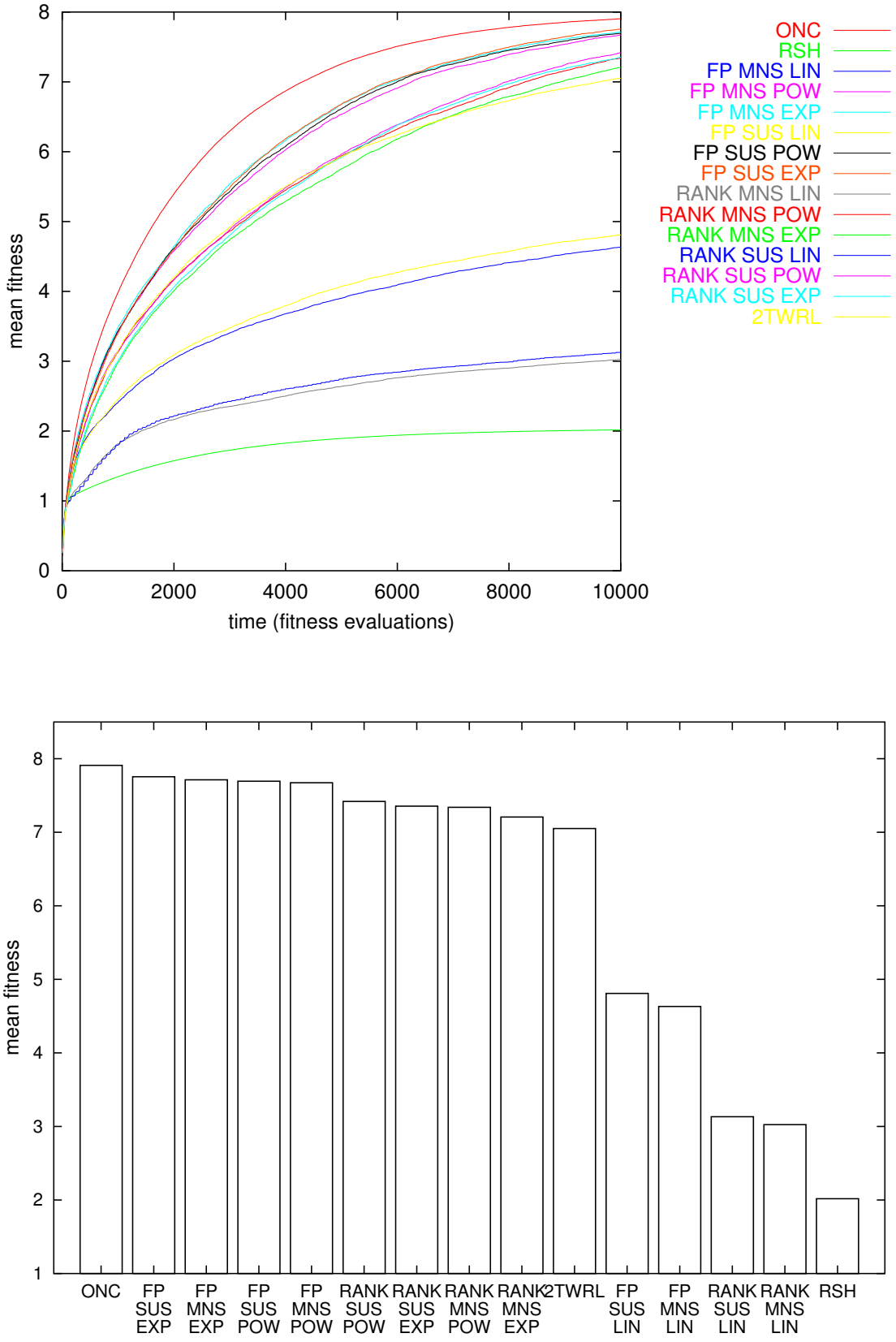

Figure 5.8: Optimised GA performance (no recombination) on a Royal Road landscape with $N = 8$, $K = 8$: mean best-so-far fitness (sample size $1,000$ runs) plotted against time in fitness evaluations. See text and Table 5.1 for key and parameters. The bottom figure shows a histogram of mean best-so-far fitness at the end of each run, ranked by performance.

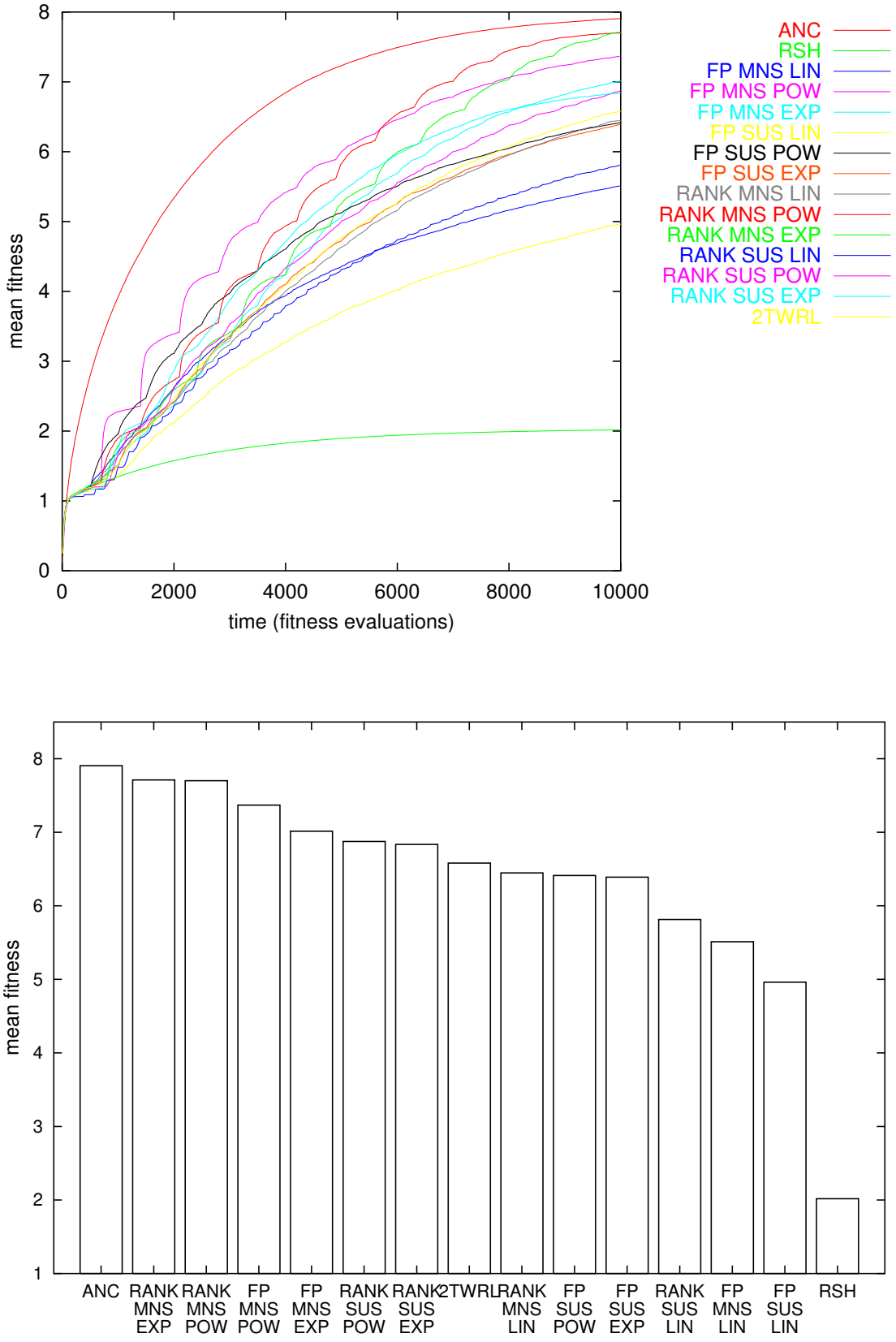

Figure 5.9: Optimised GA performance (with recombination) on a Royal Road landscape with $N = 8$, $K = 8$: mean best-so-far fitness (sample size $1,000$ runs) plotted against time in fitness evaluations. See text and Table 5.2 for key and parameters. The bottom figure shows a histogram of mean best-so-far fitness at the end of each run, ranked by performance.

selection method	sampling method	scaling method	*scale factor*	*(per-seq.) mut. rate*	*(1-point) rec. rate*	*pop. size*	*performance indicator*
ANC	–	–	–	*adaptive*	–	1	7.906
RSH	–	–	–	–	–	1	2.018
FP	MNS	LIN	0.00	0.6	1.0	250	5.511
FP	MNS	POW	6.00	1.0	1.0	700	7.370
FP	MNS	EXP	1.00	1.2	1.0	500	7.014
FP	SUS	LIN	0.00	0.3	0.5	250	4.963
FP	SUS	POW	8.00	1.0	0.9	500	6.412
FP	SUS	EXP	2.00	0.7	1.0	500	6.391
RANK	MNS	LIN	0.00	0.6	1.0	300	6.447
RANK	MNS	POW	8.00	1.0	1.0	700	7.702
RANK	MNS	EXP	0.01	0.9	1.0	800	7.712
RANK	SUS	LIN	0.00	0.5	0.4	200	5.814
RANK	SUS	POW	12.00	1.0	0.9	500	6.875
RANK	SUS	EXP	0.04	0.6	0.9	500	6.836
2TWRL	–	–	–	1.7	0.6	400	6.582

Table 5.2: Optimised GA parameters and results (with recombination), on a Royal Road landscape with $N = 8$, $K = 8$. ANC = adaptive netcrawler, RSH = random search - see text for other abbreviations. The final column *performance indicator* = mean best-so-far-fitness at $10,000$ fitness evaluations. Parameters for the adaptive netcrawler were: "window" size = 100 fitness evaluations, initial/maximum mutation rate = 8 (= number of blocks).

GA by orders of magnitude over either smaller or larger time scales. Furthermore the netcrawler still outperformed any GA finely tuned to *any particular* time scale t^* tested[17]. Similar results (not shown) were found to hold for the fitness-critical performance indicator "mean first passage time to fitness w^*" for all (non-trivial) fitness levels tested and indeed we contend that the netcrawler is optimal in the strong sense that for any t, w we have $\mathbf{P}(T_{ANC}(w) \leq t) \geq \mathbf{P}(T_{GA}(w) \leq t)$ or equivalently $\mathbf{P}(W_{ANC}(t) \geq w) \geq \mathbf{P}(W_{GA}(t) \geq w)$ for the adaptive netcrawler (ANC) for any GA (without knowledge of the landscape beyond that it is ε-correlated), where $T(w) =$ first passage time to fitness w or greater and $W(t) =$ best-so-far fitness at time t.

One possibly surprising feature of our results is the efficacy of (often quite drastic) scaling in improving GA performance. Firstly we note that for linear scaling a scale factor of $s = 0$ is always preferable - not surprisingly considering that $s = 0$ furnishes maximum selection pressure in this case. It is also clear that either power law or exponential scaling is generally useful. The simple explanation for this is probably that severe scaling tends to force the GA to *exploit* any fitness gain - and this is, as we have argued, precisely what we want for optimising an ε-correlated landscape (at least by mutation).

Another initially puzzling feature is the "stepiness" evident in the mean best-so-far fitness graphs for some of the GA's with severe scaling and large population sizes. An explanation is as follows: consider the case of severe scaling (i.e. large s), fitness-proportional selection with

[17] Not quite... see Fig. 5.10 below and discussion regarding recombination.

multinomial sampling and a large population size M. After the random initial population stage (generation 0) suppose that the highest network found is j_0 and that, say, k members in the initial population are on the j_0 network. During the next generation (generation 1), those k sequences dominate the roulette-wheel to the extent that every selection is more or less a uniform sample of those k sequences. Suppose that there is no recombination. Each of the k selectees mutates with a certain probability of finding a portal to network j_0+1. There is a reasonable chance, if the population is large, that (at least) one such mutant does so - but there is only a tiny chance (by the nature of ε-correlation) that any mutant finds a portal to a network *higher* than j_0+1. As a result, we can expect the best-so-far fitness to change (quite early on in the generation for small j_0) from j_0 to j_0+1 - *and then to stay there*. This theme is then repeated for the next generation (generation 2) at j_0+1, and so on. The "stepiness" decreases with increasing fitness, since for higher j the probability of finding a portal to $j+1$ becomes smaller and so portals tend to be discovered *later* during a generation (or not at all, so that the averaged out "steps" begin to overlap). We would expect (and indeed find) a similar effect with other sampling methods (SUS) and with fitness ranking (in this case only the top-ranked sequence (on j_0, of course) is repeatedly selected, but the effect is similar. Something analogous also appears to occur with recombination, even at small mutation rates (see below). The prerequisite is that it be "easy" to find (by whatever means) a portal to the next step up but "difficult" to find a portal to any *higher* step. These conditions appear to exist too for recombination. We would not expect (and indeed do not find) the effect in our steady-state GA.

Although not the point of this Chapter, we do have some interest in the utility of recombination in our GA's. By and large they support well the conclusions of (Forrest & Mitchell, 1993), a landmark study of the efficacy of recombination and the *Building Block Hypothesis* (Chapter 7, Section 7.1) on Royal Road landscapes. They found similarly that their GA's were invariably outperformed by a "Random Mutation Hill-Climber" (RMHC) - basically our netcrawler with constant 1-bit mutation. Their analysis of the reasons for the GA's poor performance - in particularly the apparent failure of recombination to splice together sequences with different set blocks as per the Building Block Hypothesis - fingers *hitch-hiking* (Chapter 7, Section 7.2) as the principal culprit. This is the phenomenon, long known to population genetics, whereby whenever a genotype in a population discovers a "good gene", that genotype (and its identical-but-for-mutation progeny), rapidly take over the population with all its "bad genes" in tow, thus leaving insufficient genetic variation for recombination to work with. In effect, there rarely *are* sequences with *different* good genes simultaneously in a population and recombination has nothing to splice together. Thus we may well ask what role recombination is fulfilling in our GA's. Firstly, we note that the optimised parameters generally have a *high* recombination rate (frequently near 1 and occasionally, amusingly, near the folkloric "magic" rate of 0.6 (Michaelewicz, 1996). This implies that recombination is at least not a hindrance to our GA; quite the opposite, in fact.

To test the efficacy of (1-point) recombination we ran several of our GA's with a recombination rate of 1 and a mutation rate of 0; this is, perhaps, a "Big Bang" GA (Chapter 3, Section 3.4) in its purest form - optimisation will (hopefully) continue until there is insufficient genetic variation (which may be lost through stochastic sampling or may not have been present in the initial random population) for recombination to be effective. The situation appears complex, but it appears that

with careful tuning of selection pressure ("exploitation", but danger of hitch-hiking and running out of diversity too quickly) and population size ("exploration", but danger of wastefulness) quite good results may be achieved - but parameters, as pointed out above, have to be finely tuned to meet specific time/fitness criteria. Fig. 5.10 demonstrates the "exploit/explore" (Holland, 1992) trade-offs involved. In summary, while it is clear that (1-point) recombination *can* put together "building blocks" very effectively on a Royal Road landscape[18] - indeed, this was the *raison d'etre* for their design - it seems difficult to exploit this capability in a GA (Mitchell et al., 1994) to the extent that it can compete with an (optimised/adaptive) netcrawler.

We note finally that the adaptive netcrawler achieves a performance almost as good as the optimal netcrawler and that tuning was quick and simple - there seems to be a large "sweet spot" in the parameter values (primarily the "window" size), which are also, happily, rather insensitive to time/fitness scales. In contrast, tuning the parameters of the population GA's was (as any GA practitioner will verify...) a tedious and time-consuming task - while parameters sometimes exhibit large "sweet spots" they very frequently turn out to be sensitive, difficult to predict and interact "synergistically" (i.e. performance appears to depend on highly non-linear combinations of parameters), as well as depending critically on the time/fitness scale for which one wishes to optimise performance. Population size in particular seemed often to be quite tightly related to the time scale, particularly with severe scaling and recombination (see above for a partial explanation). We note with interest that an analytical study on the related "Royal Staircase" landscapes by (Nimwegen & Crutchfield, 1998) reaches a seemingly opposite conclusion, noting a large "sweet spot" in population size and mutation rates for a simple fitness-proportional GA (without recombination). However the performance measure in that study is the (fitness-critical) first passage time to achieve maximum fitness as opposed to our (time-critical) best-so-far fitness measure, so - quite apart from the landscape differences - we shouldn't expect comparable qualitative behaviour.

5.5 Discussion

In this Chapter we introduced the statistical property of ε-correlation to describe landscapes with neutral networks for which higher networks are accessible only from the current network. For such landscapes we have calculated (Prop. 5.2.1) the optimal mutation mode/rate and argued (Conjecture 5.3.1) that there is also an optimal evolutionary search strategy which is not population-based but rather a form of hill-climber which we have dubbed the *netcrawler*. On the basis of these results we have proposed a heuristic - the $1/e$ *Neutral Mutation Rule* (Prop. 5.3.1) - which we claim to have more general application on fitness landscapes with neutral networks. We have also proposed an *adaptive* variant of the netcrawler which gathers statistical information about the landscape as it proceeds and uses this information to self-optimise.

We remark that a major motivation for the research presented in this study was a series of experiments by Thompson and Layzell in on-chip electronic circuit design by evolutionary methods (Thompson & Layzell, 2000), during which an algorithm almost identical to our netcrawler was used with some success. The mutation rate deployed in these experiments, albeit chosen on heuristic grounds different from ours, in fact turns out to be almost precisely the optimal rate predicted

[18] So well, in fact, that in Fig. 5.10 (top figure, pop. size $= 100$) we see that for (very) short time scales/low fitness recombination actually slightly outperforms the adaptive netcrawler...

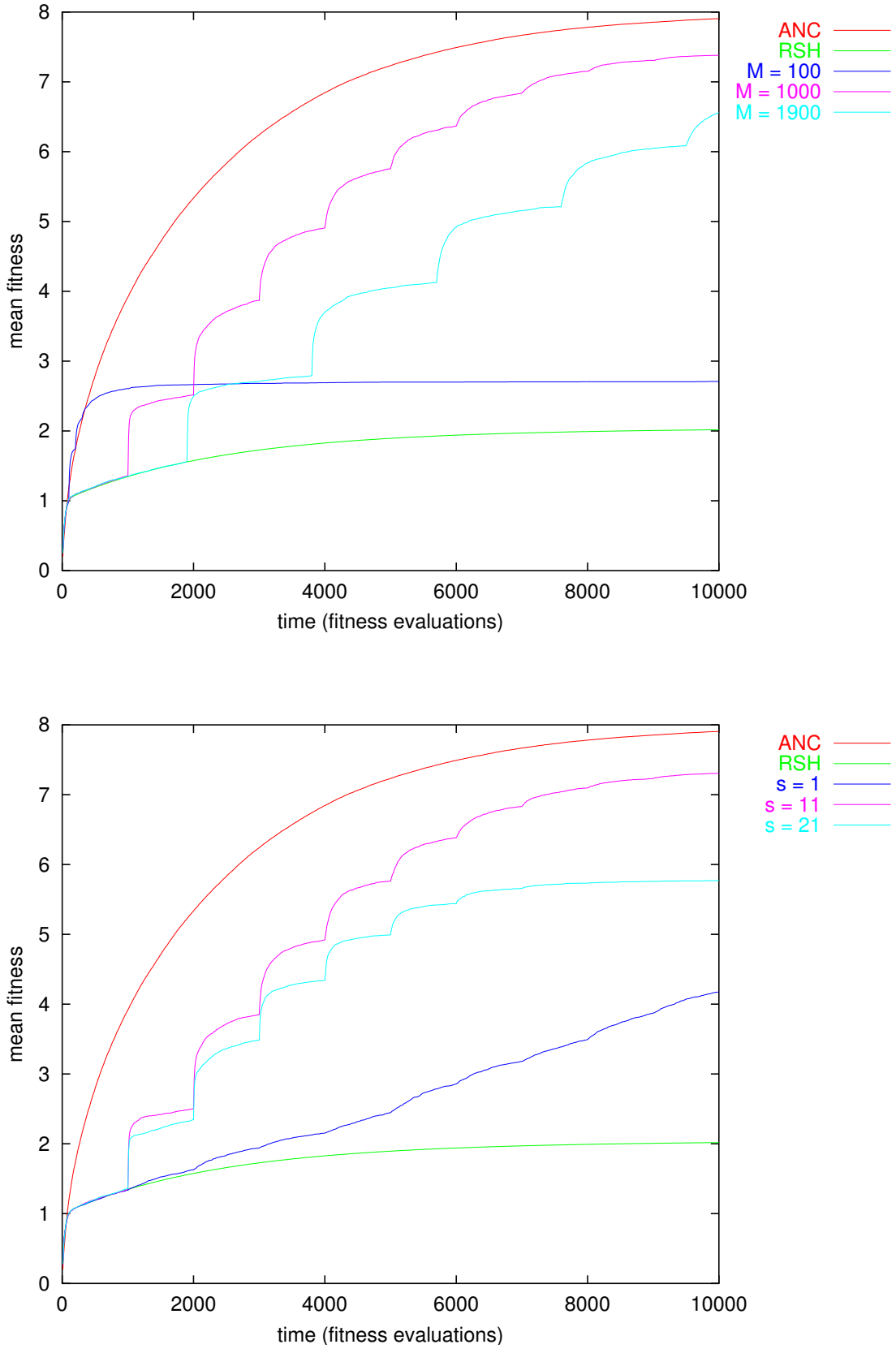

Figure 5.10: *Recombination only*: performance of a fitness-proportional GA with multinomial sampling and power law scaling on a Royal Road landscape with $N = 8$, $K = 8$: mean best-so-far fitness (sample size $1,000$ runs) plotted against time in fitness evaluations. In the top figure selection pressure is high (scale factor $= 10$) and population size is varied. In the bottom figure population size is high ($= 1,000$) and selection pressure is varied.

here, given the (estimated) neutrality inherent in the problem. There is, we note, no particular evidence that the fitness landscape in these experiments is ε-correlated to any degree; indeed, work in progress by the author suggests that the principal results presented in this Chapter may obtain under considerably less stringent assumptions (*cf.* Chapter 6) than ε-correlation.

Chapter 6

The NKp Family of Random Fitness Landscapes

6.1 Background

The NK family of random fitness landscapes were introduced by Stuart Kauffman (Kauffman & Levin, 1987; Kauffman, 1989; Kauffman, 1993) as a statistical model to investigate the phenomenon of *epistasis* - where the effect on fitness of substituting an allele at some locus depends on the particular alleles at other loci on the genotype. While this phenomenon had long been recognised by population geneticists (S. Wright, 1932) it tended either to be absent or vastly oversimplified in their models. Kauffman's model, it might be said, is more in line with random energy models from statistical mechanics, (eg. *spin glasses* (Sherrington & Kirkpatrick, 1975; Anderson, 1985)) or from combinatorial optimisation (eg. Travelling Salesman problem (Lawler, Lenstra, Kan, & Shmoys, 1985), graph bi-partitioning (Fu & Anderson, 1986), *etc.*) than standard population genetics models. The gist of Kauffman's construction is to abstract away the details of how genotype maps to fitness - that mapping is deemed unknown and inscrutable and is therefore (in the spirit of statistical mechanics) modelled as a random mapping - except that this random mapping assumes a degree of epistasis in that fitness depends (additively) on contributions from overlapping groups of "epistatically linked" loci. Crucially, the degree of epistasis in the NK model can be "tuned" by means of the K parameter, making NK landscapes a candidate test bed for investigating the effects of epistasis.

The emphasis in Kauffman's analysis was on the *"ruggedness"* or *correlation* properties of NK landscapes arising from epistatic interactions of loci and on how this ruggedness mediates the behaviour of evolutionary processes. The NK model was extended in (Barnett, 1997; Barnett, 1998) to the NKp family of landscapes, to incorporate the phenomenon of *selective neutrality* (Section 1.2). The key feature of the NKp model turns out to be the "statistical independence", in a sense to be made precise below, of epistasis (as tuned by the K parameter) and neutrality (as tuned by the p parameter), making NKp landscapes a candidate test bed for investigating the effects and interaction of epistasis and neutrality. NKp landscapes feature *neutral networks* although (probably) not, it should be remarked, structurally similar to those of the RNA folding landscapes which inspired the concept - indeed, (arguably) more realistic *random graph* statistical models (Reidys, 1995; Reidys et al., 1997) have been developed to investigate the neutral network struc-

ture of RNA folding landscapes. NKp landscapes, thus, do not set out to model these landscapes but might (we shall argue) merit consideration as models for fitness landscapes arising in *artificial evolution*/optimisation. They might, at least, serve as an aid to intuition for the structure and evolutionary dynamics on fitness landscapes featuring both epistasis and selective neutrality.

NK landscapes (and NKp landscapes) have been placed in the more general context of *random additive landscapes* by (Reidys & Stadler, 2001) in which the statistical properties of epistasis and neutrality may be analysed. In this Chapter we do generalise somewhat the original NK (and NKp) constructions but concentrate more on properties specific to these landscapes. We remark that we approach the concept of "random fitness landscape" in the "constructive" (rather than the more mathematically rigorous "prescriptive") fashion outlined in Chapter 2.

We note that the basic NKp landscape construction was originally introduced in (Barnett, 1997), where the statistical independence of epistasis and neutrality was also conjectured (to be proved rigorously by (Reidys & Stadler, 2001)). A few further statistical results on NKp landscapes were presented in (Barnett, 1998); nonetheless, the majority of the analysis in the this Chapter represents new research.

A note on notation and terminology

As regards notation, we forsake compatibility with the literature in the interests of internal consistency; in particular, we retain the notation w, w', *etc.* for fitness values (rather than x, y, ...) and W, W', ... (rather than X, Y, ...) for random variables representing fitness values. We shall, furthermore, continue to write L for sequence length, rather than N as implied by the notation "NK". We shall also, for convenience, use the term *"arbitrary"* in a specific technical sense, to mean *"drawn from a uniform random distribution"* from some set (which will generally be clear from the context). In particular, throughout this Chapter the random variable W will denote *the fitness of an arbitrary sequence (i.e. one drawn uniformly at random) from a sequence space* $\mathcal{A}^L$.

6.1.1 Construction

The Generalised NKp Model

In Kauffman's original scheme for NK landscapes (Kauffman, 1993) each locus on a sequence of length L contributes additively to the fitness of that sequence. The contribution of a locus, drawn (independently and at random) from some underlying distribution, then depends on the allele at the locus itself and the alleles at some other K (randomly chosen) loci. While in a biological context there may be some justification for considering fitness contributions on a per-locus basis, this seems less obvious if we intend to use the NK scheme to model an artificial evolution fitness landscape. Rather, we introduce a variation on the NK theme as follows: suppose that the fitness of a potential candidate for the solution of an optimisation problem depends, via some genotype $\rightarrow$ phenotype mapping, on some set of F ***features*** of the phenotype. For example, in attempting to evolve a neural network controller for a robot that is required to perform a specified task, we might consider the neural network design to be the "phenotype", while the "features" might be higher level robot "behaviours" associated with the phenotype (e.g. *move towards light, avoid collision with object*, etc.), on which fitness is ultimately evaluated. Now the sequence $\rightarrow$ feature mapping may be assumed complex and inscrutable - otherwise we would probably not bother applying artificial evolution to our problem! We thus, in the spirit of the standard NK model, assume (i)

that our F features contribute additively and independently towards fitness; (ii) that the ***fitness contribution*** of a feature depends on the alleles at some subset of loci of a sequence - we shall say that a locus ***influences*** a feature if the fitness contribution of that feature depends on the choice of allele at that locus - and (iii) for each combination of alleles at the influencing loci the fitness contribution of a feature is drawn independently from some ***underlying distribution*** (i.e. real-valued random variable) Z. As yet we make no assumptions about the assignation of influencing loci nor about the underlying distribution.

Epistasis in the NK Model

We now make the assumption that the loci influencing a given feature are chosen independently *per feature*[1] and define the ***epistasis parameter*** κ to be the probability that an (arbitrary) locus influence an (arbitrary) feature. We do allow some flexibility in the choice of loci that influence a feature; in particular, we shall consider the ***fixed epistasis*** model, where each feature is influenced by exactly K loci, chosen uniformly at random from the L possible loci (so that $\kappa = K/L$) and the ***variable epistasis*** model where, for each feature, the probability that it be influenced by a locus is decided (independently for each locus) by a biased coin-toss with probability κ. There are, of course, other possible choices for assigning epistasis (e.g. the "nearest neighbour" scheme in (Kauffman, 1993)); as we shall see, however, the significant quantity[2] will turn out to be simply the *number* of loci that influence an arbitrary feature. We note that the fixed epistasis model with $F = L$ corresponds[3] to the standard NK model with "fully random" epistasis (Kauffman, 1993; Altenberg, 1995).

The Generalised NKp Model

We shall also extend our generalised NK model to include (a generalisation of) the NKp landscapes (Barnett, 1997). The motivation for the NKp construction is that it seems reasonable to suppose that for many (if not most) combinations of alleles at the loci influencing a feature, that feature will be "ineffective" in the sense that it will make a *zero contribution* to overall fitness. Specifically, in the spirit of the NK model, we suppose that a feature make a zero contribution - independently for any combination of alleles at the influencing loci - with fixed probability $0 \leq p < 1$ so that the underlying fitness distribution Z takes the form:

$$Z = \begin{cases} 0 & \text{with probability } \; p \\ Y & \text{with probability } \; q \equiv 1 - p \end{cases} \tag{6.1}$$

where Y is a continuous, non-atomic[4] (Feller, 1966) real-valued random variable. In other words, for each feature and for each particular combination of alleles at the influencing loci, whether that feature makes a zero contribution is decided on a biased coin-toss controlled by the ***neutrality parameter*** $0 \leq p \leq 1$. Neutrality is (as we shall see) zero for $p = 0$ and increases with increasing p. The case $p = 0$ yields (almost surely) a trivial landscape where every sequence has fitness 0.

NKp landscapes are thus specialisations of generalised NK landscapes. When discussing NKp

[1] One might, alternatively, choose the features influenced by a given locus independently *per locus*. Our choice is based on analytic tractability and simplicity of (computer) implementation.

[2] At least in the absence of any consideration of *recombination*.

[3] Almost... in the usual NK construction a locus always "influences itself".

[4] I.e. $\mathbf{P}(Y = w) = 0$ for any $w \in \mathbf{R}$.

landscapes we will, by abuse of language, refer to Y (rather than Z) as the "underlying distribution".

The Underlying Distribution

In contrast to the standard NK (resp. NKp) model - where the underlying distribution is usually (but not always) taken to be uniform on $[0,1]$ - we shall frequently specialise to the case where the underlying fitness distribution Z (resp. Y) is *Gaussian with mean* 0. This case might be interpreted in a similar spirit to quantitative genetics: a feature may contribute either advantageously or detrimentally (or, in the NKp case, not at all) to fitness. Most changes to a sequence will cause comparatively small changes to the contribution of a feature while a few changes will have a more drastic (advantageous or detrimental) effect on a feature. Since we have no particular reason to expect changes in features to be biased towards the advantageous or the detrimental a normal distribution seems a reasonable choice.

Of course we cannot in this case interpret fitness in the biological sense (*cf.* the discussion in the introduction to Chapter 2) since fitness may be negative. This is not a problem if we are (as is frequently the case) concerned more with fitness *rank* than fitness itself; if we wish to interpret fitness in a more biological manner (e.g. to use fitness-proportional selection) the Gaussian model could either be made multiplicative by considering $e^{k \cdot fitness}$, where the parameter k controls selection pressure, or alternatively we might consider negative fitness as "lethal" (or perhaps, for artificial evolution, as a "constraint violation") and truncate fitness values below zero.

Another justification for the Gaussian choice is an appeal to the Central Limit Theorem: if the fitness of a feature is in reality itself due to an (additive, independent and identically distributed) combination of a fairly large number of contributions from an unknown underlying distribution, the contribution of the feature will be approximately Gaussian. It must also be conceded that the pleasant additive properties of Gaussian distributions allow us to proceed further with the analysis, particularly as regards *fitness-dependent* statistics.

The Fitness Function

The fitness function for generalised NK landscapes is described as follows: suppose that feature n is influenced by the l_n influencing loci $(\alpha_{n,1}, \alpha_{n,2}, \ldots, \alpha_{n,l_n})$; l_n and the $\alpha_{n,i}$ are to be considered as (jointly distributed) random variables, corresponding to the random assignment of influencing loci to features. Given a sequence $x \in \mathcal{A}^L$ we define $x_n = \left(x_{\alpha_{n,1}}, x_{\alpha_{n,1}}, \ldots, x_{\alpha_{n,2_n}}\right) \in \mathcal{A}^{l_n}$ to be the sequence of alleles at the influencing loci for feature n. Now we consider any element $\xi \in \mathcal{A}^{l_n}$ to be "an index into the ***fitness table***" for feature n - that is, any such index references a fitness contribution $Z_n(\xi)$ where the $Z_n(\xi)$ are (real-valued, jointly distributed) random variables mutually iid as Z, the underlying fitness distribution. The fitness of a sequence $x \in \mathcal{A}^L$ is then given by:

$$f(w) = \frac{1}{F}\left(f_1(w) + f_2(w) + \cdots + f_F(w)\right) \tag{6.2}$$

where $f_n(w) = Z_n(x_n)$ is the fitness contribution of the n-th feature.

We remark that the Royal Road landscapes of the previous Chapter may be considered as special *instances* of a class of NK (or indeed NKp) landscapes, where each of the N blocks of length K represents a feature influenced by every locus in that block, and fitness table entries are zero except for the entry indexed by the block comprising all 1's, which has an entry of $1/N$.

Throughout the remainder of this Chapter we drop the "generalised": by "NK landscape" we mean a generalised NK landscape as described above. By "NKp landscape" we mean a generalised NK landscape where, as described above, the underlying fitness distribution Z takes the form in Eq. (6.1) with neutrality parameter p and (continuous, non-atomic) underlying distribution Y. Finally, we restrict our analysis to *binary* NK landscapes. Most results generalise straightforwardly to higher order alphabets.

6.2 Statistical Analysis - Global Structure

Throughout this section $\mathcal{L}$ is a random family (in the sense of Section 2.4) of NK landscapes as described above, with F features, sequence length L, epistasis K and underlying fitness distribution Z. In the following sub-section Z may be an arbitrary real distribution[5]; thereafter we specialise to the NKp case with neutrality parameter p and underlying distribution Y.

6.2.1 NK landscapes - Correlation

Let the real-valued r.v. W be the fitness of an arbitrary (i.e. uniform random) sequence on $\mathcal{A}^L$, as in Eq. (2.75) of Chapter 2. We then have:

$$W = \frac{1}{F}(Z_1 + \cdots + Z_F) \tag{6.3}$$

where the Z's are iid as Z. This gives immediately:

$$\mathbf{E}(W) = \mathbf{E}(Z) \tag{6.4}$$

$$var(W) = \frac{1}{F} var(Z) \tag{6.5}$$

More generally, if $M(t)$ is the mgf of Z then the mgf of W is just $M(t/F)^F$.

Now let $W^{(d)}$ be the fitness of the *same* sequence with d (uniform) randomly selected loci mutated (i.e. flipped), where $0 \leq d \leq L$; that is, $W^{(d)}$ corresponds to the W' of Eq. (2.76), Chapter 2 for the constant uniform mutation operator $U^{(d)}$. We now examine the joint distribution of $W, W^{(d)}$ - how, in other words, the fitnesses of "nearby" sequences compare. Suppose that altogether n features are influenced by at least one of the d flipped loci for $n = 0, 1, \ldots, F$. Since the loci which influence a given feature are chosen independently per feature, the probability that there are exactly n such features is given by:

$$P_n(d) = \binom{F}{n} (1 - \rho(d))^n \rho(d)^{F-n} \tag{6.6}$$

where:

$$\rho(d) = \mathbf{P}(\textit{an arbitrary feature is not influenced by any of } d \textit{ arbitrary loci}) \tag{6.7}$$

The choice of notation will become clear below. For the *fixed epistasis* model we find:

$$\rho(d) = \begin{cases} \binom{L-K}{d} \Big/ \binom{L}{d} & d < L - K \\ 0 & d \geq L - K \end{cases} \tag{6.8}$$

[5] Technically, we should demand that Z possess a well-defined *moment generating function*, or at least first and second moments.

while for the *variable epistasis* model:

$$\rho(d) = (1-\kappa)^d \tag{6.9}$$

We may then write:

$$W = \frac{1}{F}(Z_1 + \cdots + Z_F) \tag{6.10}$$

$$W^{(d)} = \frac{1}{F}\left(Z'_1 + \cdots + Z'_n + Z_{n+1} + \cdots + Z_F\right) \tag{6.11}$$

where all the Z's are iid as Z. We can rewrite this as:

$$W = U_n + V_{F-n} \tag{6.12}$$

$$W^{(d)} = U'_n + V_{F-n} \tag{6.13}$$

where:

$$U_n = \frac{1}{F}(Z_1 + \cdots + Z_n) \tag{6.14}$$

$$U'_n = \frac{1}{F}\left(Z'_1 + \cdots + Z'_n\right) \tag{6.15}$$

$$V_{F-n} = \frac{1}{F}(Z_{n+1} + \cdots + Z_F) \tag{6.16}$$

The U_n, U'_n and V_{F-n} are (mutually) independent; essentially, V_{F-n} represents the *correlated* and U_n, U'_n the *uncorrelated* fitness contributions of the original sequence and its mutant. Noting that the mgf of $\frac{1}{F}(Z_1 + \cdots + Z_n)$ is given by $M(t/F)^n$ where $M(t)$ is as before the mgf of Z, we find immediately that the joint mgf $M^{(d)}(s,t)$ of $\left(W, W^{(d)}\right)$ is given by:

$$M^{(d)}(s,t) = \sum_{n=0}^{F} P_n(d) M(s/F)^n M(t/F)^n M((s+t)/F)^{F-n} \tag{6.17}$$

$$= \{(1-\rho(d))M(s/F)M(t/F) + \rho(d)M((s+t)/F)\}^F \tag{6.18}$$

From this expression we may calculate the (ensemble) auto-correlation for our landscape family. By Eq. (2.78) this is just $corr\left(W, W^{(d)}\right) = cov\left(W, W^{(d)}\right)/var(W)$, recalling that since the mutation operator $U^{(d)}$ is uniform the (marginal) distributions of W and $W^{(d)}$ are the same. The variance term has been given above. The covariance may be calculated from the joint mgf as:

$$cov\left(W, W^{(d)}\right) = \mathbf{E}\left(WW^{(d)}\right) - \mathbf{E}(W)\,\mathbf{E}\left(W^{(d)}\right) \tag{6.19}$$

$$= \left.\frac{\partial^2 M^{(d)}(s,t)}{\partial s \partial t}\right|_{s=t=0} - \mathbf{E}(Z)^2 \tag{6.20}$$

$$= \frac{1}{F}\rho(d)var(Z) \quad \text{from Eq. (6.18), after some algebra} \tag{6.21}$$

$$= \rho(d)var(W) \tag{6.22}$$

We have proved the basic result (and justified our choice of notation):

Propostion 6.2.1. *The (ensemble) auto-correlation function for a family of (generalised) NK landscapes is the* $\rho(d)$ *of Eq. (6.7).* ∎

Thus for NK landscapes auto-correlation is *independent of the number of features F and of the underlying distribution Z*. In particular, we note that *auto-correlation is independent of neutrality*, since any neutrality in our model must depend on a particular form for the fitness distribution Z. This was first conjectured for NKp landscapes in (Barnett, 1997) and proved rigorously for a wide class of *additive random landscapes* (of which NKp landscapes are an example) in (Reidys & Stadler, 2001).

We see from Eq. (6.9) that for the variable epistasis model our generalised NK landscapes are *elementary* (Section 2.3.2) with correlation length:

$$\ell = 1/\kappa \tag{6.23}$$

That is to say, auto-correlation decays exponentially with Hamming distance and with decay factor κ. For the fixed epistasis model, from Eq. (6.8) we have[6]:

$$\rho(d) \approx \left(1 - \frac{K}{L}\right)^d = (1-\kappa)^d \tag{6.24}$$

for $d \ll L-K$. Thus for the small $\kappa = K/L$ the fixed epistasis model is approximately elementary with correlation length:

$$\ell = L/K \tag{6.25}$$

6.2.2 NKp Landscapes - Contributing Features

We now specialise to the NKp case with neutrality parameter $0 \leq p < 1$ and underlying fitness distribution Y so that Z is given by Eq. (6.1). In analysing NKp landscapes, the ***Central Property*** (regarding mutation) is the following:

> A mutation at a locus is neutral iff every feature influenced by that locus makes a zero fitness contribution for *both* alleles at that locus[7]

To see this, suppose that some feature influenced by the locus in question makes a non-zero contribution for one or both alleles at that locus. Then flipping the allele will necessarily reference a *different* fitness contribution for that feature. This alternative fitness contribution will either be zero (with probability p) or a different non-zero value. In either case, by atomicity of Y, the alternative contribution will be (with probability 1) different from the original fitness contribution. The same will thus be true (again by atomicity of Y) for the fitness of the entire sequence.

NKp landscapes are by no means "homogeneous". In particular, the structure local to a sequence x depends crucially on the number of non-zero fitness contributions to the fitness of x. We introduce the terminology that for $x \in \mathcal{A}^L$, feature n is a ***contributing feature*** (for x) iff $f_n(x) \neq 0$, where $f_n(x)$ is the fitness contribution of the n-th feature for sequence x. We may thus partition $\mathcal{A}^L$ into subsets distinguished by number of contributing features:

$$\mathcal{C}_c = \left\{ x \in \mathcal{A}^L \mid f_n(x) \neq 0 \text{ for exactly } c \text{ values of } n \right\} \tag{6.26}$$

[6] Note that if $d \geq L-K$ then *every* feature must be influenced by flipping d loci, so that W, $W^{(d)}$ are completely uncorrelated.

[7] Strictly speaking, this is true *almost surely* - i.e. with probability 1. We shall not in general specify explicitly whenever a result obtains almost surely.

for $c = 0, 1, \ldots, F$. Let the r.v. C be the number of contributing features of a sequence with fitness W picked (uniformly) at random from $\mathcal{A}^L$, so that W and C are jointly distributed. Let $P_c = \mathbf{P}(C = c)$ be the probability that the sequence has exactly c contributing features. We have:

$$P_c = \binom{F}{c} q^c p^{F-c} \tag{6.27}$$

where $q = 1 - p$. We note that $|\mathcal{C}_c| = P_c \cdot \left| \mathcal{A}^L \right|$; in particular, $\mathcal{C}_0$, the subset of sequences of fitness zero, occupies a fraction p^F of the sequence space. In general (as will be seen below), to achieve a reasonable degree of neutrality, p will lie close to 1; i.e. we will have $q \ll 1$. In this case $\mathcal{C}_0$ will occupy a fraction $\approx 1 - Fq$ of the sequence space.

It is clear that in general the fitness of a sequence will depend in large part on its number of contributing features, since the fitness of a sequence with c contributing features is a sum of c r.v.'s iid as $\frac{1}{F}Y$. Specifically, the distribution of W conditional on C is given by:

$$\mathbf{P}(W < w \mid C = c) = \mathbf{P}\left(\frac{1}{F}(Y_1 + \ldots + Y_c) < w\right) \tag{6.28}$$

for real w, where $Y_1, Y_2, \ldots, Y_c$, the c nonzero fitness contributions to W, are iid as Y. Thus knowing the number of contributing features for a sequence tells us at least something about the fitness of that sequence. E.g. we have:

$$\mathbf{E}(W \mid C = c) = \frac{c}{F}\mathbf{E}(Y) \tag{6.29}$$

$$var(W \mid C = c) = \frac{c}{F^2} var(Y) \tag{6.30}$$

$$etc. \tag{6.31}$$

Later we shall examine *fitness-dependent* statistics of NKp landscapes. Here we remark that the "contributing feature-dependent" statistics which we will encounter below go at least some way towards addressing fitness-dependence of statistical properties. As can be seen from Eq. (6.30) this will be particularly true if the variance of the underlying distribution Y is small.

For reference we note that if $m(t)$ is the mgf of Y and $M(t)$ is (as in the previous Section) the mgf of Z then:

$$M(t) = p + qm(t) \tag{6.32}$$

The mgf of the fitness W of a uniform random sequence is then $[p + qm(t/F)]^F$ and we may readily calculate that:

$$\mathbf{E}(W) = q\mathbf{E}(Y) \tag{6.33}$$

$$var(W) = \frac{1}{F} qvar(Y) + \frac{1}{F} pq\mathbf{E}(Y)^2 \tag{6.34}$$

6.2.3 NKp Landscapes - Neutral and Lethal Mutation

We have already seen that auto-correlation does not depend on the underlying distribution. We now investigate neutrality and ***lethal mutations*** - i.e. those that yield a zero fitness mutant.

Neutral mutation probability

Firstly, note that if Γ is a neutral network of an NKp landscape $\mathcal{L}$ then $\Gamma \subseteq \mathcal{C}_c$ for some c; i.e. the neutral networks lie completely within the subsets $\mathcal{C}_c$. We shall return to this point later. Now let us write $\nu(d)$ for the probability that flipping d (arbitrary) loci of a (uniform) randomly selected sequence is neutral; i.e.:

$$\nu(d) = \mathbf{P}\left(W^{(d)} = W\right) \tag{6.35}$$

where $W, W^{(d)}$ are as in the previous Section. Let us write $\nu(d|c)$ for the probability that d flips of an arbitrary sequence x are neutral given that $x \in \mathcal{C}_c$; i.e.:

$$\nu(d|c) = \mathbf{P}\left(W^{(d)} = W \;\middle|\; C = c\right) \tag{6.36}$$

and we have:

$$\nu(d) = \sum_{c=0}^{F} P_c \cdot \nu(d|c) \tag{6.37}$$

Now suppose that $x \in \mathcal{C}_c$ and that flipping d loci of x is neutral. Then none of those d loci may influence any of the c contributing features. Furthermore, if any of the d loci influences a *non*-contributing feature then the fitness contribution of that feature *after* flipping the d loci must also be zero - which will occur with probability p. Now the probability that a feature is not influenced by any of the d loci is (from the previous Section) just $\rho(d)$. Thus the probability that none of the c contributing features and exactly r (say) of the $F-c$ non-contributing features is influenced by (at least one of) the d flips is $\binom{F-c}{r}(1-\rho(d))^r \rho(d)^{F-r}$. Putting this together, we find:

$$\nu(d|c) = \rho(d)^c [p + q\rho(d)]^{F-c} \tag{6.38}$$

which, with Eq. (6.37), yields:

$$\nu(d) = \left[\rho(d) + p^2(1-\rho(d))\right]^F \approx e^{-d\left(1-p^2\right)F\kappa} \tag{6.39}$$

(see also (Barnett, 1998)) where the approximation holds (for both the fixed and variable epistasis models) at small Hamming distance d, small epistasis κ and high neutrality p. Note that $\nu(d|c)$ and $\nu(d)$ depend on the epistasis/Hamming distance only via the auto-correlation $\rho(d)$ which, as we have seen, is independent of the underlying distribution and the number of features. Eq. (6.39) thus summarises succinctly the interaction between neutrality, correlation and number of phenotypic features - in short, neutrality:

- *increases* with increasing neutrality parameter p
- *increases* with increasing auto-correlation $\rho(d)$
- *decreases* with increasing number of features F

For Hamming distance $d = 1$ we have $\rho(d) = 1 - \kappa$ (for both the fixed and variable epistasis models) and the expression for $\nu(1)$ takes the particularly simple form:

$$\nu(1) = \left[1 - \left(1 - p^2\right)\kappa\right]^F \approx e^{-\left(1-p^2\right)F\kappa} \tag{6.40}$$

Lethal mutation probability

Analogously to the neutral mutation probabilities, let us write $\lambda(d)$ for the probability that flipping d (arbitrary) loci of a (uniform) randomly selected sequence is ***lethal*** - i.e. yields a sequence of fitness zero:

$$\lambda(d) = \mathbf{P}\left(W^{(d)} = 0\right) \tag{6.41}$$

with $W^{(d)}$ as previously. Since $W^{(d)}$ is identically distributed to W, we have immediately that $\lambda(d) = \mathbf{P}(W = 0) = \mathbf{P}(C = 0) = P_0$; i.e.:

$$\lambda(d) = p^F \tag{6.42}$$

Again, lethality depends on fitness only via the number of contributing features. We thus write $\lambda(d|c)$ for the probability that d flips of an arbitrary sequence are lethal given $x \in \mathcal{C}_c$; i.e.:

$$\lambda(d|c) = \mathbf{P}\left(W^{(d)} = 0 \;\middle|\; C = c\right) \tag{6.43}$$

An argument similar to that for the neutral case gives:

$$\lambda(d|c) = [p(1-\rho(d))]^c [p + q\rho(d)]^{F-c} \tag{6.44}$$

Neutral degree distribution

Another statistic of interest is the distribution of ***neutral degree*** of sequences. We define the neutral degree of a sequence to be the fraction of the L possible 1-bit mutations of that sequence which are neutral. Neutral degree relates to the *connectivity* of neutral networks and is of interest in particular as regards the phenomenon of *mutational buffering* or *mutational robustness* (A. Wagner & Stadler, 1999; Wilke, 2001) - whereby sequences in a *population* diffusing on a neutral network will tend to be found preferentially in regions of the network where the local connectivity is highest. (We note that this effect does not occur for population-of-one *hill-climbers*; see (Nimwegen et al., 1999) for a detailed analysis.)

Thus, for a (uniform random) sequence, let the r.v. Δ be the fraction of loci at which a 1-bit mutation is neutral. Let us write $\chi(n)$ for the probability that exactly n 1-bit mutations are neutral:

$$\chi(n) = \mathbf{P}\left(\Delta = \frac{n}{L}\right) \tag{6.45}$$

and let $\chi(n|c)$ be the probability that exactly n 1-bit mutations are neutral given that the sequence has c contributing features:

$$\chi(n|c) = \mathbf{P}\left(\Delta = \frac{n}{L} \;\middle|\; C = c\right) \tag{6.46}$$

so that:

$$\chi(n) = \sum_{c=0}^{F} P_c \cdot \chi(n|c) \tag{6.47}$$

We note that - from the Central Property for NKp landscapes - a 1-bit mutation of a sequence at a particular locus is neutral iff (i) that locus influences only *non*-contributing features for the sequence and (ii) the fitness contribution *after mutation* of each such feature is also zero; i.e. the feature remains non-contributing after mutation. Here we calculate the distribution of Δ only for the *variable epistasis* model; the calculation in this case is simpler since, for any subset of loci, the probabilities that each locus influence a given feature are mutually independent. For the interested

reader, a calculation (of mean and variance of Δ) for the fixed epistasis case (in fact for the more general class of additive random landscapes) may be found in (Reidys & Stadler, 2001).

For the variable epistasis case then, we may calculate that:

$$\chi(n|c) = \binom{L}{n} \nu(1|c)^n (1-\nu(1|c))^{L-n} \tag{6.48}$$

since the probability that flipping a particular single locus is neutral is just:

$$\nu(1|c) = (1-\kappa)^c (1-q\kappa)^{F-c} \tag{6.49}$$

as previously calculated in Eq. (6.38). In particular, we have:

$$\mathbf{E}(\Delta \mid C=c) = \nu(1|c) \tag{6.50}$$

$$var(\Delta \mid C=c) = \frac{1}{L}\nu(1|c)(1-\nu(1|c)) \tag{6.51}$$

The expected fraction of 1-bit mutations which are neutral is, of course, just the 1-bit neutral mutation probability. The (global) mean neutral degree may be calculated to be:

$$\mathbf{E}(\Delta) = \nu(1) = \left[1-\left(1-p^2\right)\kappa\right]^F \tag{6.52}$$

and we may calculate similarly the variance $var(\Delta)$ (see also (Reidys & Stadler, 2001)); the result is not particularly illuminating.

6.3 Statistical Analysis - Fitness-Dependent Structure

In the Introduction to this thesis it was mentioned that the global statistical properties of a fitness landscape are not necessarily particularly useful, since the sequences sampled by an evolutionary process are by no means uniform - in particular, they are (hopefully!) biased towards *fitter* sequences. This is particularly relevant for NKp landscapes - we saw in the previous Section that they are structurally far from homogeneous. In particular, "most" sequences in an NKp landscape lie in the "uninteresting" zero-fitness subspace $\mathcal{C}_0$ and any global statistics based on *uniform* sampling of the sequence space will consequently be biased towards the zero-fitness sequences. We remark that this appears to be a common feature of non-trivial "real world" optimisation problems (Thompson, 1996; Thompson, 1998; Cliff et al., 1993; Jakobi et al., 1995; Smith et al., 2001; Layzell, 2001; Harvey & Thompson, 1996; Harvey, 1997). In this Section, therefore, we examine the statistics of (mutants of) of a uniform randomly selected sequence *conditional on the fitness of the un-mutated sequence* (*cf.* Section 2.3).

6.3.1 NK Landscapes - Mean Mutant Fitness

We now investigate the distribution of $W^{(d)}$ conditional on $W=w$ for specific fitness values w, where as before W is the fitness of an arbitrary (i.e. uniform random) sequence in $\mathcal{A}^L$ and $W^{(d)}$ is the fitness (evaluated on the same landscape) of the same sequence with d arbitrary loci flipped. In particular, we shall calculate the ensemble mean mutant fitness (see Eq. (2.77)) for constant, uniform d-flip mutation:

$$\mathcal{F}(d|w) = \mathcal{F}\left(U^{(d)}|w\right) = \mathbf{E}\left(W^{(d)} \,\middle|\, W=w\right) \tag{6.53}$$

for $d=1,2,\ldots,L$. To this end, we first establish a technical lemma:

Lemma 6.3.1. *Let Z be a real-valued r.v. (with finite mean) and let $Z_1, Z_2, \ldots$ be iid as Z. Then for $n = 1, 2, \ldots$, $m > n$ and real w:*

$$\mathbf{E}\left(Z_1 + Z_2 + \ldots + Z_n \mid Z_1 + Z_2 + \ldots + Z_m = z\right) = \frac{n}{m} w \tag{6.54}$$

Proof. Since all the Z_i are iid as Z, the LHS of the above remains unchanged if we replace any of the n Z_i on the left of the "|" with any of the m Z_i on the right of the "|". From linearity of (conditional) expectation, adding up the $\binom{m}{n}$ possible combinations and noting that each Z_i appears $\binom{m-1}{n-1}$ times on the left of the "|" in the resulting sum, we find:

$$\begin{aligned}
& \binom{m}{n} \mathbf{E}\left(Z_1 + Z_2 + \ldots + Z_n \mid Z_1 + Z_2 + \ldots + Z_m = w\right) \\
= \quad & \mathbf{E}\left(\binom{m-1}{n-1} (Z_1 + Z_2 + \ldots + Z_m) \;\middle|\; Z_1 + Z_2 + \ldots + Z_m = w \right) \\
= \quad & \binom{m-1}{n-1} w
\end{aligned}$$

and the result follows. □

As in Section 6.2.1 we condition on the number of features n influenced by at least one of the d flipped loci, to derive (in the notation of Section 6.2.1):

$$\begin{aligned}
\mathcal{F}(d \mid w) &= \sum_{n=0}^{F} P_n(d) \mathbf{E}\left(U'_n + V_{F-n} \mid U_n + V_{F-n} = w\right) \\
&= \sum_{n=0}^{F} P_n(d) \left\{ \mathbf{E}\left(U'_n\right) + \mathbf{E}\left(V_{F-n} \mid U_n + V_{F-n} = w\right) \right\} \\
&= \sum_{n=0}^{F} P_n(d) \left\{ \frac{n}{F} \mathbf{E}(Z) + \frac{F-n}{F} w \right\} \quad \text{by Lemma 6.3.1} \\
&= w + (1 - \rho(d)) (\mathbf{E}(W) - w) \quad \text{by Eq. (6.6)}
\end{aligned}$$

where $\rho(d)$ is as defined in Eq. (6.7). We have thus proved:

Propostion 6.3.1. *Generalised NK fitness landscapes*[8] *are linearly correlated with respect to constant uniform mutation $U^{(d)}$ for $d = 1, 2, \ldots, L$..* ■

Note that this provides an alternative proof that the auto-correlation is indeed the $\rho(d)$ of Eq. (6.7); but note too that Prop. 6.3.1 is a *much stronger* statement than Prop. 6.2.1, which says just that the $\rho(d)$ of Eq. (6.7) is the auto-correlation.

As remarked in Chapter 2, linear correlation appears to be a remarkably ubiquitous phenomenon - we have already seen that the Royal Road landscapes of the previous Chapter are (at least approximately) linearly correlated. Preliminary research by the author (unpublished) suggests that linear correlation holds (again approximately) for RNA secondary structure folding landscapes where fitness is taken as the (tree-edit) distance (Hofacker et al., 1994) from a predefined target structure (see also (Huynen et al., 1996)), for some 1-dimensional cellular automata

[8] It would seem that the above proof should generalise to any *additive random landscape* (Reidys & Stadler, 2001).

classification landscapes (Mitchell, Crutchfield, & Das, 2000) and for some fitness landscapes based on recurrent dynamic neural networks (Beer, 1995).

We propose that linear correlation might be applied to the analysis of fitness distributions (at least in the infinite population limit), using the "statistical dynamics" techniques of (Nimwegen et al., 1997). However, it does not tell us anything about other fitness-dependent phenomena of interest, such as fitness-dependent neutrality (for NKp landscapes) or beneficial mutation probabilities - this would require knowledge of the full joint distribution (or at least higher moments) of $W, W^{(d)}$. In the following sub-sections we address these phenomena.

6.3.2 NKp Landscapes - Fitness-dependence of Neutral and Lethal Mutation

Before proceeding we introduce some notation. Let the underlying distribution of an NK landscape be Z and let $Z_1, Z_2, \ldots$ be iid as Z. We set:

$$\phi_n(w) = \text{pdf of } \frac{1}{F}(Z_1 + Z_2 + \ldots + Z_n) \tag{6.55}$$

with the convention that $\phi_0(w) = \delta(w)$, the Dirac delta pdf. In particular:

$$\phi_F(w) = \text{pdf of } W \tag{6.56}$$

Now suppose we have an NKp landscape with neutrality parameter p and underlying distribution Y, so that Z is given by Eq. (6.1). Let $Y_1, Y_2, \ldots$ be iid as Y. We then set:

$$\psi_c(w) = \text{pdf of } \frac{1}{F}(Y_1 + Y_2 \ldots + Y_c) \tag{6.57}$$

again with the $\psi_0(w) = \delta(w)$ and we have, conditioning on the number of contributing features:

$$\phi_n(w) = \sum_{c=0}^{n} \binom{n}{c} q^c p^{n-c} \, \psi_c(w) \tag{6.58}$$

Finally, we set:

$$\begin{aligned} \gamma(c|w) &= \mathbf{P}(C = c \mid W = w) \\ &= \begin{cases} \binom{F}{c} q^c p^{F-c} \frac{\psi_c(w)}{\phi_F(w)} & c > 0,\ w \neq 0 \\ 1 & c = 0,\ w = 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{6.59}$$

As will be seen below, the quantities $\gamma(c|w)$ - the probabilities of c contributing features given a fitness of w - may be used to derive fitness-dependent statistics from statistics which depend only on the number of contributing features.

We have previously calculated the probability $\nu(d|c)$ that flipping d alleles of an (arbitrary) sequence with c contributing features is neutral. From the analysis of Section 6.2.3 it is easy to see that *conditional on the number of contributing features*, the probability of d flips being neutral is independent of the actual fitness of the sequence. That is:

$$\mathbf{P}\left(W^{(d)} = W \;\middle|\; W = w, C = c\right) = \mathbf{P}\left(W^{(d)} = W \;\middle|\; C = c\right) = \nu(d|c) \tag{6.60}$$

Then, setting[9] $\nu(d|w) = \mathbf{P}\left(W^{(d)} = W \mid W = w\right)$ and conditioning on the number of contributing features, we have:

$$\nu(d|w) = \sum_{c=0}^{F} \nu(d|c)\,\gamma(c|w) \tag{6.61}$$

where $\nu(d|c)$ is given by Eq. (6.38) and $\gamma(c|w)$ by Eq. (6.59). This formula may be used to calculate $\nu(d|w)$ for a particular underlying distribution Y. Similarly, for lethal mutation, setting $\lambda(d|w) = \mathbf{P}\left(W^{(d)} = 0 \mid W = w\right)$ we have:

$$\lambda(d|w) = \sum_{c=0}^{F} \lambda(d|c)\,\gamma(c|w) \tag{6.62}$$

with $\lambda(d|c)$ given by Eq. (6.44). Fig. 6.1 plots $\nu(d|w)$ and $\lambda(d|w)$ against d, w for a range of values, for a Gaussian underlying distribution (i.e. Y is normally distributed). For the fitness-dependent distribution of neutral degree, we may calculate $\chi(n|w) = \mathbf{P}\left(\Delta = \frac{n}{L} \mid W = w\right)$ as:

$$\chi(n|w) = \sum_{c=0}^{F} \chi(n|c)\,\gamma(c|w) \tag{6.63}$$

with $\chi(n|c)$ as in Eq. (6.48). This yields in particular:

$$\mathbf{E}\left(\Delta \mid W = w\right) = \nu(1|w) \tag{6.64}$$

$$var\left(\Delta \mid W = w\right) = \left(1 - \frac{1}{L}\right) \sum_{c=0}^{F} \nu(1|c)^2\,\gamma(c|w) + \frac{1}{L}\nu(1|w) - \nu(1|w)^2 \tag{6.65}$$

Fig. 6.2 plots $var\left(\Delta \mid W = w\right)$ against a range of w values and auto-correlation $\rho(1) = 1 - \kappa$, again for a normally distributed underlying distribution Y.

6.3.3 NKp Landscapes - Mutant Distribution

In this section we will examine the full distribution of the fitness of a d-point mutant on an NKp landscape - that is, the distribution of $W^{(d)}$ conditional on $W = w$. We calculate the mutant fitness distribution explicitly for a Gaussian underlying distribution and use it to calculate the *evolvability* (Section 2.3.4):

$$\mathcal{E}\left(d\,|w\right) = \mathcal{E}\left(U^{(d)}\,|w\right) = \mathbf{P}\left(W^{(d)} > w \;\middle|\; W = w\right) \tag{6.66}$$

Let us define the (conditional) mgf:

$$M^{(d)}(t|w) = \mathbf{E}\left(e^{tW^{(d)}} \;\middle|\; W = w\right) \tag{6.67}$$

The obstacle to calculation of $M^{(d)}(t|w)$ is that there is no analog of Lemma 6.3.1 for higher moments. The best we can do is calculate the distribution for specific underlying distributions Y. We have:

$$\begin{aligned} M^{(d)}(t|w) &= \sum_{n=0}^{F} P_n(d)\,\mathbf{E}\left(\exp\left(t(U'_n + V_{F-n})\right) \mid U_n + V_{F-n} = w\right) \\ &= \sum_{n=0}^{F} P_n(d)\,\mathbf{E}\left(\exp\left(tU'_n\right)\right)\mathbf{E}\left(e^{tV_{F-n}} \mid U_n + V_{F-n} = w\right) \end{aligned}$$

[9] Note that for $w = 0$ our notation is consistent with the definition Eq. (6.36) of $\nu(d|c)$, since $W = 0 \Leftrightarrow C = 0$ (a.s.).

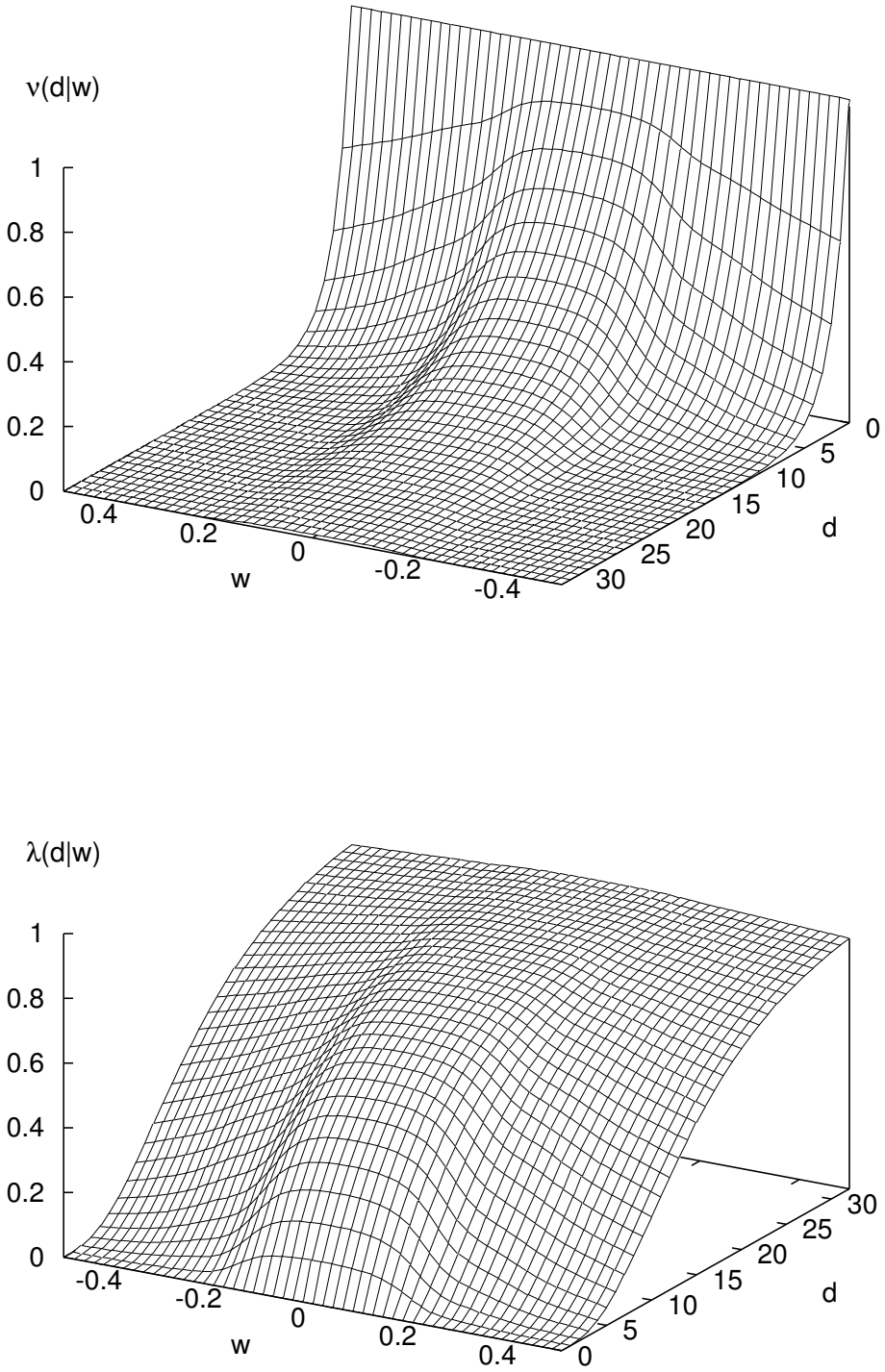

Figure 6.1: Fitness-dependent neutrality (top figure) and lethal mutation probability (bottom figure) for NKp landscapes plotted against d, w for a range of w values. Parameters: variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 20$, $N = 32$, $\kappa = 0.125$, $p = 0.99$.

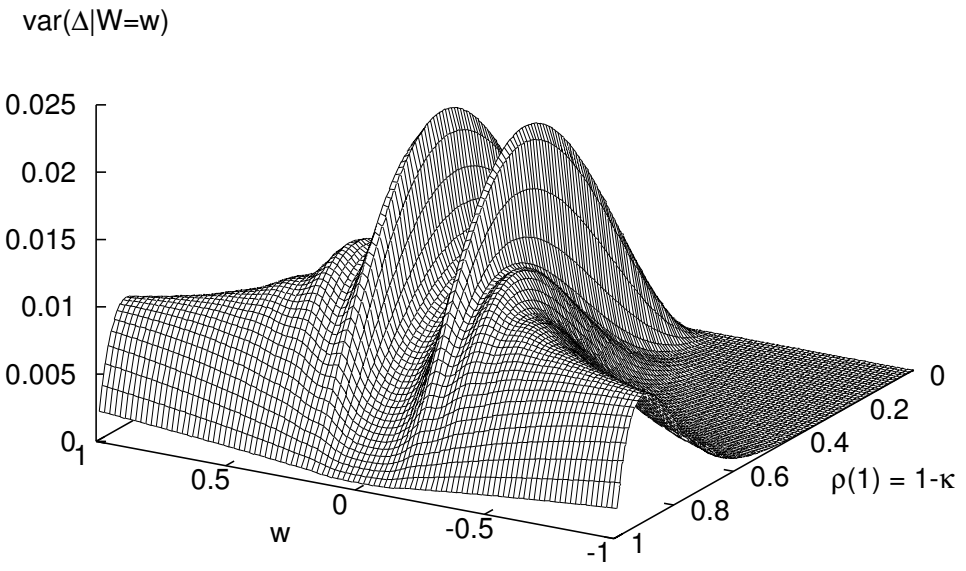

Figure 6.2: Fitness-dependent neutral degree variance $var(\Delta \mid W = w)$ plotted against a range of w values and auto-correlation $\rho(1) = 1 - \kappa$. Parameters: variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 20$, $N = 32$, $\kappa = 0.125$, $p = 0.99$.

$$
\begin{aligned}
&= \sum_{n=0}^{F} P_n(d)\, M(t/F)^n\, \mathbf{E}\left(\exp\left(tV_{F-n}\right) \mid U_n + V_{F-n} = w\right) \\
&= \sum_{n=0}^{F} P_n(d)\, M(t/F)^n \int_v e^{tv} \phi_n(w-v)\phi_{F-n}(v)\phi_F(w)^{-1} dv \\
&= \sum_{n=0}^{F} P_n(d)\, M(t/F)^n \sum_{a=0}^{n} \sum_{c=a}^{F-n+a} \binom{n}{a}\binom{F-n}{c-a} q^c p^{F-c} \frac{\psi_c(w)}{\phi_F(w)} \\
&\quad \times \int_v e^{tv} \psi_a(w-v)\psi_{c-a}(v)\psi_c(w)^{-1} dv \\
&= \sum_{n=0}^{F} P_n(d)\, M(t/F)^n \sum_{a=0}^{n} \sum_{c=a}^{F-n+a} \binom{n}{a}\binom{F-n}{c-a}\binom{F}{c}^{-1} \gamma(c|w) \\
&\quad \times \int_v e^{tv} \psi_a(w-v)\psi_{c-a}(v)\psi_c(w)^{-1} dv
\end{aligned}
$$

Expanding $P_n(d)$ and rearranging some terms we get:

$$
\begin{aligned}
M^{(d)}(t|w) &= \sum_{c=0}^{F} \gamma(c|w) \sum_{a=0}^{c} \sum_{n=a}^{F-c+a} \binom{c}{a}\binom{F-c}{n-a} \rho(d)^{F-n}(1-\rho(d))^n\, M(t/F)^n \\
&\quad \times \int_v e^{tv} \psi_a(w-v)\psi_{c-a}(v)\psi_c(w)^{-1} dv
\end{aligned} \tag{6.68}
$$

The integral in this expression may be thought of as the mgf of $\frac{1}{F}\left(Y_1 + \ldots + Y_{c-a}\right)$ conditional on $\frac{1}{F}\left(Y_1 + \ldots + Y_c\right) = w$. We calculate the corresponding conditional distributions for the case where the underlying distribution Y is Gaussian.

Gaussian underlying distribution

Suppose now that the underlying distribution for our NKp landscape is Gaussian with $\frac{1}{F}Y \sim N\left(0,\, \sigma^2\right)$. Noting that for $c = 1, 2, \ldots$ we have $\frac{1}{F}\left(Y_1 + \ldots + Y_c\right) \sim N\left(0,\, c\,\sigma^2\right)$ we have:

$$
\psi_c(w) = \frac{1}{\sqrt{2\pi}}\, \frac{1}{\sqrt{c}\,\sigma}\, \exp\left(-\frac{1}{2}\frac{w^2}{c\,\sigma^2}\right) \tag{6.69}
$$

and we may calculate $\gamma(c|w)$ immediately from Eq. (6.59). Next we state:

Propostion 6.3.2. *Let Y_1, Y_2 be independent Gaussian r.v.'s with $Y_1 \sim N\left(0,\, \sigma_1^2\right)$ and $Y_2 \sim N\left(0,\, \sigma_2^2\right)$. Then:*

$$
Y_1 \mid (Y_1 + Y_2 = w) \sim N\left(\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\, w,\; \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right) \tag{6.70}
$$

Proof. Straightforward calculation. □

Setting $\sigma_1^2 = (c-a)\sigma^2$ and $\sigma_2^2 = a\sigma^2$ in Prop. 6.3.2 we thus find:

$$
\int_v e^{tv} \psi_a(w-v)\psi_{c-a}(v)\psi_c(w)^{-1} dv = \exp\left(\frac{c-a}{c} wt + \frac{1}{2}\frac{a(c-a)}{c}\sigma^2 t^2\right) \tag{6.71}
$$

which is the mgf of a Gaussian distribution with mean $\frac{c-a}{c}w$ and variance $\frac{a(c-a)}{c}\sigma^2$. Note that the mean - as it must be by Lemma 6.3.1 - is linear in w. We also have:

$$
M(t/F) = p + q\, m(t/F) = p + q\, \exp\left(\frac{1}{2}\sigma^2 t^2\right) \tag{6.72}
$$

Putting this all together, we find:

$$\begin{aligned} M^{(d)}(t|w) &= \sum_{c=0}^{F} \gamma(c|w) \sum_{a=0}^{c} \binom{c}{a} \sum_{n=a}^{F-c+a} \binom{F-c}{n-a} \rho(d)^{F-n}(1-\rho(d))^n \sum_{b=0}^{n} \binom{n}{b} q^b p^{n-b} \\ &\quad \times \exp\left(\frac{c-a}{c} wt + \frac{1}{2}\left[\frac{a(c-a)}{c} + b\right]\sigma^2 t^2\right) \end{aligned} \tag{6.73}$$

We note that this equation is singular when $c=0$ or when $b=0$ and $a=0$ or $a=c$. Now $c=0$ corresponds to $C=0 \Leftrightarrow W=0$ (a.s.). For $w \neq 0$, then, $\gamma(0|w)=0$ so that the $c=0$ term does not contribute and should be omitted from the summation. Still for $w \neq 0$, the $b=0,\ a=0$ term corresponds to a Dirac delta distribution around w, while the $b=0,\ a=c$ corresponds to a Dirac delta distribution around 0. Expanding the $b=0,\ a=0$ term for given $c>0$, we find that the coefficient of the delta distribution is just the conditional neutrality $\nu(d|w)$ as given by Eq. (6.61). The coefficient of the delta distribution for the $b=0,\ a=c$ term may similarly be calculated to be the conditional lethal mutation probability $\lambda(d|w)$ of Eq. (6.62). For $w \neq 0$ we thus write symbolically:

$$\begin{aligned} W^{(d)}\Big|(W=w) &\sim \nu(d|w)\, D(w) \\ &+ \lambda(d|w)\, D(0) \\ &+ \sum_{c=1}^{F} \gamma(c|w) \sum_{a=0}^{c} \binom{c}{a} \sum_{n=a}^{F-c+a} \binom{F-c}{n-a} \rho(d)^{F-n}(1-\rho(d))^n \sum_{b=0}^{n} \binom{n}{b} q^b p^{n-b} \\ &\quad \times N\left(\frac{c-a}{c} w,\ \left[\frac{a(c-a)}{c} + b\right]\sigma^2\right) \end{aligned} \tag{6.74}$$

where the summations are to be understood as *superpositions* of distributions, $D(\cdot)$ indicates a Dirac delta distribution and the terms for $a=0,\ b=0$ and $a=c,\ b=0$ are to be omitted in the summation. The $w=0$ case yields:

$$\begin{aligned} W^{(d)}\Big|(W=0) &\sim \nu(d|0)\, D(0) \\ &+ \sum_{n=0}^{F} \binom{F}{n} \rho(d)^{F-n}(1-\rho(d))^n \sum_{b=1}^{n} \binom{n}{b} q^b p^{n-b}\, N\left(0,\ b\sigma^2\right) \end{aligned} \tag{6.75}$$

Fig. 6.3 plots the continuous part of the conditional probability density function $\varphi^{(d)}(w'|w)$ of the distribution of $W^{(d)}\big|(W=w)$ against w' for several (positive) values of w, for $d=1,2$. Note that as w increases, the distribution becomes increasingly multi-modal. From Eq. (6.74) we may calculate the evolvability for $w \neq 0$ to be:

$$\begin{aligned} \mathcal{E}(d|w) &= \lambda(d|w)(1-H(w)) \\ &+ \sum_{c=1}^{F} \gamma(c|w) \sum_{a=0}^{c} \binom{c}{a} \sum_{n=a}^{F-c+a} \binom{F-c}{n-a} \rho(d)^{F-n}(1-\rho(d))^n \sum_{b=0}^{n} \binom{n}{b} q^b p^{n-b} \\ &\quad \times \Psi\left(\frac{a}{\sqrt{c\,(ac+bc-a^2)}} \frac{w}{\sigma}\right) \end{aligned} \tag{6.76}$$

(omit $a=0,\ b=0$ and $a=c,\ b=0$ terms) where:

$$H(w) = \begin{cases} 1 & w > 0 \\ 0 & w \leq 0 \end{cases} \tag{6.77}$$

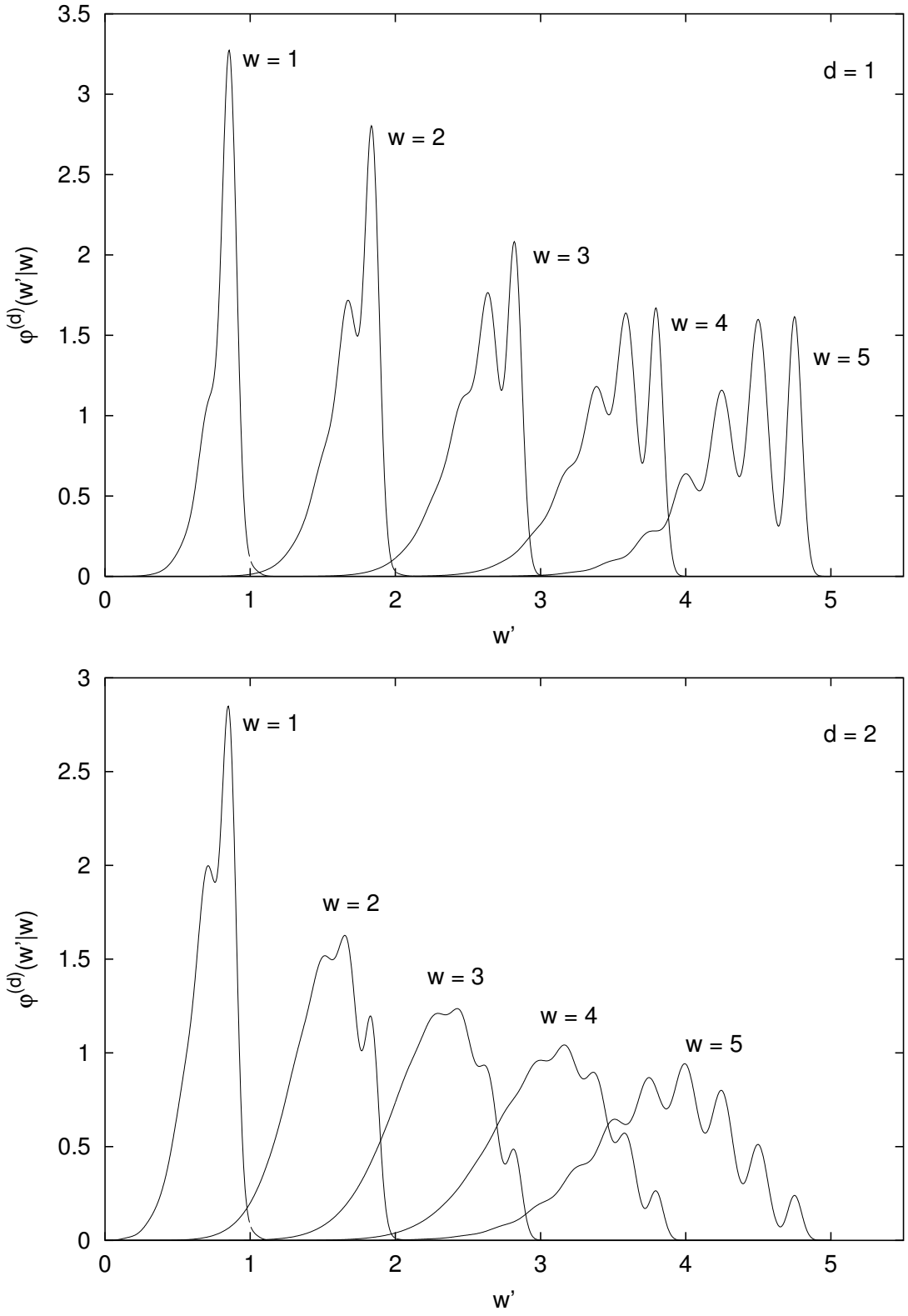

Figure 6.3: The conditional probability density function $\varphi^{(d)}(w'|w)$ of the continuous part of the NKp mutant distribution $W^{(d)}\big|\,(W = w)$ of Eq. (6.74) at Hamming distance $d = 1$ (top figure) and $d = 2$ (bottom figure), plotted against w' for several (positive) values of w. Parameters: variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 20$, $N = 32$, $\kappa = 0.125$, $p = 0.99$.

is the *Heaviside* (step) distribution function and:

$$\Psi(w) = \frac{1}{\sqrt{2\pi}} \int_w^\infty \exp\left(-\frac{1}{2}u^2\right) du \tag{6.78}$$

is the *complementary error function* (Gaussian tail), defined by $\Psi(w) = \mathbf{P}(Y > w)$ for $Y \sim N(0,\ 1)$. Note that the neutral term $\nu(d|w)\ D(w)$ does not, of course, contribute to evolvability. For $w = 0$ it is clear (by symmetry) that:

$$\mathcal{E}(d\,|0) = \frac{1}{2} - \frac{1}{2}\nu(d|0) \tag{6.79}$$

i.e. given that a mutation of a fitness zero sequence is not neutral, it has an equal chance of being fitness-increasing or fitness-decreasing. Fig. 6.4 plots $\mathcal{E}(d\,|w)$ against d, w over a range of (non-negative) w values (see also (Smith et al., 2001; Smith, Husbands, et al., 2002)).

The presence of the Gaussian tail in the expression for evolvability indicates a *decay* of the order of e^{-kw^2}/w for some k of evolvability against fitness. This rapid decay suggests that NKp landscapes will be *hard* to optimise (Section 6.4.4) - as we move up the landscape fitness-increasing mutations quickly become difficult to find. This may be compared with the (approximately) *linear* evolvability decay (Section 5.4.1) of the Royal Road landscapes of the previous Chapter, which are consequently far easier to optimise (Section 5.4.2).

Optimal mutation rates

It is evident that for a given fitness w there must be an *optimal* mutation rate (*cf.* Section 2.3.4); that is, a $d = d^*(w)$ which maximises $\mathcal{E}(d\,|w)$, the probability that mutation finds a higher fitness sequence (this may be seen clearly in Fig. 6.4). Note that d only enters Eq. (6.76) via the autocorrelation term $\rho(d)$. Now $\rho(d) = (1-\kappa)^d$ for the variable epistasis model and $\approx (1-\kappa)^d$ for the fixed model for small d, κ, so that $\rho'(d) = (1-\kappa)^d \log(1-\kappa)$. For fixed w we can thus differentiate $\mathcal{E}(d\,|w)$ with respect to d, set the result to zero and solve (at least numerically) for an optimal d. This may be extended to other mutation modes; for instance, for Poisson mutation we may calculate an optimum per-sequence mutation rate $\overline{u}^*(w)$.

From Eq. (6.79) we see that for the particular case $w = 0$ this amounts to minimising the probability that a d-bit mutation is *neutral*; but this implies setting d to its maximum value of L: if we have a sequence of fitness zero, we should flip *every* bit! This may seem peculiar, until we note the essential difference from e.g. the situation in the previous Chapter. There we were dealing with a single fitness landscapes. Here we are dealing with *ensemble* evolvability statistics, which implies that in collating the statistics through sampling we sample a different landscape on each trial. Our conclusion - that we should flip every bit - is correct. It is, however, evidently less than useful, as it does not tell us what to do if our mutant does *not* find an innovation and we have to mutate again; Eq. (6.79) then only tells us what we might expect for a *different landscape*! (We certainly do not, for instance, wish to flip all L loci back to their original alleles...)

This is, in a sense, an inherent problem with ensemble statistics, at least insofar as they don't "self-average" (as is evidently the case for the $\mathcal{E}(d\,|w)$ of Eq. (6.76)). A more meaningful statistic - in the current case of evolvability - might be the probability that of, say, k uniform randomly selected d-flip mutants of k sequences of given fitness *on the same NKp landscape*, (at least) one of them be fitness-increasing. We might then attempt to find a mutation rate so as to minimise the expected time to discovery of such beneficial mutations. This approach, while perhaps not entirely intractable to analysis, would certainly involve far more work... we do not pursue it here.

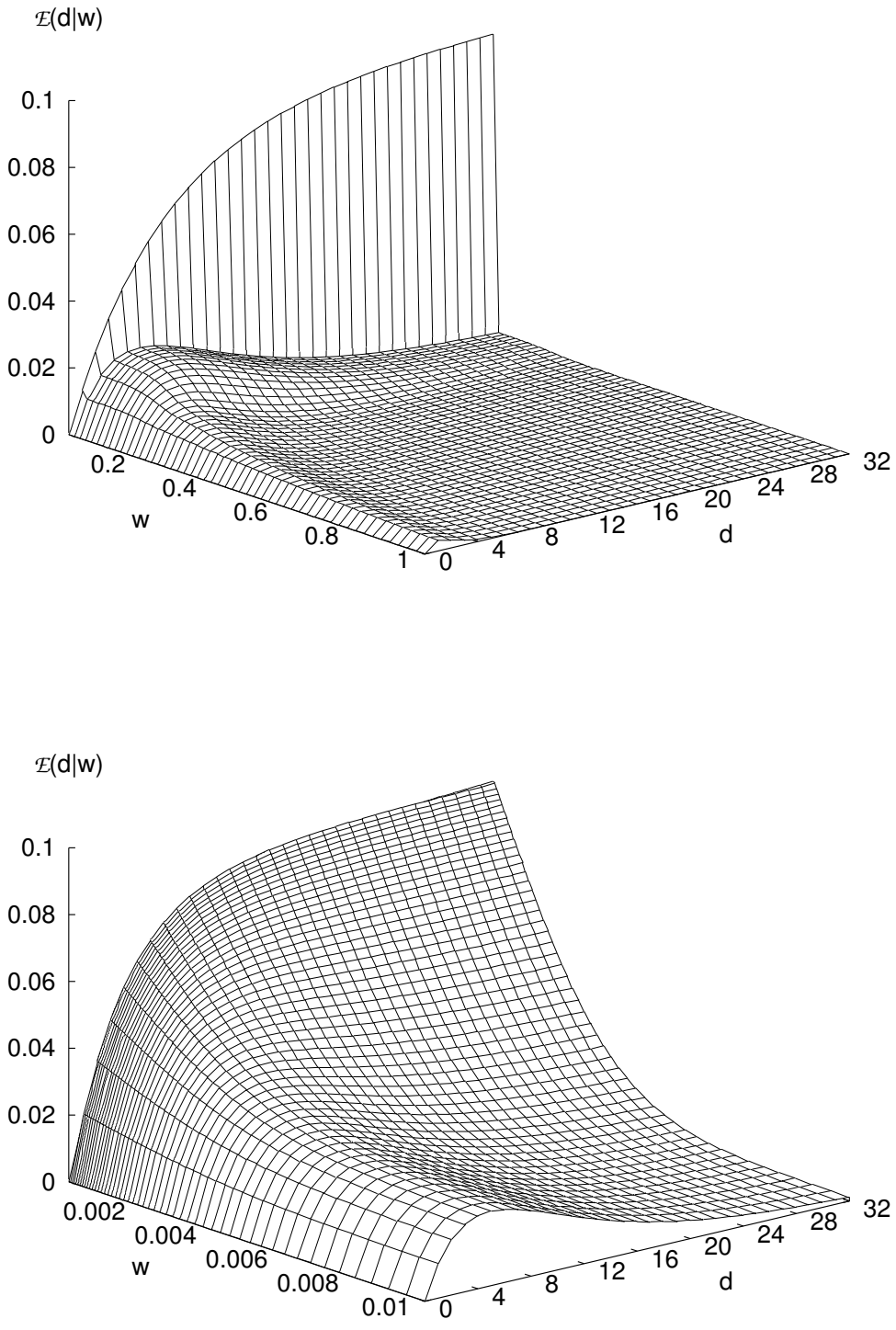

Figure 6.4: The NKp evolvability statistic $\mathcal{E}(d\,|w)$ of Eq. (6.76) and Eq. (6.79) plotted against d and $0 \leq w \leq 1$ (top figure) and $0 \leq w \leq 0.01$ (bottom figure). Parameters: variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 20$, $N = 32$, $\kappa = 0.125$, $p = 0.99$.

Nonetheless, the mutation rate $d^*(w)$ which maximises $\mathcal{E}(d|w)$ might still be useful as a "heuristic" for setting a mutation rate in the circumstance that we are attempting to optimise on an (unknown) member of an NKp family of landscapes. We shall test this hypothesis below.

The "$1/e$ rule" revisited

Another possibility was raised in the conclusion to the previous Chapter: we might assume that NKp landscapes are "locally ε-correlated" and use an on-the-fly neutrality estimate plus our $1/e$ rule to set a mutation rate. We may justify this procedure further by the following argument, which suggests that the $1/e$ rule may apply in more general circumstances than ε-correlated landscapes. Inman Harvey[10] has espoused the idea that, in the presence of neutrality *and correlation*, in order to maximise the probability of finding a portal to a higher-fitness network by mutation we should mutate "just enough" to get off the current network but (due to the assumption of correlation) at the same time to stay as near as possible to the current network. We already know from Prop. 2.3.1 that we should use constant mutation - i.e. flip a fixed number of bits. However, without detailed knowledge of the local network topology it would not seem possible to calculate what the optimal *rate* might be. As a crude approximation let us suppose that neutral networks in our landscape are (at least locally) "block-like": suppose specifically that in the locality of our sequence - i.e. within small Hamming distances - there are n "neutral loci" and $L-n$ "non-neutral loci", so that locally $\nu = \frac{n}{L}$. Then if we flip exactly d (arbitrary) loci, the probability that k of these loci are *non*-neutral and $d-k$ are neutral is given by:

$$\binom{L-n}{k}\binom{n}{d-k}\Big/\binom{L}{d} \approx \binom{d}{k}\nu^{d-k}(1-\nu)^k \tag{6.80}$$

where the approximation holds for $d \ll n$ - a reasonable assumption if neutrality is high and, as we are in any case assuming, the "block-like" approximation holds for small Hamming distances. Now in order to "get off the network" but remain "as close as possible" to it, we want to choose d so as to maximise the probability that $k=1$. This is tantamount to choosing d so as to maximise $d\nu^{d-1}$. As in Prop. 5.2.1 of the previous Chapter we find that the optimal rate d is approximated by the nearest integer to $-\frac{1}{\log\nu}$ which implies the $1/e$ rule for observed neutrality. For Poisson mutation we may check that the optimal (per-sequence) rate $\bar{u}$ is given, again as in Chapter 5, by $\frac{1}{1-\nu}$, which again yields a $1/e$ rule for observed neutrality.

We remark that for NKp landscapes with reasonably high neutrality and correlation, investigations by the author (not shown) suggest that neutral networks are in fact quite "block-like" locally. To test the viability of $1/e$ Neutral Mutation Rule mutation we calculated, for w in a given range, the optimum mutation rates $d^*(w)$ (resp. $\bar{u}^*(w)$) for fixed (resp. Poisson) mutation from Eq. (6.76). Then, for each w in the range, we calculated the (1-flip) neutrality $\nu = \nu(1|w)$ from Eq. (6.61) and the mutation rates $d_{\text{est}} = -\frac{1}{\log\nu}$ (resp. $\bar{u}_{\text{est}} = \frac{1}{1-\nu}$) predicted by the $1/e$ rule for (1-flip) neutrality ν. These estimated optimum rates were then compared with the "true" optimum rates $d^*(w)$ (resp. $\bar{u}^*(w)$). Except at very small fitness (in which case discovering innovations is comparatively simple and mutation rates non-critical) the estimated optimum rates calculated in this fashion proved to track the true optimum rates surprisingly well. Fig. 6.5 plots a sample calculation for constant and Poisson mutation. As in the previous Chapter there is, due to the diminishing of neutrality with increasing fitness, a tendency slightly to overestimate optimum mutation rates, particularly at low

[10] Personal communication.

fitness. With the caveat regarding ensemble statistics, these results are encouraging and suggest that the $1/e$ rule may be at the very least a useful heuristic for estimating optimal mutation rates on NKp(-like) landscapes. This proposition will be further tested below where we run *adaptive netcrawlers* on NKp landscapes.

6.4 Landscape Modelling with NKp Landscapes

The"typical" scenario for the type of real-world artificial evolutionary optimisation problem we wish to address exhibits the following features:

- large search space - i.e. long sequence lengths
- substantial neutrality, especially at low fitness
- reasonable degree of correlation
- "most" sequences have low/zero fitness
- higher fitness networks "percolate" to some degree - i.e. are accessible via a few mutations from an arbitrary sequence

It is this type of landscape that we hope to model using NKp landscapes. In simulation, for the purposes of gathering statistics (where substantial sampling is likely to be necessary) there will inevitably be a trade-off between realism and time/space/processing constraints. After some experimentation we arrived at the following NKp parameters which (hopefully) capture the features itemised above:

$$\begin{aligned} F &= 40 \\ N &= 64 \\ \kappa &= 0.1875 \quad \text{(variable epistasis)} \\ p &= 0.999 \end{aligned}$$

These parameters settings, which we shall refer to as defining our ***long sequence length baseline*** landscapes, yield the following statistics:

- sequence space size $= 2^{64} \approx 1.84 \times 10^{19}$ sequences
- $\approx 96.08\%$ of landscape is zero fitness
- neutrality at zero fitness ≈ 0.99
- (auto-)correlation at Hamming distance 1 is 0.8125

with network percolation to be investigated.

Now the large sequence space and high degree of "lethality" presents a sampling problem: we are (for reasons already explained) interested in fitness-dependent statistics, but uniform sampling introduces a heavy bias towards lethal/low fitness sequences. It is non-trivial even to *find* higher fitness sequences (if it weren't we would hardly attempt to model a difficult optimisation problem using NKp landscapes!) so that ultimately we must use a search technique to locate higher-fitness sequences. But this will inevitably introduce (probably unknown) biases into our sampling. There

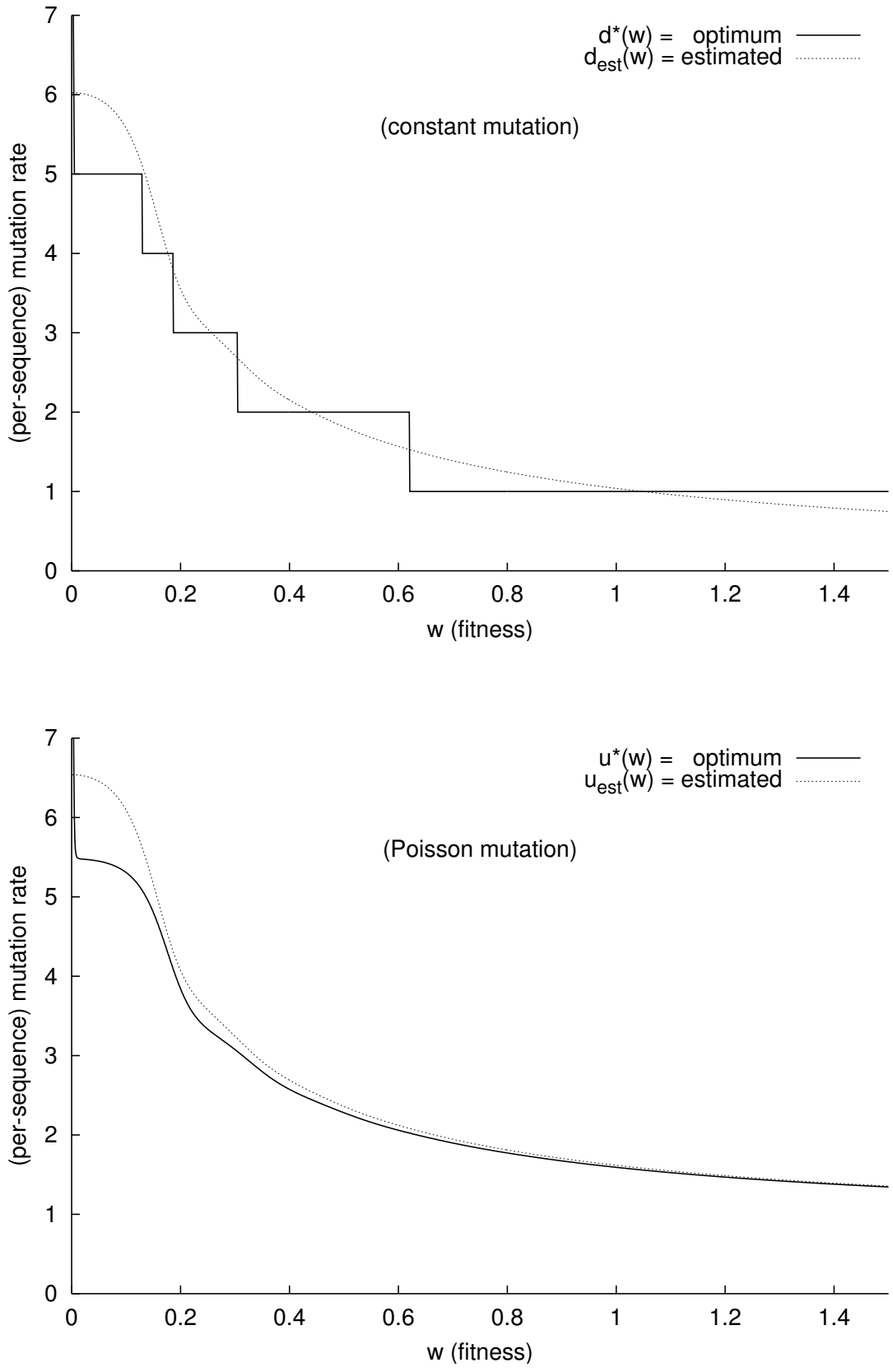

Figure 6.5: Optimum mutation rates calculated from the evolvability statistic (Eq. 6.76) and estimated optimum rates based on neutrality (Eq. 6.61) and the $1/e$ Neutral Mutation Rule (see text) for constant (top figure) and Poisson (bottom figure) mutation modes. Parameters: variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 20$, $N = 32$, $\kappa = 0.125$, $p = 0.99$.

would not seem to be a way around this conundrum. The best we can perhaps do, is to check our sample statistics against *small* (i.e. short sequence length) landscapes, where uniform sampling is feasible by exhaustive search; if our optimisation-based sampling technique produces statistics that tally well with those produced by true uniform sampling at short sequence lengths, we then hope that our methods scale up benignly to longer sequence lengths... Such checks were carried out as far as possible on all statistical analyses presented in the following sub-Sections.

Anticipating some results below on optimising on NKp landscapes, the technique we chose to sample long sequence length landscapes was *simulated annealing* (Chapter 3, Example 3.2.7) with constant mutation at the theoretical (fitness-dependent) optimal rate based on Eq. (6.76). The annealing schedule was as follows[11]: temperature $T(\tau)$ decays exponentially with time. Temperature decay rate and "Boltzmann's constant" k were controlled by two (constant) parameters:

1. a fitness decrement Δw defined by the property that at the start of a run (i.e. $\tau = 0$) a drop in fitness of size Δw is accepted with probability $\frac{1}{2}$
2. a "half-life" $\tau_{1/2}$ such that the temperature halves in $\tau_{1/2}$ time steps

Exhaustive (and exhausting!) experimentation revealed that, for long sequence length baseline landscapes, best results (over a range of time scales) were achieved with parameter settings: $\Delta w = 0.01$ and $\tau_{1/2} = 0.2 \times (\textit{run time})$ (*cf.* Eq. 6.83). It was found by lengthy simulations that the mean maximum fitness achieved by this technique for our baseline landscapes is approximately[12] 0.2 with a standard deviation of approximately 0.03. Due to the atomicity of the underlying distribution around zero, statistics for fitness zero networks were generally omitted from optimising runs and compiled separately.

Of course there are some statistics which, due to the size of the sequence space and of the neutral networks themselves cannot be estimated by sampling. These include neutral network connectivity, distribution of network size and of network number. For these statistics the best we can do is exhaustive sampling of short sequence length landscapes and again hope that (qualitative) results scale to higher sequence lengths. We chose our ***short sequence length baseline*** landscape parameters to be:

$$\begin{aligned} F &= 20 \\ N &= 16 \\ \kappa &= 0.375 \quad \text{(variable epistasis)} \\ p &= 0.99 \end{aligned}$$

yielding:

- sequence space size $= 2^{16} = 65536$ sequences
- $\approx 81.79\%$ of landscape is zero fitness

[11] We experimented with other (possibly more conventional) annealing schedules, but the exponential cooling scheme described here turned out to be the most effective.

[12] We suspect that this is close to the mean global optimum fitness for the long sequence length baseline family of NKp landscapes. Runs of up to $1,000,000$ evaluations were used to derive these results. For comparison, in a run of similar length, *random search* finds a mean maximum fitness of around 0.1.

- neutrality at zero fitness ≈ 0.93
- (auto-)correlation at Hamming distance 1 is 0.625
- network percolation to be investigated

For both long and short sequence length baseline landscapes, the underlying distribution used was Gaussian with variance $\sigma^2 = 1$.

6.4.1 Estimating landscape parameters

A question of interest is the following: suppose we are given an artificial evolution landscape to optimise and we suspect that it may resemble structurally an NKp landscape with variable epistasis and Gaussian underlying distribution (*cf.* the discussion in Chapter 1, Section 1.1.2). How might we then go about verifying our suspicion and estimating the landscape parameters F, κ, p and σ^2 (with a view, perhaps, to exploiting some of the theoretical results of this Chapter)? Let us first suppose that the "ground level" of our landscape - the set $\mathcal{C}_0$ of sequences with zero contributing features - is indeed at fitness zero. This set may be easily established for our unknown landscape by evaluating fitness for a sample of random sequences, the overwhelming majority of which will (by assumption that we can indeed model our landscape as an NKp landscape) yield the same "poor" fitness value. If this value is *not* zero, we may either have to offset fitness by an appropriate amount or, if fitness is always positive and we suspect that the fitness of features aggregate *multiplicatively*, we might redefine fitness to its logarithm. Next we might explore the *correlation* properties of our landscape. Thus we begin to optimise (using, perhaps, a simulated annealer as described above) and gather statistics on the mean fitness of d-bit mutants of sequences of a given fitness. If mean mutant fitness appeared to satisfy the *linear correlation* property (Prop. 6.3.1) then we would have $\mathcal{F}(d|w) = \rho(d)w$. Repeating this process for several values of d, we could then check how well the relation $\rho(d) = (1-\kappa)^d$ (Eq. 6.9) holds up and estimate the epistasis κ. Next we check the neutrality and lethality properties of our landscape. We should have: $\nu(d) \approx \exp\left(-d(1-p^2)F\kappa\right)$ (Eq. 6.39) and $\lambda(d) = p^F$ (Eq. 6.42) for any d. If these properties appear to obtain we may use them to estimate neutrality p and number of features F. It remains to estimate the variance σ^2 of the underlying distribution. We may verify from Eq. (6.34) that if, as in Section 6.3.3, the underlying fitness distribution Y is defined by $\frac{1}{F}Y \sim N\left(0,\ \sigma^2\right)$, then the fitness variance of an arbitrary sequence is given by $var(W) = F(1-p)\sigma^2$. Alternatively, we may calculate from Eq. (6.75) that the variance of a d-bit mutant of a fitness zero sequence is given by $var\left(W^{(d)} \mid W = 0\right) = F(1-p)(1-\rho(d))\sigma^2$. Of course the statistics suggested above to estimate model parameters are *ensemble* statistics, but, we hope, may nonetheless yield useful insights into the structure of our landscape; there is much scope for research in this area.

6.4.2 Notes on NKp computer implementation

Due to the necessarily intensive nature of our statistical sampling, efficiency of implementation was paramount. An inherent problem regarding computer implementation of NK landscapes in general is storage requirements: for each feature, if there are k influencing loci, the fitness table for that feature must contain 2^k (real) numbers. If k is too large, available computer memory resources may be insufficient and the processing overhead of pseudo-random number generation

required to fill the tables unacceptable. Several schemes to overcome these problems have been suggested in the literature - see eg. (Altenberg, 1995). One option to reduce storage requirements is to store *seeds* to the random number generator in tables, rather than fitness itself. Fitness is then calculated "on-the-fly" using the random number generator with the appropriate seed. This effectively shifts resource limitation problems from storage to processing time.

We addressed this issue with something of a compromise: we did not actually require very high epistasis for the modelling intended. For consistency with our analysis we *did*, however, wish to use variable-length epistasis, which implies that there could potentially be up to L influencing loci, with a storage requirement of 2^L real values. The practical limit to the number of influencing loci as regards storage/processing was found to be about $k = 20$, requiring about 64Mb of storage per table[13]. However, it may be checked that for the κ and L in our long sequence length baseline parameter settings, the probability of more than 20 influencing loci is ≈ 0.005, which was found to be acceptably small; we simply rejected $k > 20$ when assigning epistasis. This was found to be statistically insignificant.

For (pseudo-)random number generation - for assignation of epistasis as well as sampling the underlying distribution for the fitness tables - we used the *Mersenne Twister* generator (Matsumoto & Nishimura, 1998), which combines speed and good statistical properties, as well as having a very long period. Fitness table values were generated from 64-bit random deviates and stored in double-precision (64-bit) floating point format[14].

As regards neutrality, we have found from past experience on a number of computer platforms that comparing floating-point fitness values in order to determine whether two sequences are actually of equal fitness - i.e. in the same neutral network - can often be *unreliable*, due to floating-point arithmetic rounding error. To avoid this problem, we devised an efficient binary "phenotype" for NKp landscapes as follows, based on the "Central Property" (Section 6.2.2) for NKp landscapes: for each feature the phenotypes has a string of bits of length $L+1$. The first (low) bit is set to 1 (resp. 0) according as that feature is contributing (resp. non-contributing). If the feature is contributing, the remaining L bits are filled sequentially (low to high) with the (binary) allele at each of the $(\leq L)$ loci (again read low to high on the sequence bit-string) influencing the given feature and then padded with zeroes; if the feature is non-contributing, the remaining L bits are filled with zeroes. These phenotypes may be safely compared (bit-wise) to decide whether two sequences are of equal fitness.

6.4.3 Neutral Network Statistics

We now turn our attention to some statistical properties of the neutral networks on NKp landscapes, namely *connectivity*, *network size/number* and *percolation/innovation*. All but network size are not easily amenable to mathematical analysis. Connectivity and network size/number[15] distribution, furthermore require exhaustive sampling; we use our short sequence length baseline parameters to compile exhaustive-sampling statistics. Percolation/innovation statistics may be col-

[13] Based on 64 bits per double-precision floating-point number.

[14] This is arguably "overkill" as regards our statistical needs. For instance, even with 32 bit precision, the probability that two sequences evaluate to the same fitness when different fitness table entries are indexed - thus violating the "Central Property" - would be $< 2^{-32}$.

[15] Investigations into network connectivity and number distribution are at the time of writing still preliminary. We hope to present more detailed results at a later stage.

lated by the optimisation-based method described above; we use our long sequence length baseline parameters.

Distribution of network size

To estimate the fitness dependence of neutral network size, a little thought convinces that, since all sequences on a neutral network share the same non-zero fitness contributions, the distribution of *size* of neutral networks for NKp landscapes depends just on the number of contributing features rather than on fitness itself. We thus (for the variable epistasis case) estimate the expected size of the neutral network of an arbitrary sequence given that that sequence has c contributing features.

Consider a sequence chosen uniformly at random from an arbitrary NKp landscape. Let the random variable C be the number of contributing features of our sequence and let the (jointly distributed) random variable S be the size of the (maximal) neutral network of which our chosen sequence is a member. We wish to calculate $\mathbf{E}(S \mid C = c)$.

We condition on the r.v. R representing the number of loci which *do not influence any* of the c contributing features of our sequence: given $R = r$, it is clear that altering the allele at any locus other than these r cannot be neutral, so that the size of our neutral network is consequently $\leq 2^r$. The probability $\mathbf{P}(R = r)$ that there are r such loci is easily seen to be $\binom{L}{r} a^r (1-a)^{L-r}$ where $a = (1-\kappa)^c$. Now the number of neutral mutants of our sequence among the 2^r sequences that may be formed by altering the r loci is precisely the number of sequences with zero contributing features that we would expect to find on an NKp landscape with the same epistasis and neutrality parameters, but with $F - c$ features and sequence length r. From Eq. (6.27) the probability that an arbitrary sequence chosen from an arbitrary NKp landscape with F features is fitness zero (i.e. has zero contributing features) is just p^F. The *expected* number of sequences of fitness zero, where sequence length is r and there are $F - c$ features, may consequently be approximated as $p^{F-c} \cdot 2^r$ (this will not be exact, since the probabilities that different sequences on the *same* landscape are fitness zero are not independent). Thus we derive, summing conditionally over r:

$$\mathbf{E}(S \mid C = c) \approx p^{F-c}[1 + (1-\kappa)^c]^L \tag{6.81}$$

We may now use Eq. (6.59) to approximate the fitness-dependence of expected network size: if W is the fitness of an arbitrary sequence, then:

$$\mathbf{E}(S \mid W = w) \approx \sum_{c=0}^{F} \mathbf{E}(S \mid C = c)\, \gamma(c|w) \tag{6.82}$$

Exhaustive sampling on short sequence length landscapes ($L \leq 20$) indicated that the approximation is reasonably accurate (Fig. 6.6). Fig. 6.7 plots fitness-dependence of (estimated) mean network size $\mathbf{E}(S \mid W = w)$ from Eq. (6.81) and (6.82) against neutrality p (top figure) and epistasis κ (bottom figure) for a range of fitness values w. Except at very low fitness, we see that for small to medium fitness values network size drops off quite slowly with fitness. Beyond a (roughly defined) critical fitness (the "ridge" in the size/fitness plots) network size attenuates roughly exponentially with increasing fitness. For given fitness, network size increases approximately exponentially with increasing neutrality. Scaling of network size with epistasis is somewhat more complex. At low-medium fitness, network size increase roughly exponentially with decreasing epistasis. Beyond

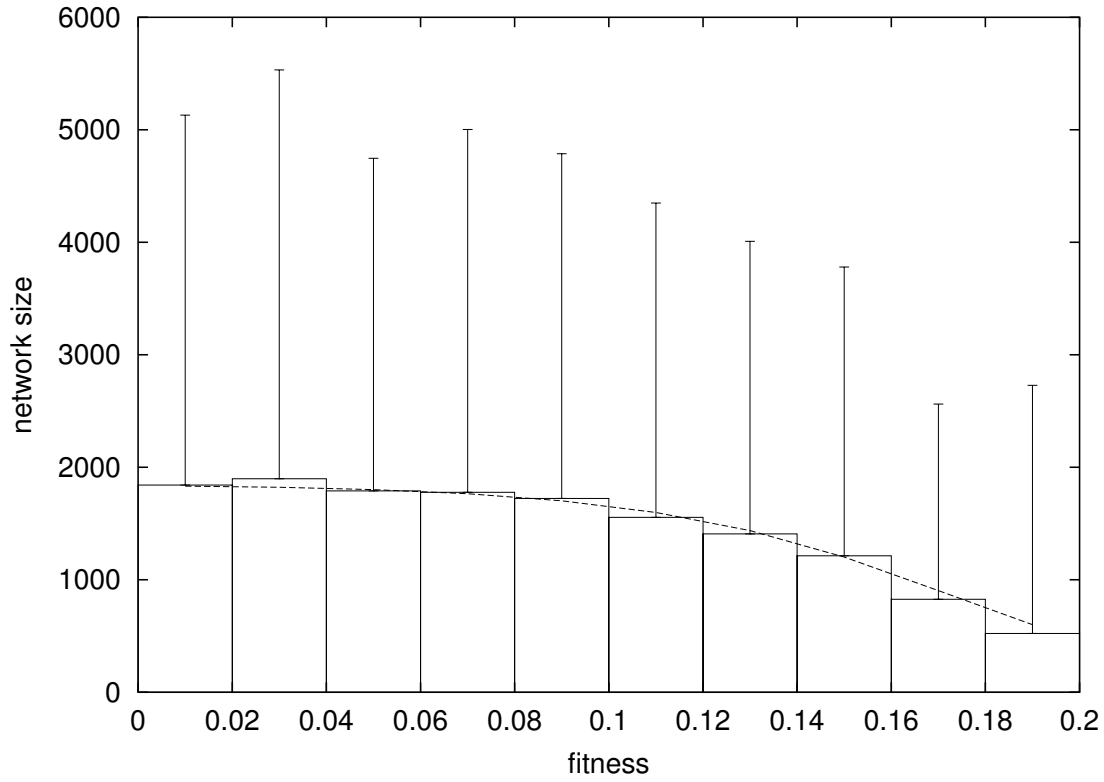

Figure 6.6: Exhaustively sampled NKp mean neutral network size for short sequence length baseline parameters (variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 20$, $N = 16$, $\kappa = 0.375$, $p = 0.99$) plotted against fitness. Number of landscapes sampled $= 10,000$. Error bars indicate 1 standard deviation (note large variance). The dashed line is the analytic (estimated) value $\mathbf{E}(S \mid W = w)$ of Eq. (6.81) and (6.82).

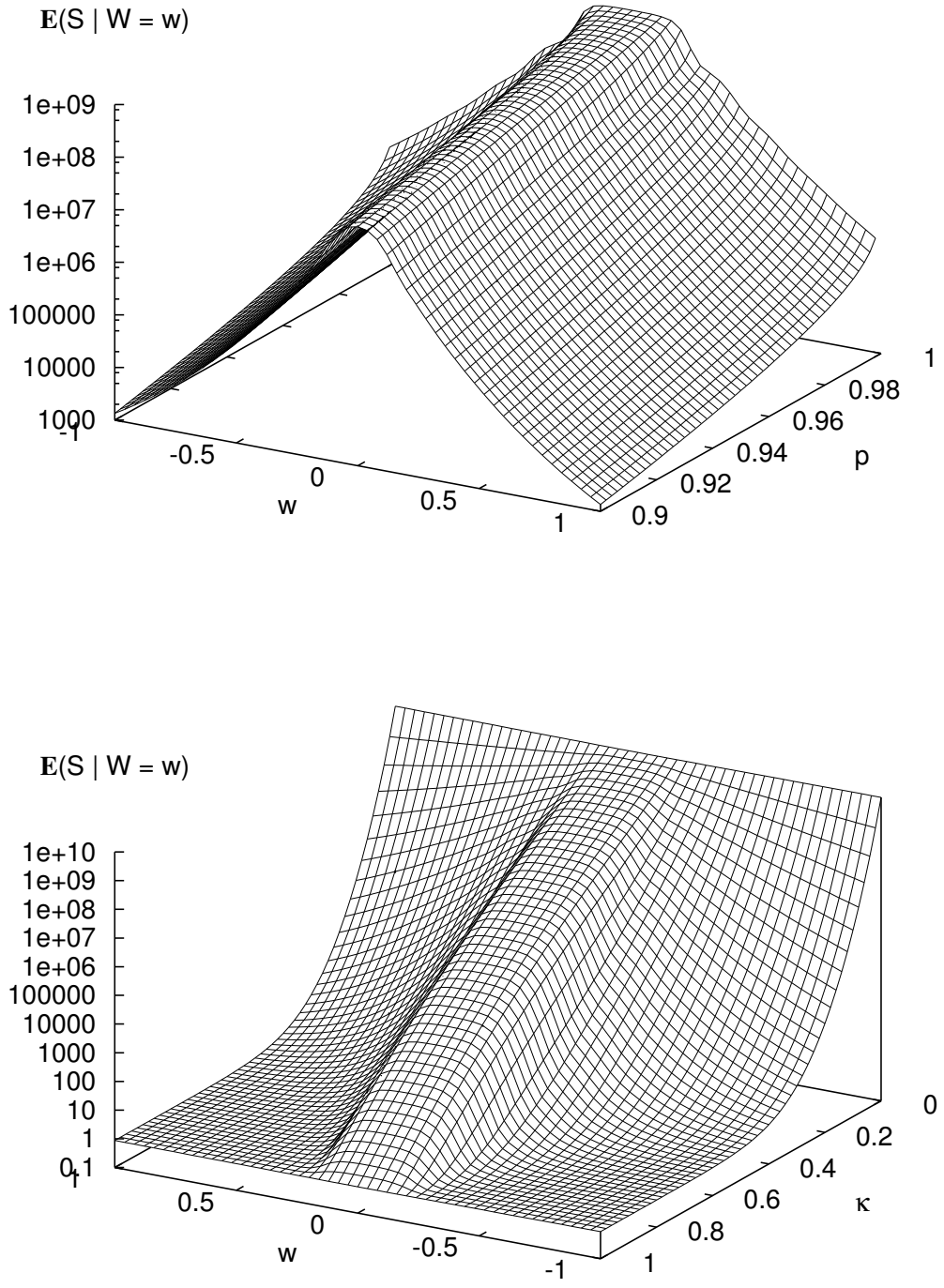

Figure 6.7: Fitness-dependence of (estimated) mean neutral network size $\mathbf{E}(S \mid W = w)$ (note log scale) for NKp landscapes plotted against neutrality (top figure) and epistasis (bottom figure) for a range of w values. Parameters: variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 20$, $N = 32$, $\kappa = 0.125$ (top figure), $p = 0.99$ (bottom figure).

the critical fitness "ridge", network size increases at a roughly exponential-exponential rate with decreasing epistasis[16].

Fitness distribution of number of networks (preliminary results)

We exhaustively sampled short sequence length baseline landscapes to identify neutral networks, binning them according to fitness (we thus approximate the "density" of networks vs. fitness). Results of one such experiment are illustrated in Fig. 6.8 (Fig. 6.8). Further experimentation with

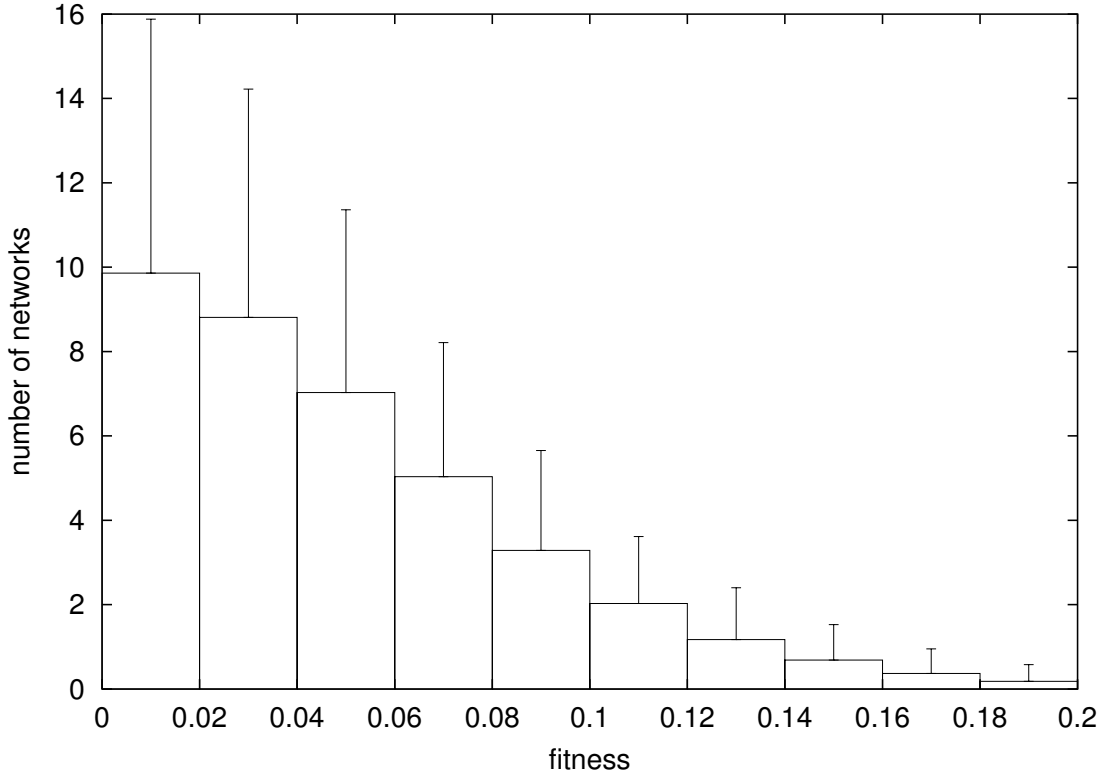

Figure 6.8: Exhaustively sampled NKp mean number of networks (binned vs. fitness) for short sequence length baseline parameters (variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 20$, $N = 16$, $\kappa = 0.375$, $p = 0.99$). Number of landscapes sampled $= 1000$, error bars indicate 1 standard deviation. Mean total number of networks per landscape was $\approx 78.3 \pm 44.1$.

different parameter values (not illustrated) suggested the following:

- Number of networks per fitness band drops off with increasing fitness; the attenuation is approximately linear for small fitness values
- Number of networks increases with decreasing neutrality p
- Number of networks increases with increasing number of features F
- Number of networks increases with increasing epistasis κ

[16] See also (Barnett, 1998) for the (somewhat different) scaling of network size for the fixed-epistasis, uniform fitness distribution case.

Further research is required to elucidate the relationship between number of networks, network size and network fitness.

Network connectivity (preliminary results)

We partitioned short sequence length baseline landscapes into maximal neutral networks (by exhaustive sampling) and then decomposed the maximal networks into connected components with respect to the Hamming graph structure (i.e. 1-bit mutation). It was found that *dis*connected networks were rare: as an example, for one experiment, out of a sample of $10,000$ landscapes - for which 78,299 (maximal) neutral networks were found - only 117 ($\approx 0.15\%$) of the neutral networks were found to be disconnected. Further research is required to elucidate the relationship between network connectivity, size and fitness.

Percolation and Innovation

Simulated annealing as outlined previously was used to collate percolation/innovation statistics (Section 2.2.6) for our long sequence length baseline landscapes. As suggested our sampling technique was also applied to the short sequence length baseline landscapes and results compared with equivalent statistics obtained by exhaustive sampling. Results were in good agreement so that we may hope that sampling bias does not distort results significantly.

The technique was applied as follows: for each landscape of the family sampled, a simulated annealing run was performed. Each time the simulated annealer discovered a higher fitness network (i.e. a new best-so-far fitness of the current simulated annealing run) a blind ant neutral walk (Example 3.2.5, Section 4.1) of a given number of steps was performed on that network using the extant mutation mode/rate[17]. A blind ant neutral walk spends asymptotically equal amounts of time at every sequence of a (connected) network (Section 4.1.1) and would thus be expected to sample the neutral network approximately uniformly. At each sequence along the walk a mutant was created - again according to the extant mutation mode/rate - and its neutral network collated. The networks encountered were then used to compile the percolation index (Eq. 2.49) and cumulative innovation statistics for the current network. Results were binned according to fitness and means and variances of the (binned) samples calculated. As mentioned previously, fitness zero statistics are quoted separately - *the "zero bin" does* not *include fitness zero statistics!* We remark that percolation statistics thus compiled appeared to be very robust with respect to the *length* of the neutral walks.

Fig. 6.9 and Fig. 6.10 demonstrate the fitness-dependence of effective number of accessible networks and cumulative innovations for constant 1-bit and 4-bit mutation respectively, for a sample of $1,000$ long sequence length baseline landscapes over neutral walks of $10,000$ steps. Results support the following (somewhat counter-intuitive) interpretation:

- As we move higher up the landscape in fitness, neutral networks percolate more (in the sense that they have more accessible neighbouring networks) despite the fact that, as we have already seen, they decrease in size. Thus, although we start a neutral walk seeing new (i.e. previously unseen) neighbouring networks at a higher rate, we "run out" of accessible networks sooner than for lower fitness networks.

[17] The mutation rate used by the simulated annealer was not necessarily the same as that used to compile network statistics.

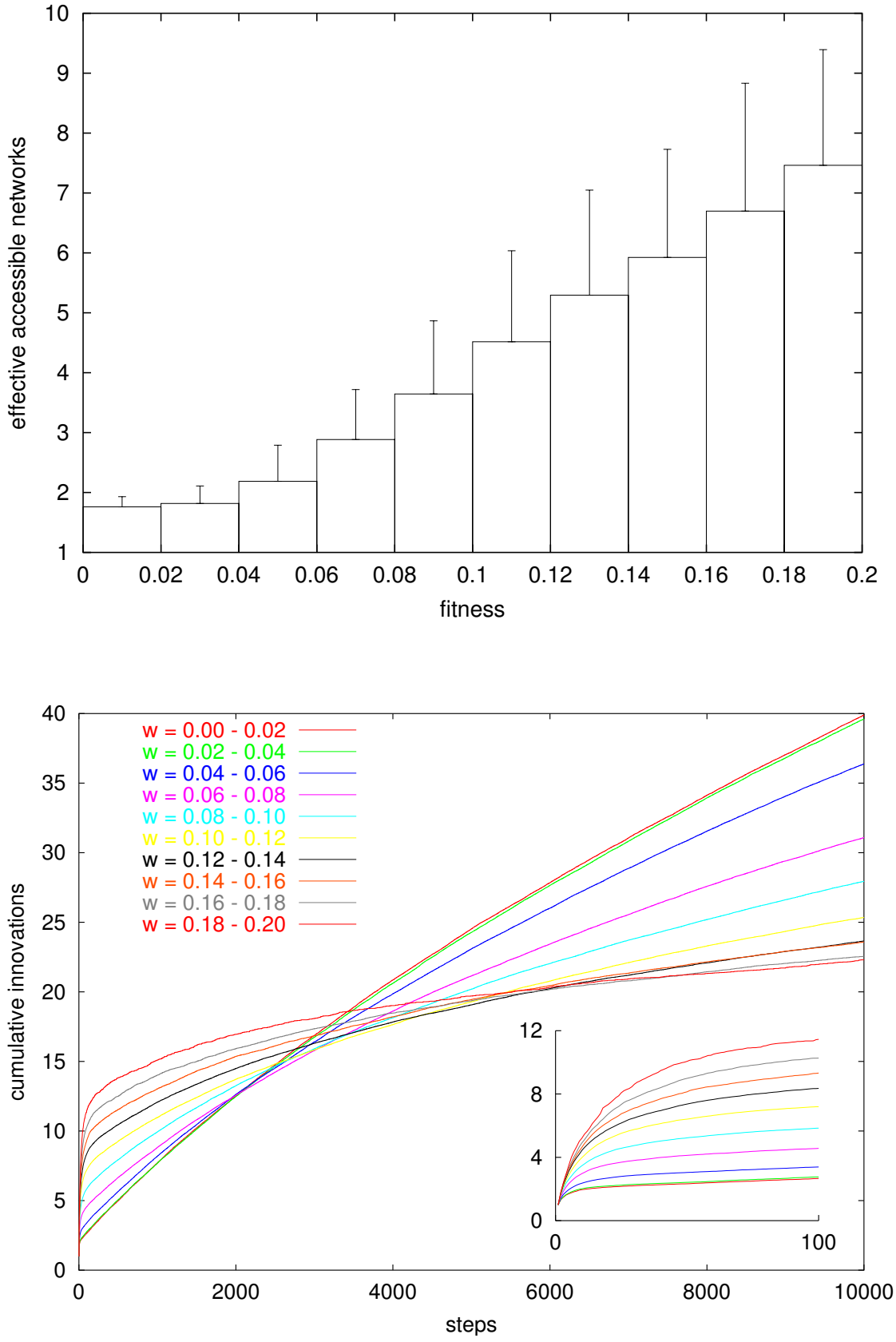

Figure 6.9: *Top:* Mean (sampled) effective accessible networks (percolation index) for constant 1-bit mutation plotted against network fitness. Error bars denote one standard deviation. *Bottom:* Cumulative innovations plotted against neutral walk steps. Inset shows first 100 steps of walk. Parameters: variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 40$, $N = 64$, $\kappa = 0.1875$, $p = 0.999$. Sample size = $1,000$ landscapes, neutral walk steps = $10,000$.

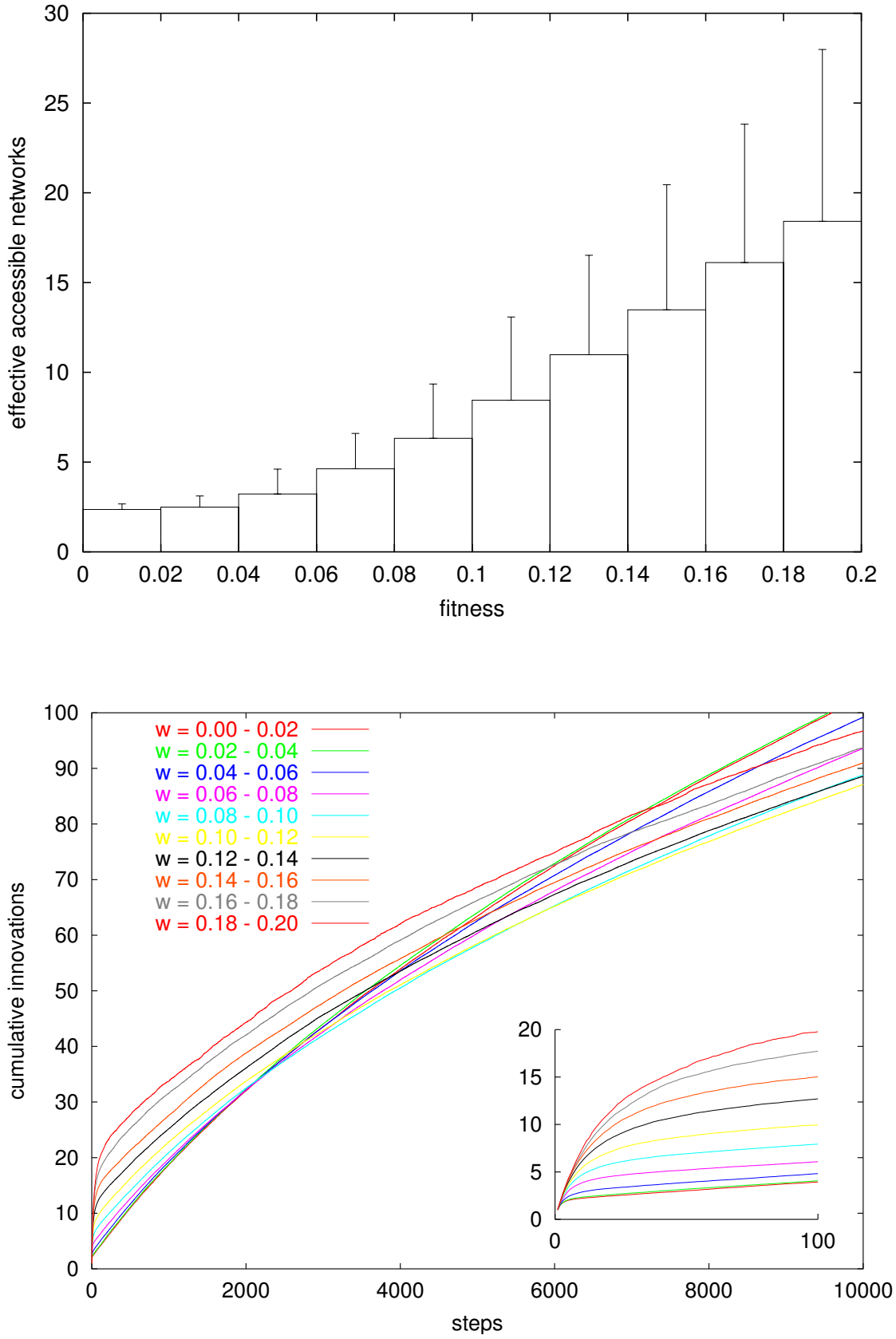

Figure 6.10: *Top:* Mean (sampled) effective accessible networks (percolation index) for constant 4-bit mutation plotted against network fitness. Error bars denote one standard deviation. *Bottom:* Cumulative innovations plotted against neutral walk steps. Inset shows first 100 steps of walk. Parameters: variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 40$, $N = 64$, $\kappa = 0.1875$, $p = 0.999$. Sample size = $1,000$ landscapes, neutral walk steps = $10,000$.

Now we also know that networks *neutrality* decreases as we ascend the landscape. We compiled *neutrally adjusted* percolation statistics for the same parameter values (Eq. 2.53, Section 2.2.6); i.e. neutral mutations are omitted from the entropy calculation. Results showed surprisingly little difference from the comparable un-adjusted statistics. We could interpret this as follows:

- Lower percolation at lower fitness *cannot* be ascribed simply to higher network neutrality.

As might be expected, network percolation also increases with increasing mutation rate. We also compiled statistics for Poisson mutation. Results (omitted) were similar to those for constant mutation.

6.4.4 Hill-climbing on NKp Landscapes

A thorough (empirical) comparative investigation of evolutionary optimisation performance on NKp landscapes - perhaps along the lines of Section 5.4.2 for Royal Road landscapes - is beyond the scope of this study. We did, however, conduct some experiments with *stochastic hill-climbers* (Chapter 3, Example 3.2.5) to test some of our theoretically motivated assertions regarding optimal mutation mode/rate, adaptive netcrawling and the $1/e$ Neutral Mutation Rule. The artificial evolution scenario we attempt to model is the following: we are *not* necessarily attempting to locate a global optimum fitness; rather, much as in the previous Chapter, we take a *time-critical* perspective (*cf.* Section 3.5) and suppose that there is a maximum acceptable number of fitness evaluations T^*, say, beyond which we will terminate any optimisation run. The object of optimisation is then to achieve the highest possible fitness within the T^* available fitness evaluations. Our performance measure is the expected best-so-far fitness achieved in T^* evaluations.

We chose $T^* = 10,000$ evaluations as our benchmark for evolutionary optimisation. At this number of evaluations, the best achieved performances (see below) were in the region of fitness $= 1.8$, which we think to be (see footnote above) approximately 90% of the mean global optimum fitness for the landscape family; we consider this to be a credible scenario for a real-world optimisation problem, in the sense that 90% of maximum achievable fitness may be a reasonably realistic "acceptable" figure for fitness. The hill-climbers tested were a netcrawler (NCR) and stochastic annealing with the "multiplicative" schedule/parameters described in the introduction to this Section (SAM). These were each run with constant (CON) and Poisson (POI) mutation at both a fixed, "hand-optimised" rate (FIX) or at the theoretical optimal rate (OPT) - i.e. the (fitness-dependent) rate which maximises the evolvability $\mathcal{E}(U|w)$ as calculated from Eq. (6.76). The netcrawler was also run in an adaptive mode (ADP) as described in Section 5.4.2 for Royal Road landscapes[18], so as to implement the $1/e$ Neutral Mutation Rule (Prop. 5.3.1). Random search results (RSH) are

[18]Note that a comparable adaptive mutation rate scheme for a simulated annealer would not be practicable, as the annealer will not generally linger long enough on a network to estimate its neutrality reliably, particularly at higher temperatures.

included for comparison. Hill-climber parameters were as follows:

$$
\begin{array}{llllll}
\text{NCR} & \text{CON} & \text{FIX} & \bar{u}=4 & & \\
\text{NCR} & \text{POI} & \text{FIX} & \bar{u}=4.0 & & \\
\text{NCR} & \text{CON} & \text{ADP} & & t_{lag}=350 & \\
\text{NCR} & \text{POI} & \text{ADP} & & t_{lag}=500 & \\
\text{SAM} & \text{CON} & \text{FIX} & \bar{u}=2 & \Delta w=0.006 & \tau_{1/2}=0.40\times T^{*} \\
\text{SAM} & \text{POI} & \text{FIX} & \bar{u}=3.25 & \Delta w=0.007 & \tau_{1/2}=0.35\times T^{*} \\
\text{SAM} & \text{CON} & \text{OPT} & & \Delta w=0.010 & \tau_{1/2}=0.20\times T^{*} \\
\text{SAM} & \text{POI} & \text{OPT} & & \Delta w=0.007 & \tau_{1/2}=0.25\times T^{*}
\end{array}
\tag{6.83}
$$

(note that NCR CON OPT and NCR POI OPT do not have any tunable parameters). We make the following observations on results:

1. NKp landscapes are *hard* to optimise (certainly compared, eg., with Royal Road landscapes; contrast Fig. 6.11 with the comparable Fig. 5.8 of the previous Chapter and note that in the former the time scale is *logarithmic*). This is consonant with our theoretical results on *evolvability* (Section 6.3.3), in particular our observations on the decay of evolvability with increasing fitness.

2. Constant (n-bit) mutation (CON) generally outperforms Poisson mutation (POI), as correctly predicted by Prop. 2.3.1. For constant mutation, furthermore, since (per-sequence) mutation rates are generally low ($\bar{u}\approx 1-3$ for the theoretical optimal (OPT) and adaptive (ADP) mutation regimes at medium-to-high fitness) portals are being discovered close to the current network. This implies that optimisation tends to proceed via the crossing of *entropy* rather than *fitness* barriers as described in Section 3.4.1. Even for quite high fitness, neutral network volumes are large (as may be calculated from Eq. 6.82) , but portals to still higher networks become increasingly sparse and difficult to find. However...

3. ...the simulated annealers generally outperform the netcrawlers[19]. This suggests that there are indeed fitness barriers separating the *highest* neutral networks from *locally optimal* networks, which the annealers, with their capacity to "back down" from a local optimum, are able to escape[20] (note, however, that towards the end of a run, as the annealing temperature approaches zero, the process increasingly - as remarked in Example 3.2.7 - resembles a netcrawler). The behaviour of simulated annealers seems to suggest a "global" structure consistent with observations on local optima on (non-neutral) NK landscapes, as reported eg. in (Kauffman, 1993).

4. Theoretical optimal mutation rates (OPT) - which we recall are based on the *ensemble* statistic $\mathcal{E}(U|w)$ - generally outperform fixed mutation rates (FIX). This suggests that in the parameter regime of the experiment ensemble statistics may be useful, although...

5. ...adaptive mutation rates (ADP) outperform theoretical ensemble optimal rates (OPT) for the netcrawler. Since setting a "true" optimal mutation rate on a per-network basis would be expected to outperform the "per-fitness" ensemble rate (OPT), this provides good evidence for the efficacy of the $1/e$ Neutral Mutation Rule as a heuristic for setting mutation rates.

[19] In fact, in our preliminary investigations, simulated annealing outperformed every GA tested, both with and without recombination.

[20] This conclusion is further supported by results of the author (unpublished) on *multiple independent netcrawlers* on NKp landscapes: over some time scales and population sizes, multiple netcrawlers (with random start) outperform single netcrawlers. Analysis reveals that this is because the multiple netcrawlers may "hedge their bets" - while most will generally start in the "basin of attraction" of a locally sub-optimal network, a few tend to strike it lucky and start from the basin of attraction of a globally optimal, or at least high fitness, network.

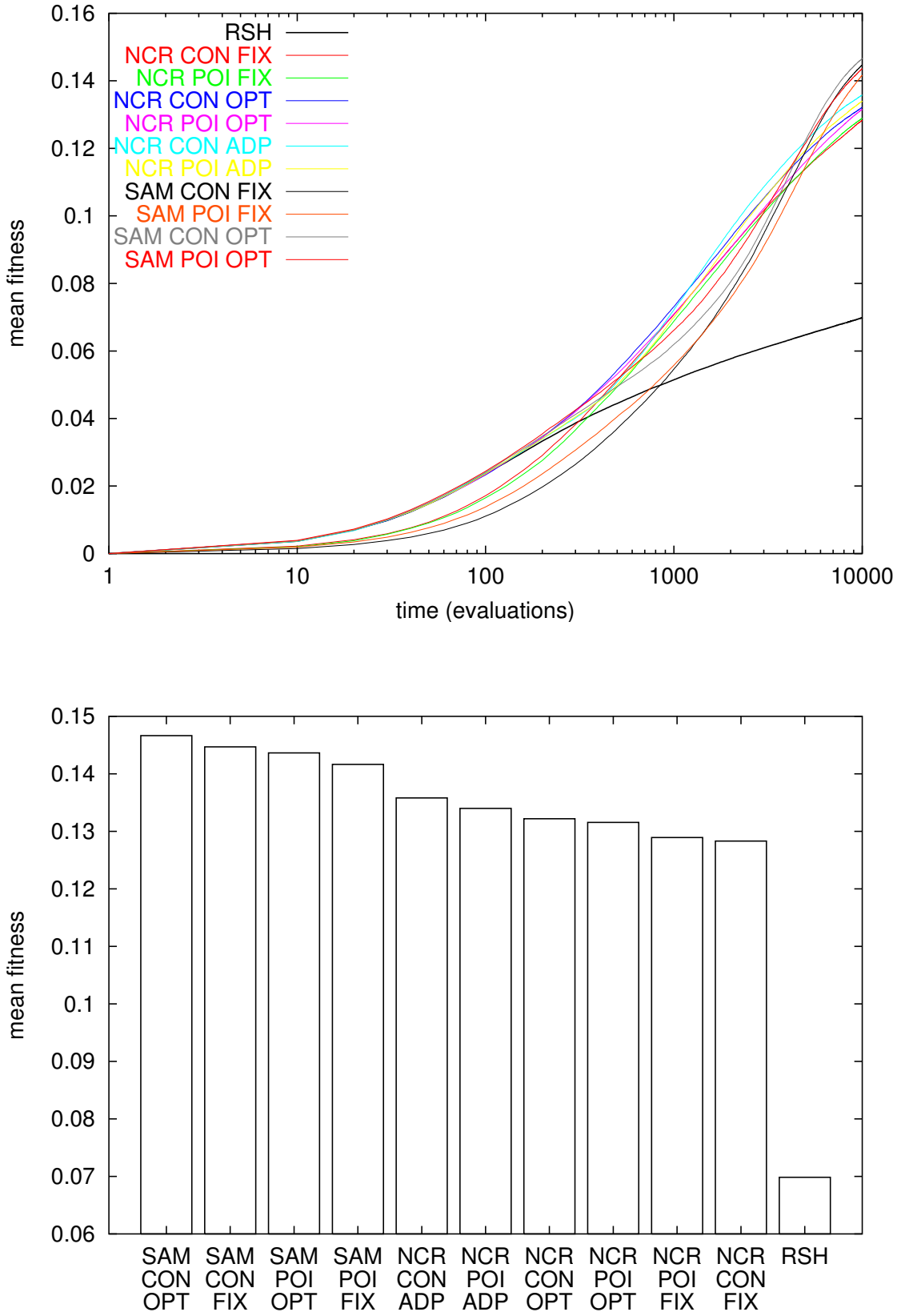

Figure 6.11: *Top:* Optimised hill-climber performance on long sequence length baseline NKp landscapes (variable epistasis, Gaussian underlying distribution with variance $\sigma^2 = 1$, $F = 40$, $N = 64$, $\kappa = 0.1875$, $p = 0.999$): mean best-so-far fitness (sample size $10,000$ runs) plotted against time in fitness evaluations. See text for key and parameters. The bottom figure shows a histogram of mean best-so-far fitness at the end of each run, ranked by performance.

We see in particular that the long sequence length, variable epistasis, Gaussian NKp landscapes of our experiment differ significantly from the ε-correlated landscapes of the previous Chapter. From a practical point of view we also note that the adaptive netcrawlers have a single parameter (the "window" time-lag t_{lag}) which has a very large "sweet spot", making them particularly simple to tune, almost independently of time-scale (*cf.* Chapter 5). Simulated annealing parameters were also comparatively easy to tune, although there was some dependence on run length.

Chapter 7

Recombination

We have previously in this thesis, counter to orthodoxy, expressly rejected recombination as an effective mechanism in evolutionary optimisation. In this Chapter we present some justification for this prejudice; we discuss three reasons why recombination may be ineffective or actually counter-productive. In brief, they are:

- Failure of the Building Block Hypothesis
- Genetic drift ("convergence") and hitch-hiking
- Error thresholds and the "bi-stability barrier"

The Building Block Hypothesis may be regarded as *structural* by nature; it is deeply bound to the coding of an artificial evolution problem. Although it has been quite extensively criticised in the literature (Grefenstette & Baker, 1989; Altenberg, 1994) it nevertheless still appears to underpin (consciously or not) much of the thinking in the GA community; to this author's mind, this may be ascribed, to some extent, to over-reliance on unrealistic model fitness landscapes. The related phenomena of genetic drift and hitch-hiking arise from finite-population stochastic population sampling. Genetic drift - under its *nom-de-guerre* of "(premature) convergence" - is widely perceived as a serious problem for genetic algorithms and generates a wealth of literature (although surprisingly little serious analysis). Its partner in crime, hitch-hiking, although not as widely appreciated as it should be, has been identified and quite thoroughly analysed in the literature (Forrest & Mitchell, 1993; Mitchell et al., 1992). The third phenomenon - the main subject of this Chapter - has not, to the author's knowledge, been identified previously. It makes its presence felt in the (deterministic) *infinite population limit* but is exacerbated by finite population effects.

We do not wish to write-off recombination completely; indeed, GA researchers routinely report improved performance with recombination - although to what extent this may be due to unsuitable evolutionary algorithms or poor choice of parameters (in particular mutation rates) remains moot. It is perhaps worth noting that, while the evolution of sex and recombination remains a highly active (indeed frequently somewhat overheated) topic in evolutionary genetics, few population geneticists would quote similar justification for recombination as the standard GA perspective

might have it; for whatever reasons recombination evolved in the natural world, those reasons are unlikely to be those identified as "useful" from the traditional GA perspective.

7.1 The Building Block Hypothesis

Perhaps at the root of the traditional GA perspective on recombination is the so-called *Building Block Hypothesis* promoted by John Holland and subsequent researchers (Holland, 1992). This identifies genetic "building blocks" with *schemata* - subsets of loci along with specific alleles at those loci. A genotype is said to "instantiate" a schema[1] if the genotype possesses the requisite alleles at corresponding loci. It is then supposed that the power of GA's derives principally from the ability of recombination to assemble short, fitness-enhancing schemata present in different genotypes in a population into new, fit genotypes. The "building block" schemata are assumed short in comparison to the genotype sequence length, so that they are not too frequently disrupted by (say, one- or two-point) crossover (Fig. 7.1).

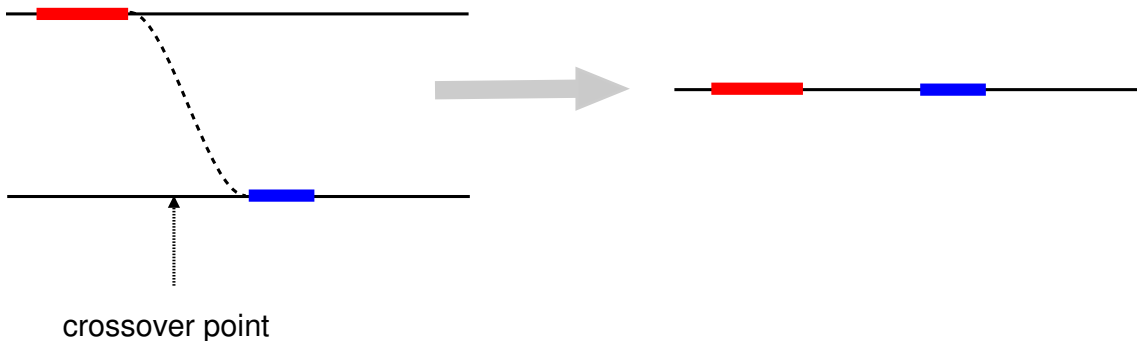

Figure 7.1: The Building Block Hypothesis: recombination (here with single-point crossover) splices two parent genotypes with short, fitness-enhancing schemata, so that both schemata are present in the offspring genotype.

In the next Section we shall question how effective this mechanism is likely to be with regard to the dynamics of population-based evolutionary algorithms. Here, we ask two perhaps more fundamental questions: *Why should we expect fitness-enhancing schemata to recombine successfully?* and: *Why should we expect short, fitness-enhancing schemata to exist at all?* To answer these question we need to examine more carefully what we mean by a "fitness-enhancing" schema. We might be tempted to describe a schema as fitness-enhancing if sequences instantiating that schema are fitter "on average". A schema ξ, say, may be identified with the subset of all those sequences that instantiate it; i.e. we may consider $\xi \subset \mathcal{A}^L$ where $\mathcal{A}^L$ is the sequence space[2]. Thus we might call ξ fitness-enhancing if the mean fitness of sequences instantiating ξ is higher than the mean fitness of an arbitrary sequence; more precisely, if the fitness function for our landscape is $f : \mathcal{A}^L \longrightarrow \mathbf{R}$ we have:

$$\frac{1}{|\xi|}\sum_{x\in\xi} f(x) > \frac{1}{|\mathcal{A}^L|}\sum_{x\in\mathcal{A}^L} f(x) \tag{7.1}$$

[1] Alternatively, the schema is "present" in the genotype.

[2] Note, though, that not every subset of $\mathcal{A}^L$ may be identified with a schema.

This doesn't, of course, imply that $f(x)$ is higher than average for *every* $x \in \xi$. Nor does it follow that there will be *short* schemata satisfying this condition. Now suppose $A \subset \mathcal{A}^L$ is some subset of sequence space. Then (7.1) does *not* imply that sequences in $\xi \cap A$ are on average fitter than an arbitrary sequence drawn from A. This is due to *epistasis*: the effect on fitness of a particular allele at a particular locus may depend on the alleles at other loci. The effect on fitness of a schema is in this sense *context-dependent*. This observation engenders two major implications:

1. *A is the set of sequences represented in an evolving population.* We have stressed previously (Section 1.1.3), that uniform sampling is likely to be highly unrepresentative of sequences sampled by an evolutionary process; even if a schema is fitness-enhancing "on average" it need not be fitness-enhancing for sequences in an evolving population.

2. *The set A is that defined by another schema.* The implication is that even if a schema enhances fitness on average for arbitrary sequences, it need not enhance fitness for sequences that instantiate some *other* schema.

The first point might conceivably be remedied by re-defining "fitness-enhancing" to mean "fitter than average within the context of a (given) population". The second point seems more difficult to address; it implies in particular that we cannot assume in general that splicing sequences instantiating fitness-enhancing schemata will yield above-average fitness sequences.

Of course, there are landscapes where we may readily identify fitness-enhancing (in some sense of "above average fitness") schemata which manifestly *do* recombine successfully; perhaps the best-known example would be the Royal Road landscapes (Chapter 5), which were designed specifically with this in mind (Mitchell et al., 1992); here, set blocks may be considered as eminently splice-able building blocks. The above objections do not apply to such landscapes (although those of the next Section do).

A related problem with the Building Block Hypothesis is that recombination can only assemble *disjoint* schemata. For the Royal Road landscapes this is not a problem, since the "good" (i.e. short, fitness-enhancing) schemata are in fact disjoint. In contrast, consider the NKp landscapes of Chapter 6: given some feature, we may may pick out the loci influencing that feature and a set of alleles for those loci which reference a high-fitness contribution; this yields a candidate fitness-enhancing schema. We note that for the "random epistasis" models presented in Chapter 6 these schemata are not particularly likely to be short; other epistatic assignment schemes such as "adjacent neighbourhood" (Kauffman, 1993) might be expected to yield shorter schemata. But (even for nearest-neighbour models) if we examined these "natural" schemata for an *actual* high-fitness sequence, we would be likely to find, particularly at higher epistasis, that they *overlap* to a large degree and thus could not have been spliced together by crossover. Epistasis in the NKp model dictates that "good" schemata are not generally good in the context of other "good" schemata - they frustrate each other. In the terminology of Gavrilets' *holey landscapes* (Gavrilets & Gravner, 1997), on an epistatic landscape recombination lands us in the holes. (Altenberg, 1995) addresses similar concerns. He uses Price's Covariance and Selection Theorem (Price, 1970) to derive a version of Holland's Schema Theorem (Holland, 1992) which goes some way to identifying "good" schemata (with respect to a particular recombination operator - essentially the short, disjoint schemata for one or two point crossover) and, by incorporating the fitness distribution of

recombinant offspring into the analysis, quantifies the extent to which a GA is liable to exploit them successfully:

> "[The variant Schema Theorem] ... makes explicit the intuition about how schema processing can provide a GA with good performance, namely: (1) that the recombination operator determines which schemata are being recombined; and (2) that there needs to be a correlation between [disjoint] schemata of high fitness and the fitness distributions of their recombinant offspring in order for the GA to increase the chance of sampling fitter individuals."

We turn next to some real-world artificial evolutionary problems and ask whether we might expect to find combinable building blocks. Consider the thought experiment (the "Fitness Landscaper") of the introductory Chapter: here the problem to be solved is the design of a software controller for a robot required to perform, say, a navigational task. The chosen implementation is a highly recursive neural network, with the connections, weights and timescale parameters encoded in a genotype after some "natural" fashion. Now we might expect that the fitness function is "modular" in the sense that high fitness is conferred by success at various behaviours ("move towards light", "turn away from obstacle", "move in an arc", *etc.*). Yet, if we looked for a causal origin of these behaviours in the neural network - that is, if we attempted to map modularity at the fitness level to structural modularity in the neural network - we would be likely to fail. The reason for this is that highly interconnected recursive neural networks are notoriously *synergistic* in their functioning; every part effects every other (recursively!) and, though a network may indeed appear to exhibit discrete behaviours, these behaviours cannot in general be localised, say, to *sub-networks*. Now it may be that the genotype → neural network mapping is "modular", insofar as codings for sub-networks may be localised on the genotype; notwithstanding, there will still be no modularity (or localisation on the genotype) in the mapping from genotype to *behaviour* (much in the same way as fitness contributions in the random epistasis NKp model are not in general localised on the genotype). Where then are we to find building blocks - fitness-enhancing schemata - on the genotype? And even if such building blocks existed, network synergy would be likely to induce extreme context-dependence; we would not expect schemata to recombine successfully, since (like schemata in the NKp model) they would likely interact to their mutual detriment. Note that this is not an argument against *correlation*; neural network landscapes may well be quite highly correlated (they may also exhibit substantial *neutrality*). Similar synergistic "phenotypes", with concomitant lack of a modular genotype → fitness mapping, may also be found in other real-world evolution problems, such as evolving electronic circuits, either on-chip (Thompson, 1996; Thompson, 1998; Harvey & Thompson, 1996) or in software emulation (Layzell, 2001).

In summary, for a class of real-world optimisation problems - those featuring what we might term *synergistic phenotypes* - we can expect the Building Block Hypothesis to fail because: (1) it is not clear that suitable building blocks will exist and (2) even if they do exist, they are unlikely to recombine successfully. We might contrast this with that favourite GA "benchmark", the Travelling Salesman Problem. Here, (depending on the coding) we may well have a modular genotype → fitness mapping; *sub-tours* may be fitness-enhancing, may be coded for locally on the genotype and may well recombine successfully. Note that, nonetheless, TSP landscapes are in general quite epistatic (Stadler, 1996) (as indeed are Royal Road landscapes) - correlation in itself does not tell us much about whether preconditions for the Building Block Hypothesis to apply will prevail.

7.2 Genetic Drift and Hitch-hiking

Genetic drift is the phenomenon whereby stochastic sampling reduces *genetic variation* within a finite (fixed size) population. It is most easily apprehended in the absence of mutation - i.e. in the absence of any mechanism for producing new alleles at a locus. Suppose that at a given locus, locus *i*, say, and for a given allele *a* at locus *i* there are, say, *n* sequences in the population with allele *a* at locus *i*. After some number *t* of generations, there is a *finite* probability that, through failure (via sampling fluctuations) of such sequences to be selected, all *n* of these sequences have disappeared from the population; then, since there is no mechanism for regenerating *a* at locus *i*, allele *a* is irrevocably lost from the population at locus *i*. This extends to all alleles at all loci. The result is that after a sufficient number of generations, there is a finite probability that *all sequences in the population are copies of a single sequence!* - all genetic variation has disappeared.

Crucially, recombination does not affect this conclusion: recombination can only "mix-and-match" alleles at a given locus. Mutation will counteract this effect by generating new alleles, but at low mutation rates (and traditionally mutation has been relegated to the status of "background operator", implying low mutation rates) there will still be a pronounced loss of variation; the population will be *localised* in sequence space, somewhat like a classical *quasi-species* (*cf.* the next Section). If, now, a GA relies on recombination as the principal search mechanism, this loss is catastrophic; recombination requires genetic variation to produce novel sequences. The term *premature convergence* (Goldberg, 1989) has been used to describe the effect whereby variation is lost - and crossover rendered ineffective - before appreciable fitness levels have been achieved. From the traditional GA perspective, premature convergence has been perceived as perhaps *the* most important problem for the GA designer to overcome, and a remarkably large percentage of the GA literature is devoted to schemes (crowding, nicheing, fitness sharing and spatial distribution to name but a few) to reduce the effect.

Hitch-hiking (Mitchell et al., 1992)may be viewed as an exacerbation of the effects of genetic drift by strong selection: if a new sequence with a strong selective advantage is discovered (by whatever mechanism), then that sequence (and its neutral offspring) will be strongly selected at the expense of other sequences. Within a short "takeover" time, *all* sequences in the population will be descendants of the new sequence and its neutral variants, with a concomitant drastic loss of genetic variation. If, in particular, there were useful "building blocks" present in sequences *other* than the new fit sequence, these building blocks will not survive into post-takeover populations. Conversely, "bad" building blocks in the new fit variant will "hitch-hike" along into future generations; it is then up to mutation to regenerate variation. Now the takeover time for a sequence with a strong selective advantage tends to be orders of magnitude shorter than the times between discovery of fitter variants (Nimwegen et al., 1997). The result is that, even if good, potentially recombine-able building blocks exist for a GA implementation, *good building blocks will rarely be present simultaneously in different sequences in an evolving population*. Recombination thus is not afforded the opportunity to assemble good building blocks.

7.3 Recombination, Error Thresholds and the Bi-stability Barrier

In the previous two Sections, we have argued that recombination is likely to be ineffective in the sense of the Building Block Hypothesis; that is, at discovering fit sequences by recombining building blocks. But, we should perhaps ask, might not recombination be useful for *creating* good building blocks? We are assuming that our fitness landscapes are to some degree correlated; this provides a rationale for supposing that *mutation* might discover high-fitness sequences (*cf.* Section 2.3.4). Might recombination, then, have a similar value? We observe that recombination is a "contracting" operator; the Hamming distance between a recombinant offspring and either of its parents is always less than or equal to the Hamming distance between the parents themselves. Thus the offspring of sequences nearby in sequence space do not stray too far from their parents - but in this case mutation could achieve the same effect. If, on the other hand, parent sequences are *distant* in sequence space (and, as we have seen in the previous Section, localisation of the population in sequence space implies that this is in fact unlikely to be the case much of the time), there does not seem to be any reason to suppose that recombination should "respect" the (mutational) correlation structure; more probable that recombination then merely act as a *macromutation* - an uncorrelated "jump into the void". Indeed, Gavrilets' "holey landscape" theory (Gavrilets & Gravner, 1997) posits this as a mechanism for sympatric speciation.

One particular phenomenon associated with recombination has been identified by population geneticists: that recombination has the ability, in the right circumstances, to aid evolution by reducing the accumulation of deleterious mutations; more specifically, it may reduce the "genetic load" - the reduction in population mean fitness engendered by the cumulative effect of mutation (Kondrashov, 1982; Kimura & Maruyama, 1966; Charlesworth, 1990; Higgs, 1994) (in some sense recombination acts here as a kind of "error repair" mechanism). A consequence is that a population evolving with recombination can bear a higher mutation rate without "destabilising" than in the absence of recombination. It seems reasonable that this may be an advantage in artificial evolution for the following reason: we have previously seen (Section 2.3.4) that setting optimal mutation rates involves a "balancing act" between mutating away from a current (sub-)optimal network but not mutating too far off. Now a potential problem, particularly if the selective advantage of the current optimum is small, is that if the mutation rate is too high (and "too high" might be just what we require for efficient search!), the population may be unable to maintain its "foothold" on the current optimum in the face of the information-degrading entropy of mutation. At this point the population may slip to a lower optimum, or even wander at random in sequence space (Bonhoeffer & Stadler, 1993; Schuster & Stadler, 1994). This phenomenon, known as the (mutational) *error threshold* is addressed by Manfred Eigen's theory of the *molecular quasi-species*, which analyses information-processing in populations of self-replicating biomolecules (Eigen, 1971; Eigen et al., 1989; Swetina & Schuster, 1982). Although quasi-species theory is formulated strictly in the infinite population-size limit, the effects of error thresholds may actually be be amplified by genetic drift in finite populations (Nowak & Schuster, 1989).

The property of a fitness landscape for which recombination can be demonstrated to be advantageous in the above sense is known as *synergistic epistasis* (Kondrashov, 1982; Kimura & Maruyama, 1966) - roughly, that around a fitness peak, fitness drops off at a greater-than-exponential rate with respect to Hamming distance from the peak. Now it is not clear (there is

much scope for research here) why one might expect fitness landscapes in artificial evolution to demonstrate synergistic epistasis (or not) in the neighbourhood of fitness (sub-)optima. It seems at least worthwhile to investigate the situation regarding recombination when the *opposite* of synergistic epistasis obtains; that is, when fitness drop off at a *lower*-than-exponential rate with increased Hamming distance from a fitness peak. This is what we do in this Section. We investigate the infinite-population dynamics (under the quasi-species formalism) of a population on the simplest non-synergistic landscape, the single-spike landscape. This research was inspired by a recent study of recombination in a retro-viral quasi-species (Boerlijst, Bonhoeffer, & Nowak, 1996). There, the dynamics were found to exhibit *bi-stability*, with a stable and an unstable equilibrium. We find a similar phenomenon in our research. We show that the stable and unstable equilibria coalesce at an error threshold which represents a first order phase transition, in contrast to the classical mutational error threshold which is a second order (discontinuous) phase transition. We derive analytical expressions for the equilibria and error threshold and analyse the stability of the equilibria. Implications of results for artificial evolution are discussed; in particular, we argue that the unstable equilibrium represents a ***bi-stability barrier*** to the fixation of a newly discovered fit sequence.

We note that the landscape is, of course, unrealistic with respect to any serious optimisation problem; in particular, we do not take neutrality into account. We do note, however, that although neutrality was not explicitly included in Eigen's pioneering work, the formalism is certainly flexible enough to take neutrality into account; error thresholds (for mutation only) have indeed been analysed for landscapes with neutrality (Reidys & Stadler, 2001) and some experimental research has been performed where recombination is present (Ochoa & Harvey, 1999; Ochoa, Harvey, & Buxton, 1999). The author hopes to extend the current analysis to the neutral case in the near future; it would not appear to involve particular difficulties.

7.3.1 The Quasi-species Model

Manfred Eigen, in his quasi-species formalism (Eigen, 1971; Eigen & Schuster, 1979; Eigen et al., 1989), developed an approach to analysing the evolution of large populations of information-encoding sequences based on (deterministic) flow-reactor kinetics, whereby concentrations of sequence types change according to differential rates of replication, destruction and, via mutation, transformation to different types. This formalism led to the concept of a *quasi-species* as a distribution of sequences localised in sequence space and clustered around the most frequent sequence variety. The existence of an *error threshold* of mutation (or replication fidelity) was established, beyond which the fittest sequence type would inevitably be lost from the population. The implication is that if the mutation rate is too high a favourable mutant can never become established in an evolving population. Furthermore, the error threshold typically decreases with increasing sequence length, so that there is effectively (for a given per-allele mutation rate) a limit to the sequence length beyond which an evolving population of sequences can maintain sufficiently high fitness to be viable. This observation leads to the so-called "error catastrophe"; in nature, the genomes of organisms have comparatively low effective per-allele mutation rates due to the existence of *error correction* mechanisms in effect during replication. However, these error correction mechanisms must themselves be coded for in the organism's genome - they are functionally

non-trivial and are likely to require lengthy coding sequences, greater than the "raw" per-allele mutation rate would permit due to the error threshold. How, then, could these error correction mechanisms (and hence complex organisms requiring long genomes) have evolved?

There has been a persistent and recurrent idea that an answer to this conundrum may lie with sex and recombination (Kimura & Maruyama, 1966; Maynard Smith, 1978; Kondrashov, 1982; Charlesworth, 1990). Thus it has been suggested that, under certain circumstances, recombination can act as a kind of error repair mechanism. It is, therefore, of great interest to examine the effects of recombination on the dynamics of a quasi-species and on error thresholds in particular. In attempting to extend the "classical" quasi-species formalism to include recombination we immediately come up against three problems. The first is that in the asexual case analysis of the quasi-species dynamics is greatly abetted by the (near) *linearity* of the system; recombination introduces a quadratic non-linearity. Secondly, in the asexual case (and particularly if sequence lengths are long) we are generally entitled to ignore "back-mutation" of less fit sequences to the fittest sequence type. This simplifies the analysis considerably, enabling us to separate out the dynamics of the concentration of the fittest sequence variety. When recombination is present we may still neglect back-mutation, but we *cannot* ignore "back-recombination" (this is in a sense the essence of the error-correction potential of recombination) so that the dynamics of the fittest sequence type are inextricably linked to the concentrations of types nearby in sequence space. Thirdly, the equations are complicated by the presence of *linkage disequilibrium* (Maynard Smith, 1998; Crow & Kimura, 1970), where the particular alleles to be found on a sequence at "linked" loci cannot be assumed independent. Our approach then is to develop approximations that reflect, at least qualitatively, the dynamics of the sexual quasi-species.

The basic quasi-species model employed in this Section is as follows: we consider a large (effectively infinite) population of binary sequences of fixed length L evolving under selection, mutation and recombination. There is a single "optimal" sequence[3] and the fitness of any sequence depends only on the number of errors; i.e. the Hamming distance of that sequence from the optimal sequence. We shall be interested mainly in the *long sequence length limit* $L \to \infty$; all analytical results are strictly valid only in this limit. Numerical simulations are of necessity performed with finite sequence length, although care was taken to use the longest sequence lengths compatible with clarity and feasible within the constraints of computational resources. In what follows (unless otherwise stated) all Latin indices $i, j, \ldots$, Greek indices $\alpha, \beta, \ldots$ and summations run from 0 to L, where L may be ∞.

Let w_i denote the fitness of a sequence with i errors. We now specialise to a "spike" fitness landscape defined by:

$$w_i = 1 + \delta_{i0}\sigma = \begin{cases} 1+\sigma & \text{if } i = 0 \\ 1 & \text{if } i > 0 \end{cases} \tag{7.2}$$

where $\sigma > 0$ is the *selection coefficient*[4] of the optimum sequence. As previously noted, while this fitness landscape arguably lacks relevance to any realistic artificial evolutionary landscape - although it might be argued that fitness spikes surrounded by selectively neutral "plateaux" may

[3] Also known in the literature as the "wild-type" or "master sequence".

[4] It is commonplace in the population genetics literature to take the optimum fitness as 1 and that of other sequences as $1-s$. Since we shall only consider fitness-proportional selection, there is no essential difference; σ and s are related by $1+\sigma = \frac{1}{1-s}$.

be *local* features of more complex landscapes (and we already know that evolving populations are generally localised in sequence space) - it has the advantage of simplicity and allows for direct comparison with known results from asexual quasi-species theory.

The set of all sequences in the landscape with exactly i errors is known as the i-th *error class*; note that it defines a (non-maximal) neutral partitioning of the landscape. We use $x_i(t)$ to denote the proportion (or *concentration*) of sequences with i errors at generation t, so that $\sum_i x_i(t) = 1$. $(x_i)_{i=1,\ldots,L}$ represents the *quasi-species distribution* of the population[5]. We will use the generating functions $g_t(z)$ for the $x_i(t)$ defined by:

$$g_t(z) \equiv \sum_k x_k(t)(1-z)^k \tag{7.3}$$

Note that $g_t(0) = 1$ and (by convention) $g_t(1) = x_0$. We also define:

$$\theta(t) \equiv \sum_k k x_k(t) \tag{7.4}$$

the *mean number of errors per sequence*. In terms of the generating functions $g_t(z)$ we have:

$$\theta(t) = -g_t'(0) \tag{7.5}$$

where the prime denotes differentiation with respect to z. If the concentrations $x_i(t)$ are time-independent we drop the argument t.

The remainder of this Section is organised as follows: 7.3.2 reviews the pertinent features of the model in the absence of recombination. 7.3.3 introduces recombination to the model while 7.3.4 presents the approximations used to analyse the sexual quasi-species. 7.3.5 addresses stability issues while 7.3.6 discusses some implications of results to optimisation.

7.3.2 The Asexual quasi-species

We suppose that evolution of the quasi-species operates as follows: generations are non-overlapping. At each generation sequences are selected for reproduction proportional to their fitness. Each allele of a selected sequence then mutates (i.e. the binary allele flips) independently with probability $0 < u < \frac{1}{2}$. We also set $U \equiv Lu =$ mean number of mutations per sequence; we thus have multinomial fitness-proportional selection[6] (Section 3.2.1) with Poisson mutation at per-sequence rate U. Note that the maximum entropy approximation Eq. (3.18) with respect to our mutation operator holds *exactly* for the error class partitioning. We then have[7]:

$$x_i(t+1) = \frac{1}{W(t)} \sum_j m_{ij} w_j x_j(t) \tag{7.6}$$

where $\boldsymbol{m}$ is the mutation matrix:

$$m_{ij} \equiv \mathbf{P}\,(\text{a sequence with } j \text{ errors mutates to a sequence with } i \text{ errors}) \tag{7.7}$$

[5]Note that notation in this Section differs slightly from that in previous Chapters; in particular, the Roman "t" denotes *generations* rather than fitness evaluations and per-sequence mutation rate is written as U rather than $\bar{u}$.

[6]Selection is thus not elitist (Section 3.2.1). While we might expect the results of this Section to be qualitatively similar for other selection schemes, our results will patently *not* apply for selection with elitism.

[7]We have, essentially, a choice between examining concentrations *before* or *after* mutation; for convenience, we choose the latter.

and $W(t)$ is simply the *population mean fitness*:

$$W(t) \equiv \sum_k w_k x_k(t) = \sigma x_0(t) + 1 = \sigma g_t(1) + 1 \tag{7.8}$$

Equation (7.6) may be viewed as defining a (discrete) L-dimensional dynamical system. A straightforward calculation gives, for the mutation probabilities m_{ij}:

$$m_{ij} = \sum_{\alpha,\beta} \delta_{i,j-\alpha+\beta} \binom{j}{\alpha} \binom{L-j}{\beta} u^{\alpha+\beta}(1-u)^{L-(\alpha+\beta)} \tag{7.9}$$

In terms of the generating function $g_t(z)$ we note the following: if (x_i) is the quasi-species distribution at a given generation and $g(z)$ its generating function (7.3) then selection transforms $g(z)$ according to:

$$g(z) \mapsto \frac{1}{W}\left[\sigma x_0 + g(z)\right] = \frac{\sigma g(1) + g(z)}{\sigma g(1) + 1} \tag{7.10}$$

In the long sequence length limit $L \to \infty$ the action of mutation on the generating function is (see Appendix B.1):

$$g(z) \mapsto e^{-Uz} g(z) \tag{7.11}$$

Note that it follows that in the long sequence length limit $m_{ij} = 0$ for $i < j$; i.e. back-mutation becomes negligible. We may write (7.6) in terms of the generating function as:

$$g_{t+1}(z) = e^{-Uz} \frac{\sigma g_t(1) + g_t(z)}{\sigma g_t(1) + 1} \tag{7.12}$$

If the population is in dynamic equilibrium, $x_i(t) = x_i$ for all i and t, then (7.12) becomes:

$$g(z) = e^{-Uz} \frac{\sigma g(1) + g(z)}{\sigma g(1) + 1} \tag{7.13}$$

which may be solved directly for $g(z)$. We find in particular, setting $z = 1$, that the optimum sequence concentration is given by either $x_0 = g(1) = 0$ or:

$$x_0 = g(1) = \frac{1}{\sigma}\left[e^{-U}(\sigma + 1) - 1)\right] \tag{7.14}$$

Now x_0 must be non-negative. From examination of Eq. (7.14) we see that, given a selection coefficient σ, there can only be an equilibrium solution with a non-vanishing concentration of the optimum sequence if U is less than a certain critical value U_a given by:

$$U_a = \log_e(1 + \sigma) \tag{7.15}$$

This critical mutation rate has been termed the *error threshold*. The behaviour of the model is illustrated in Figs. 7.2 and 7.3. In Fig. 7.2 the optimum sequence concentration $x_0(t)$ as calculated from (7.6) is plotted against time for $U < U_a$. We see that there is a single stable equilibrium. As the mutation rate is increased to the critical rate U_a the equilibrium approaches zero discontinuously. Beyond U_a the ≈ 0 equilibrium corresponds to a "delocalised" population that "sees" only a selectively neutral landscape. In Fig. 7.3 equilibrium optimum sequence concentrations are plotted against per-sequence mutation rate for a few selection coefficients. The transition in the equilibrium behaviour of the quasi-species as the parameter U crosses the error threshold U_a is of a form that would be recognised by physicists as a *second order (discontinuous) phase transition*.

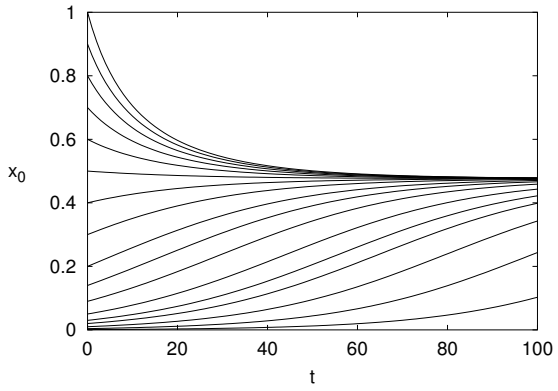

Figure 7.2: Sequence concentrations $x_0(t)$ for the asexual quasi-species (7.6) plotted against time. Sequence length $L = 20$, selection coefficient $\sigma = 0.1$, per-sequence mutation rate $U = 0.05$. We note that for this value of σ, $U_a \approx 0.0953$.

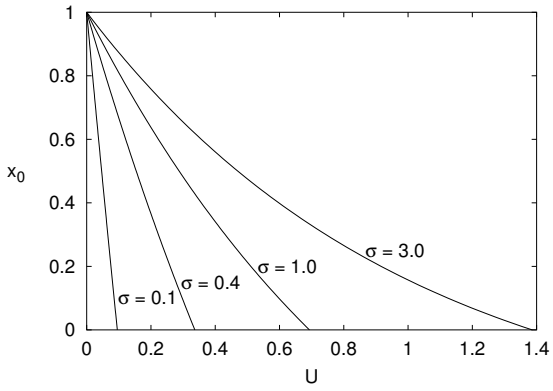

Figure 7.3: Equilibria of (7.6) plotted against per-sequence mutation rate.

7.3.3 The Sexual quasi-species

We now introduce recombination to our quasi-species model as follows: at each generation sequences are selected for reproduction proportional to their fitness. Selected sequences pair off at random; each pair produces an offspring with *uniform crossover* (Syswerda, 1989) - i.e. each allele in the offspring sequence is chosen independently from one of its two parents with probability $\frac{1}{2}$. Each allele of the offspring then mutates as before. This model is similar to the model of retro-virus replication with super-infection presented in (Boerlijst et al., 1996).

To calculate the evolution of the quasi-species distribution we need the probability that recombination of a sequence from, say, error class k with one from error class l produce a sequence in error class j. In contrast to mutation we cannot strictly do this from the quasi-species distribution alone; recombination probabilities will depend on the *particular* sequences chosen from the respective error classes. We thus make a maximum entropy-like assumption: namely, that the frequency distribution of sequences *within* each error class is (approximately) uniform. Under this approximation, Eq. (7.6) becomes:

$$x_i(t+1) = \frac{1}{W(t)^2} \sum_{j,k,l} m_{ij} r_{jkl} w_k w_l x_k(t) x_l(t) \tag{7.16}$$

where:

$$r_{jkl} \equiv \mathbf{P}\left(\begin{array}{c} \text{the offspring of a sequence with } k \text{ errors recombined} \\ \text{with a sequence with } l \text{ errors has } j \text{ errors} \end{array} \right) \tag{7.17}$$

(the tensor $\boldsymbol{r} = (r_{jkl})$ represent the analogue for recombination of the mutation matrix). Our approximation then gives:

$$r_{jkl} = \sum_{\alpha} \binom{k}{\alpha} \binom{L-k}{l-\alpha} \binom{L}{l}^{-1} \binom{k+l-2\alpha}{j-\alpha} \left(\frac{1}{2}\right)^{k+l-2\alpha} \tag{7.18}$$

(note that this is actually symmetric in k, l). How well, then, is our maximum entropy assumption likely to hold? Firstly, it is clear that *linkage disequilibrium* will violate uniformity. Now it is

well-known (Crow & Kimura, 1970; Maynard Smith, 1998) [Maynard Smith] that we can expect to find linkage disequilibrium where there is strong selection with epistasis, and in small finite populations. Linkage disequilibrium is *destroyed*, on the other hand, by mutation and recombination. Some thought indicates that the same factors ought to affect uniformity of sequence distribution within error classes. In our scenario there is no selective differential between sequences within an error class, so the only factor mitigating against our assumption is likely to be finite population drift. We might thus expect our approximation to hold up as long as population size and mutation rate are not too small. Experiment bears this out: we performed Monte Carlo simulations of the full (finite population, stochastic) quasi-species dynamics for populations in the range of $100 - 10,000$ sequences. Results (not shown) indicated that even at quite low mutation rates (and particularly for long sequence lengths) the uniform distribution assumption holds up reasonably well and that in particular, the infinite-population model (as specified by Eqs. 7.16 and 7.18) provides a good approximation to the full dynamics (but see also Section 7.3.5 below). We also remark that (again, particularly for long sequence lengths) experiments with one- and multi-point crossover indicate that the recombination *mode* appears not to be very significant to the qualitative (and indeed quantitative) dynamics.

Analogous to Eq. (7.11), in the long sequence length limit $L \to \infty$ the action of recombination on the generating function (7.3) is given by (see Appendix B.2):

$$g(z) \mapsto g\left(\tfrac{1}{2}z\right)^2 \tag{7.19}$$

Note that in deriving this limit we assume that the number of errors is $\ll$ the sequence length L. We may then write (7.16) in terms of the generating function as:

$$g_{t+1}(z) = e^{-Uz}\left(\frac{\sigma g_t(1) + g_t\left(\frac{1}{2}z\right)}{\sigma g_t(1) + 1}\right)^2 \tag{7.20}$$

At equilibrium (7.20) becomes:

$$g(z) = e^{-Uz}\left(\frac{\sigma g(1) + g\left(\frac{1}{2}z\right)}{\sigma g(1) + 1}\right)^2 \tag{7.21}$$

Unlike (7.13) we cannot solve this equation explicitly for $g(z)$ or indeed for $x_0 = g(1)$. We can, however, simulate (7.16) numerically; some results are illustrated in Fig. 7.4. Here the optimum sequence concentration $x_0(t)$ as calculated from (7.16) is plotted against time. For the initial conditions binomial quasi-species distributions were chosen (see Section 7.3.4 below for justification). We see that at the lower mutation rate the dynamical system (7.16) apparently has a stable equilibrium (at $x_0 \approx 0.6$) and an unstable equilibrium (at $x_0 \approx 0.1$). There is also a stable equilibrium at $x_0 \approx 0$ which again corresponds to a delocalised neutrally drifting population. At the higher mutation rate only the delocalised equilibrium remains. At a critical per-sequence mutation rate U_s between these values the system bifurcates (Baake & Wiehe, 1997), the unstable and stable equilibria coalescing and vanishing. We identify this critical mutation rate as an error threshold, since beyond this value the population inevitably delocalises; a physicist would describe the transition as a *first order (continuous) phase transition*.

7.3.4 Approximations for the Sexual quasi-species

Simulation of the sexual quasi-species model indicates that, due to the "shuffling" effect of recombination, the quasi-species distribution rapidly attains (from any initial conditions) a distribution

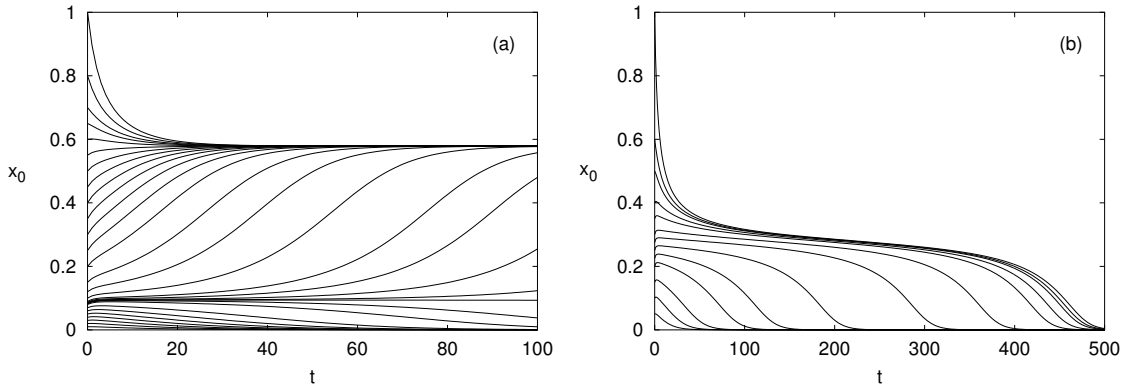

Figure 7.4: Sequence concentrations $x_0(t)$ for the sexual quasi-species (7.16) plotted against time. Sequence length $L = 20$, selection coefficient $\sigma = 0.4$ and (a) per-sequence mutation rate $U = 0.11$, (b) $U = 0.15$.

close to a *binomial distribution*, which, in the long sequence length limit approaches a *Poisson* distribution. We thus proceed as follows: taking at generation t the Poisson distribution:

$$x_k(t) = e^{-\theta(t)} \frac{\theta(t)^k}{k!} \tag{7.22}$$

with generating function:

$$g_t(z) = e^{-\theta(t)z} \tag{7.23}$$

the evolutionary equation (7.16) yields for the next generation a distribution which will be "nearly Poisson". We approximate this distribution by another Poisson distribution, choosing $\theta(t+1)$ judiciously. This we shall do in two ways, according as the selection coefficient σ is small or large; in either case we effectively reduce the evolution of the quasi-species from an L-dimensional to a 1-dimensional dynamical system.

Small-σ Approximation

If σ is small, the evolution of the quasi-species from one generation to the next was found empirically to be dominated by the mean number of errors $\theta(t)$. For the long sequence length limit we thus choose $\theta(t+1)$ to be the mean number of errors one generation on, starting with a Poisson distribution (7.22) at generation t. Substituting $g_t(z)$ from (7.23) in the right hand side of (7.20) then using the relation (7.5) we find immediately:

$$\theta(t+1) = U + \frac{\theta(t)}{\sigma e^{-\theta(t)} + 1} \tag{7.24}$$

The equilibrium condition $\theta(t) = \theta(t+1) = \ldots = \theta$ yields, after re-arranging terms:

$$e^{-\theta} = \frac{U}{\sigma} \frac{1}{\theta - U} \tag{7.25}$$

which may be solved numerically for $x_0 = e^{-\theta}$. Equation (7.25) is observed to have two solutions for U smaller than a threshold value $\widehat{U}_s$ which approximates the error threshold U_s of the exact model (7.16) for small σ.

We can calculate the approximate error threshold $\widehat{U}_s$ as follows: the two solutions for θ of (7.25) correspond to the points where the curves $f(\theta) = e^{\theta}$ and $g(\theta) = \dfrac{\sigma}{U}(\theta - U)$ intersect. At the

approximate error threshold $U = \widehat{U}_s$ these curves are tangential; i.e. $f(\theta) = g(\theta)$ and $f'(\theta) = g'(\theta)$. Solving these equations we find that $\widehat{U}_s$ is the (unique) solution of:

$$Ue^{U+1} = \sigma \tag{7.26}$$

which may be solved numerically for $\widehat{U}_s$ in terms of σ. We note that for small σ, $\widehat{U}_s$ is of the same order as σ and we have:

$$\widehat{U}_s = \frac{\sigma}{e} + \boldsymbol{O}\left(\sigma^2\right) \tag{7.27}$$

This may be compared with $U_a = \sigma + \boldsymbol{O}\left(\sigma^2\right)$ for the asexual case (7.15). It is also not difficult to show that at the error threshold:

$$x_0 = \frac{1}{e} + \boldsymbol{O}\left(\sigma\right) \tag{7.28}$$

Large-σ Approximation

If σ is large, the evolution of the quasi-species was found to be dominated by the optimum sequence concentration $x_0(t)$. We proceed as for the small-σ case, except that we now choose $\theta(t+1)$ such that $x_0(t+1) = e^{-\theta(t+1)}$ is the optimum sequence concentration in the next generation, again starting with the Poisson distribution (7.22) at generation t. Substituting $g_t(z)$ from (7.23) in the right hand side of (7.20), setting $z = 1$ and noting that $x_0(t) = e^{-\theta(t)}$ we find:

$$x_0(t+1) = e^{-U}\left(\frac{\sigma x_0(t) + \sqrt{x_0(t)}}{\sigma x_0(t) + 1}\right)^2 \tag{7.29}$$

At equilibrium, $x_0(t) = x_0(t+1) = \ldots = x_0$, we find (assuming $x_0 > 0$ and taking square roots of both sides):

$$\sigma x_0 + 1 = e^{-\frac{1}{2}U}(\sigma\sqrt{x_0} + 1) \tag{7.30}$$

This is a quadratic equation for $\sqrt{x_0}$ which may be solved explicitly, yielding two values for x_0 so long as U is less than a critical value $\widetilde{U}_s$ which approximates the error threshold U_s of the exact model (7.16) for large σ. $\widetilde{U}_s$ is easily found to be:

$$\widetilde{U}_s = -2\log_e\left(\frac{2}{\sigma}(\sqrt{1+\sigma} - 1)\right) \tag{7.31}$$

For large σ we see that $\widetilde{U}_s$ scales as:

$$\widetilde{U}_s = \log_e\frac{\sigma}{4} + \boldsymbol{O}\left(\sigma^{-\frac{1}{2}}\right) \tag{7.32}$$

so that $U_a - \widetilde{U}_s = \log_e 4 + \boldsymbol{O}\left(\sigma^{-\frac{1}{2}}\right) \approx 1.3863$ for large σ. We also find that at the error threshold:

$$x_0 = \frac{1}{\sigma^2}(\sigma - 2\sqrt{1+\sigma}) \tag{7.33}$$

which, for large σ, scales as:

$$x_0 = \frac{1}{\sigma} + \boldsymbol{O}\left(\sigma^{-\frac{3}{2}}\right) \tag{7.34}$$

In Fig. 7.5 we plot optimum sequences concentration x_0 for the equilibria of (7.16) with $L = 60$, against per-sequence mutation rate U for several values of the selection coefficient σ. The small- and large-σ approximations (7.25), (7.30) for x_0 are plotted on the same graph. In this figure the upper branches of the curves represent the stable and the lower branches the unstable equilibria. It

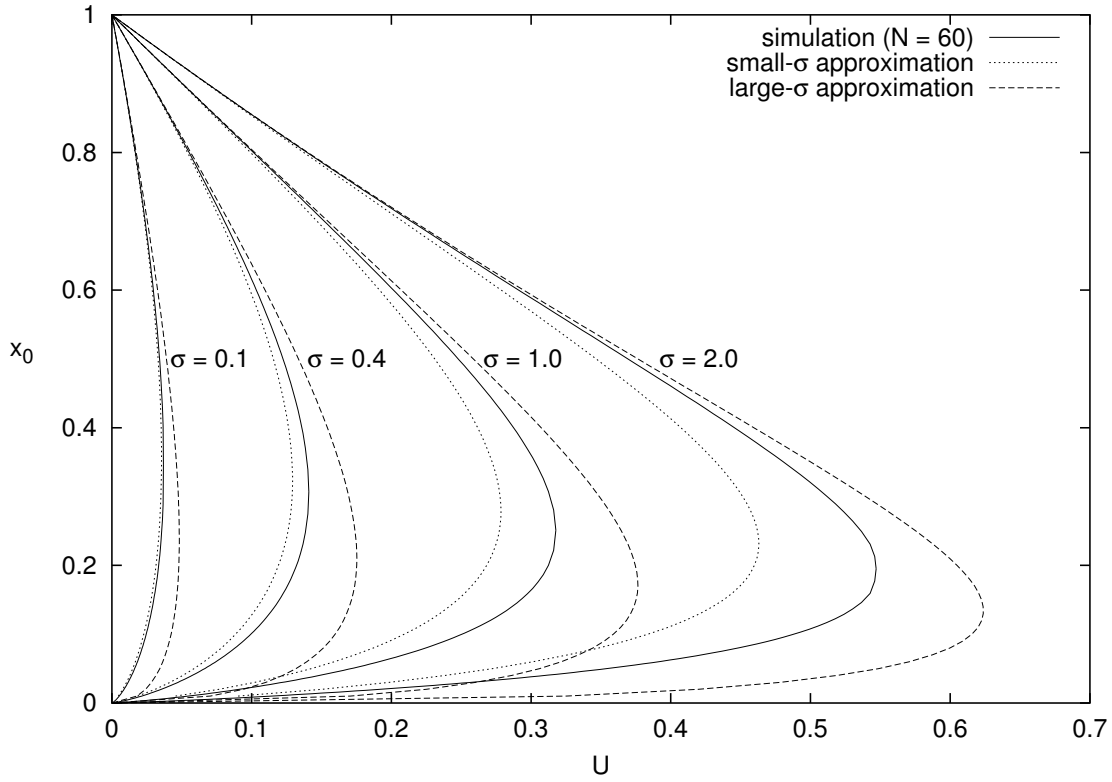

Figure 7.5: Equilibria of (7.16) and approximations (7.25), (7.30) plotted against per-sequence mutation rate.

was also found that for any σ, U the optimum sequence concentration x_0 at equilibrium is always smaller with recombination than without.

Fig. 7.6 plots the error threshold U_s computed from numerical simulation of (7.16) with sequence length $L = 80$ as well as the small- and large-σ approximations $\widehat{U}_s$ and $\widetilde{U}_s$ against σ. The asexual error threshold U_a is also plotted for comparison.

7.3.5 Stability of Equilibria

We wish to investigate the *stability* of the equilibrium solutions to (7.16). This is of particular importance to analysis of finite-population models for which (7.16) may be an approximation, since stochastic fluctuations will occur in the concentrations $x_i(t)$ which might destabilise a deterministic equilibrium. Furthermore, we note that, particularly for small σ, the system may persist in a state apparently close to the unstable equilibrium for a considerable time before destabilising (Fig. 7.7); we should like to elucidate the mechanism by which these "nearly stable" quasi-species destabilise.

Consider a discrete dynamical system:

$$\boldsymbol{x}(t+1) = \boldsymbol{F}(\boldsymbol{x}(t)) \tag{7.35}$$

where $\boldsymbol{x}$ is a real vector (x_i) and $\boldsymbol{F}(\boldsymbol{x})$ a (smooth) vector-valued function with component functions $F_i(\boldsymbol{x})$. Suppose further that ξ is a *fixed-point* of (7.35); i.e.:

$$\xi = \boldsymbol{F}(\xi) \tag{7.36}$$

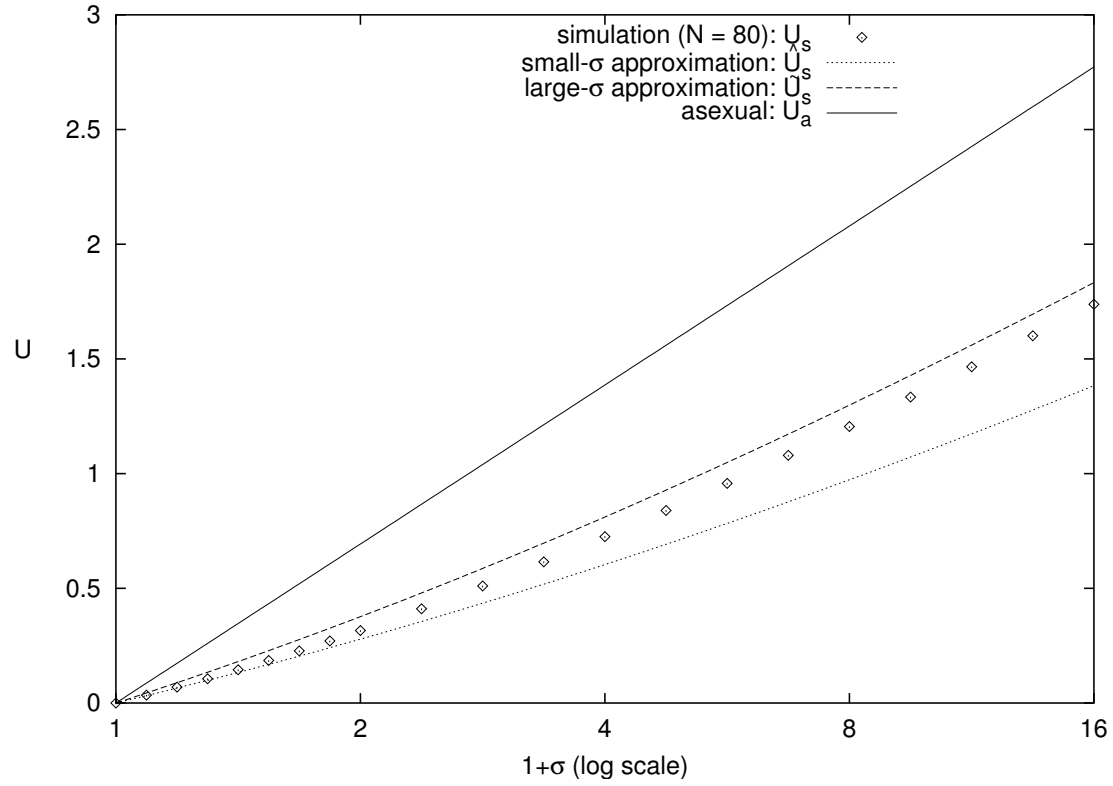

Figure 7.6: Error thresholds U_s, $\widehat{U}_s$, $\widetilde{U}_s$ and U_a plotted against $1+\sigma$ (note logarithmic scale).

Suppose now that at time t, $\boldsymbol{x}(t)$ is close to ξ; i.e. $\delta \equiv |\boldsymbol{x}(t)-\xi|$ is small. We find then from (7.35) and (7.36) that:

$$\boldsymbol{x}(t+1)-\xi = \nabla \boldsymbol{F}(\xi)\cdot(\boldsymbol{x}(t)-\xi)+\boldsymbol{o}(\delta) \tag{7.37}$$

where $\nabla \boldsymbol{F}(\xi)$ is the matrix with components $\left.\dfrac{\partial F_i}{\partial x_j}\right|_{\boldsymbol{x}=\xi}$ and (7.37) is the *linearisation* of the dynamical system (7.35) about the fixed-point ξ. It represents the linear transformation mapping points in the vicinity of a fixed-point to their positions in the next generation. Now the *principal eigenvalue* of a linear transformation indicates the degree of "stretching" in the direction of greatest stretching; a fixed-point of a dynamical system (7.35) will be stable iff $|\lambda_0|<1$ where λ_0 is the principal eigenvalue of $\nabla \boldsymbol{F}$ at that fixed-point. Our evolutionary equations (7.16) are of the form (7.35) with $\boldsymbol{F}$ given by:

$$F_i(\boldsymbol{x}) = \frac{1}{W(\boldsymbol{x})^2}\sum_{j,k,l} m_{ij} r_{jkl} w_k w_l x_k x_l \tag{7.38}$$

with the added constraint $\sum_i x_i = 1$. We find that at a fixed-point ξ:

$$[\nabla \boldsymbol{F}(\xi)]_{ij} = \frac{2w_j}{W(\xi)}\left\{-\xi_i + \frac{1}{W(\xi)}\sum_{k,l} m_{il} r_{ljk} w_k \xi_k\right\} \tag{7.39}$$

To analyse the linear transformation $\nabla \boldsymbol{F}(\xi)$ given by (7.39) we calculated its eigenvalues $\lambda_0 > \lambda_1 > \lambda_2 > \ldots > \lambda_L = 0$ for the stable and unstable equilibria[8]. Fig. 7.8 plots the principal eigenvalues λ_0 for a range of mutation rates and a few σ values, for the stable (lower branches) and unstable

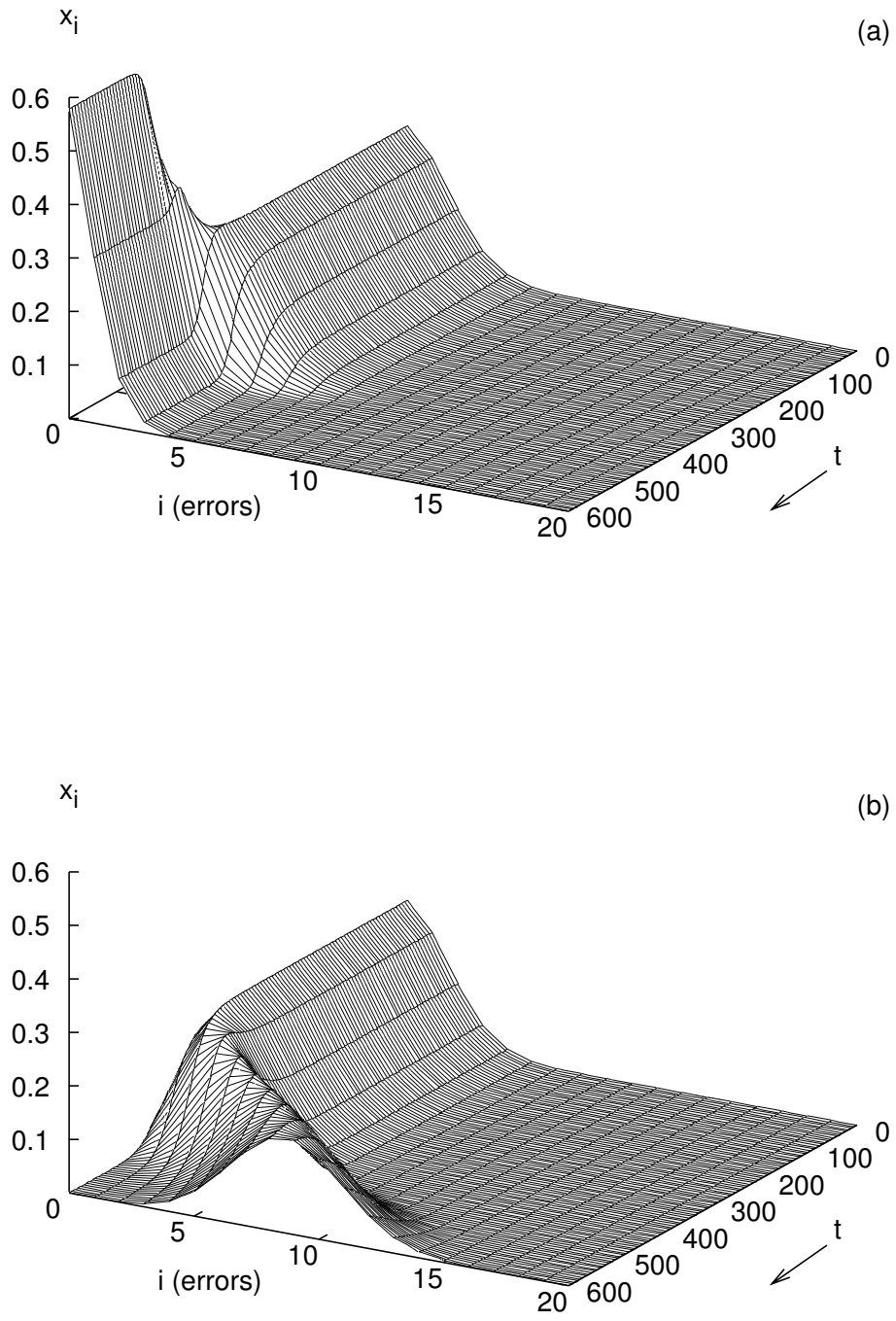

Figure 7.7: Behaviour of the sexual quasi-species near the unstable equilibrium. In both cases $L = 20$, $\sigma = 0.4$, $U = 0.11$ and the population was initialised with a binomial distribution (a) just above and (b) just below the unstable equilibrium.

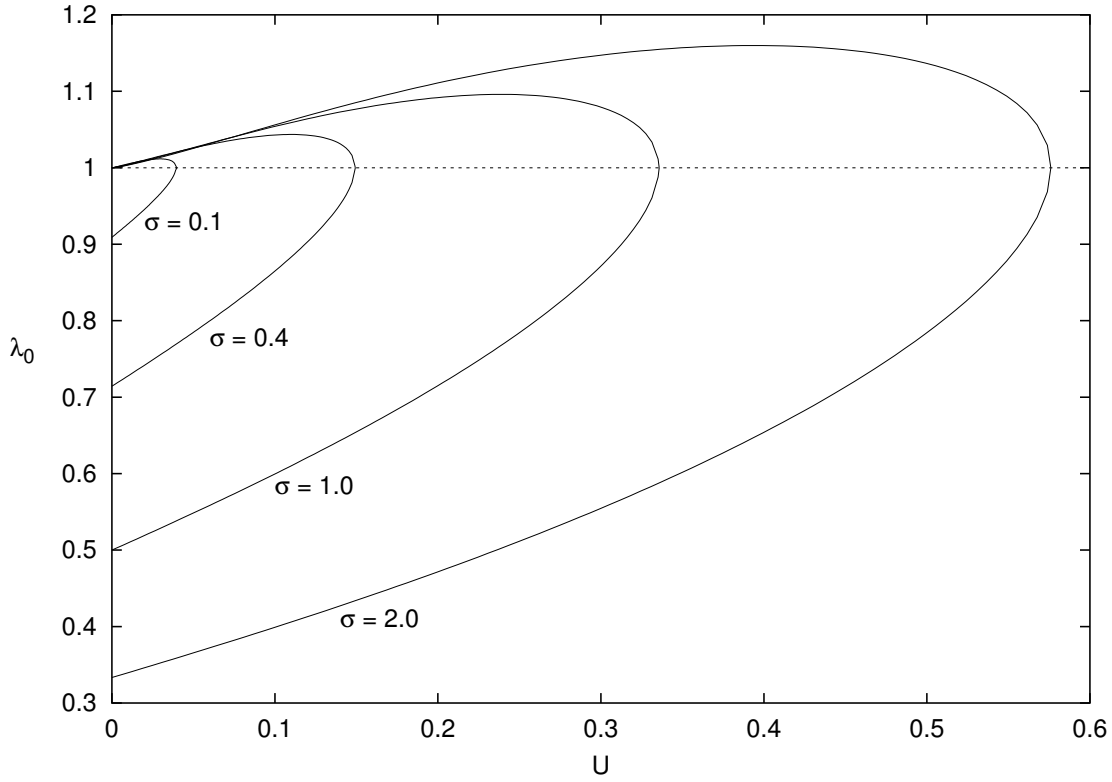

Figure 7.8: Principal eigenvalues of $\nabla \boldsymbol{F}(\xi)$ for the stable (lower branches) and unstable (upper branches) equilibria of (7.16). Sequence length $L = 60$.

(upper branches) equilibria. It was also found empirically that the remaining eigenvalues fall off roughly exponentially; i.e. for fixed σ and U there is a constant $c \approx \frac{1}{2}$ such that for $k = 1, 2, \ldots$ we have $\lambda_k \approx c^k \lambda_0$. It was certainly the case that for the stable equilibrium $|\lambda_0| < 1$ (confirming stability) while for the unstable equilibrium $|\lambda_0| > 1$ (confirming instability) and that in both cases $|\lambda_k| < 1$ for $k > 0$. This latter implies in particular that the unstable equilibrium of (7.16) is only unstable along a single dimension - we might think of it as a narrow steep-walled saddle-shaped gully with a shallow curvature in the direction of the principal eigenvector of $\nabla \boldsymbol{F}(\xi)$. For small σ (see Fig. 7.8 and analysis below) we see that λ_0 is only slightly larger than 1. This explains the comparative stability of the unstable equilibrium (Fig. 7.7). It is also interesting to note that for a given selection coefficient σ there is a critical mutation rate at which the instability of the unstable equilibrium is greatest. For higher mutation rates the unstable equilibrium becomes less unstable as the error threshold is approached.

To approximate the principal eigenvalues, we proceed as follows: in Section 7.3.4 we approximated the L-dimensional system (7.16) by the 1-dimensional systems (7.24) and (7.29). Consider the general situation where there is a vector function $\phi(y) = (\phi_i(y))$ of a new variable y and a scalar function $f(y)$ satisfying the relation $\phi(f(y)) = \boldsymbol{F}(\phi(y)) \; \forall \, y$ or, in functional notation:

$$\phi \circ f = \boldsymbol{F} \circ \phi \tag{7.40}$$

[8] It was found (although not proven analytically) that all eigenvalues were non-negative. We note that $\sum_i x_i = 1$ implies that $\nabla \boldsymbol{F}(\xi)$ is a *projection*, so that there must exist at least one zero eigenvalue.

This equation[9] formalises the notion of "reducing the dimension" of the dynamical system (7.35) to the new 1-dimensional dynamical system $y(t+1) = f(y(t))$. We then have:

$$\phi_i'(f(y))f'(y) = \sum_j F_{i,j}(\phi(y))\phi_j'(y) \quad \forall\, y \tag{7.41}$$

where primes denote differentiation, so that if η is a fixed-point of f then $f'(\eta)$ is an eigenvalue of $\nabla \boldsymbol{F}(\xi)$ for $\xi = \phi(\eta)$, with eigenvector $\phi'(\eta)$.

The small-σ approximation of 7.3.4 is an approximation to just such a reduction of dimension (in the sense that the relation (7.40) is "almost" satisfied) if we identify y with θ. $\phi(\theta)$ is then specified by (7.22) and $f(\theta)$ by (7.24). The eigenvalue $\hat{\lambda}_0 \equiv f'(\theta)$ at the stable (resp. unstable) fixed-point θ is found to be:

$$\hat{\lambda}_0 = (1+U)\left(1 - \frac{U}{\theta}\right) \tag{7.42}$$

where θ represents the stable (resp. unstable) solution of the equilibrium equation (7.25).

For the large-σ approximation of 7.3.4 we identify y with x_0; $\phi(x_0)$ is then specified by (7.22) and $f(x_0)$ by (7.29). The eigenvalue $\tilde{\lambda}_0 \equiv f'(x_0)$ at the stable (resp. unstable) fixed-point x_0 is found to be:

$$\tilde{\lambda}_0 = \frac{2 - e^{-\frac{1}{2}U}}{\sigma x_0 + 1} \tag{7.43}$$

where x_0 represents the stable (resp. unstable) solution of the equilibrium equation (7.30).

Numerical computation of $\hat{\lambda}_0$ and $\tilde{\lambda}_0$ showed them to be reasonable approximations to the principal eigenvalue λ_0 of $\nabla \boldsymbol{F}(\xi)$ (for both stable and unstable equilibria) for small and large values of σ respectively. We may also conclude that for small σ the unstable equilibrium is most sensitive to perturbations of θ, the mean number of errors per sequence, while for large σ it is more sensitive to perturbations of x_0.

Finally, we return to our remark in Section 7.3.3 that the infinite-population model (7.16 and 7.18) is generally a good approximation to the corresponding finite-population (stochastic) model. This is not entirely true near the unstable equilibrium; unsurprisingly stochastic fluctuations will tend to dislodge the population from the vicinity of the unstable equilibrium, whence the population will either converge to the stable (upper) equilibrium, or errors will accumulate and the population delocalise (Fig. 7.9).

7.3.6 Discussion

Comparing the behaviour of the sexual with the asexual quasi-species there are several striking differences. In particular it seems clear that on the spike fitness landscape recombination is a distinct disadvantage for the following principal reasons:

- For the same sequence length and mutation rate, the error threshold is lower with recombination than without.
- Suppose that in a *finite* population our optimum sequence has been recently discovered by mutation/recombination. Even if any copies of the new optimum sequence survived elimination by random drift, the concentration of the new sequence would have to *drift*

[9] In mathematical "Category Theory" Equation (7.40) would define ϕ as an *endomorphism* within the category of (discrete) dynamical systems.

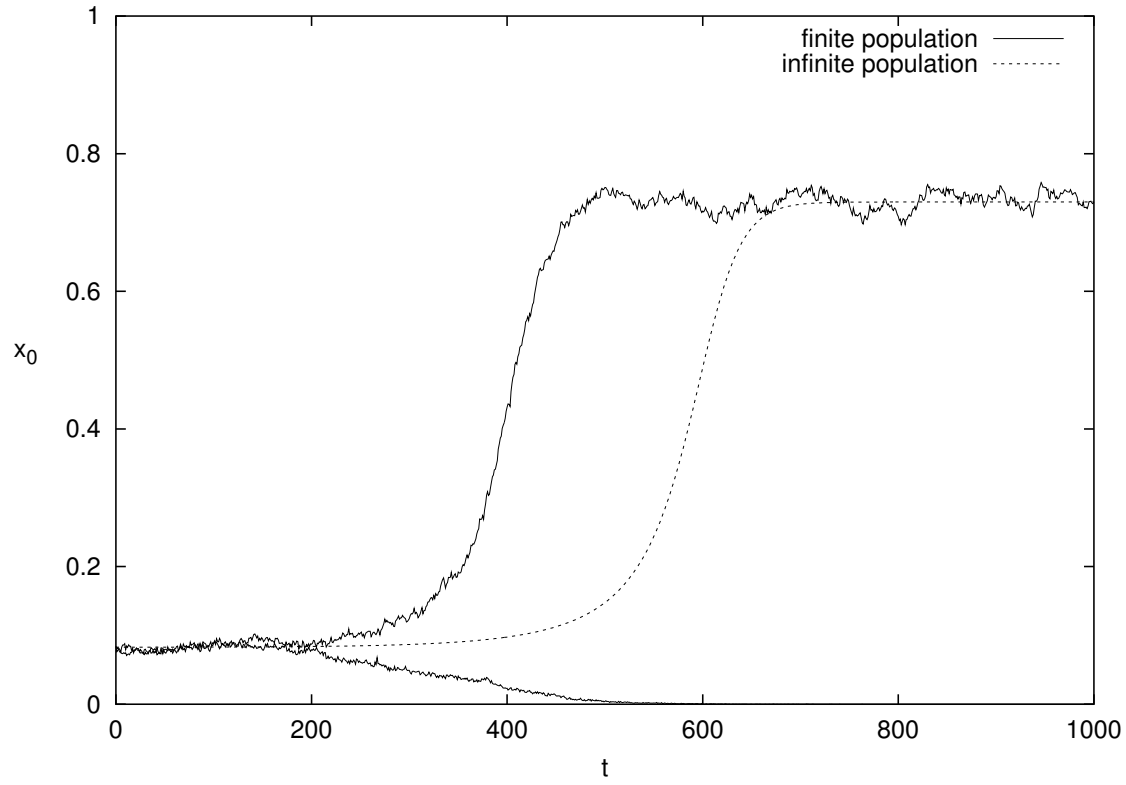

Figure 7.9: optimum sequence concentration plotted against time for two typical simulations of a finite population (stochastic) sexual quasi-species initialised near the unstable equilibrium, alongside the corresponding infinite-population model (7.16). Sequence length is $L = 80$, selection coefficient $\sigma = 0.4$, per-sequence mutation rate $U = 0.1$ and population size for the finite population runs is 10,000.

above the level of the unstable equilibrium before selection could begin to "pull" it towards fixation - and in the meantime mutation and recombination actually conspire to *reduce* its concentration.

We term the latter effect the ***bi-stability barrier***. For large populations in particular, it is difficult to see how a new, selectively advantageous sequence could ever fixate; far from acting as an "error repair" mechanism, recombination appears to act as a "selective advantage obliteration" mechanism!

Another striking difference is the following: in the asexual case, if the quasi-species is in equilibrium just within the error threshold we would expect to see a low concentration x_0 of the optimal sequence (Eq. 7.14 and Fig. 7.3). With recombination, at the stable equilibrium, we would expect to see a substantial concentration of the optimal sequence (Fig. 7.5), particularly if the selection coefficient σ is small - in which case (Eq. 7.28) we have $x_0 \approx 1/e \approx 0.3679$. Thus if we observed a sexual population in equilibrium to have a reasonably high concentration of the optimum sequence we could not infer, as we might in the asexual case, that the mutation rate was well within the error threshold; in effect, a small change in mutation rate or selection pressure could push a seemingly stable sexual population catastrophically over the error threshold. Furthermore, near the error threshold the stable and unstable equilibria are close together; a stochastic fluctuation could easily bump the optimum concentration below the unstable equilibrium.

Finally, it was remarked in Section 7.3.3 that our model is similar to that in (Boerlijst et al., 1996). The principal difference is that in their model recombination occurs only with a given probability < 1. They also consider fitness landscapes with a "plateau" of higher fitness around the optimum sequence as well as an isolated fitness spike. We conjecture that the picture presented in this Section holds in general for (local) non-synergistic fitness optima. We do note that an optimum representing a fitness plateau (rather than, as in our case, a spike) might alter our conclusions somewhat; in particular back-recombination might be expected to reduce the bi-stability barrier. Simulations by (Ochoa & Harvey, 1999) suggest that this might be the case; analysis would, however, be more difficult under this scenario. By contrast, a locally optimal neutral network representing a "ridge"-like rather than a "plateau"-like optimum should be easier to analyse (Reidys & Stadler, 2001) and we would expect similar conclusions to those presented here. We should also like to analyse finite population effects, although this is difficult even for the asexual case (Nowak & Schuster, 1989).

Chapter 8

Conclusion

The thesis presented in this work is rooted in the following philosophy: *the more we know of the statistical properties of a class of fitness landscapes, the better equipped will we be for the design of effective search algorithms for such landscapes*. On the basis of some defining assumptions regarding the class of fitness landscape which we explicitly address, we have characterised evolutionary processes for such landscapes and attempted to develop statistics that could yield insights into and assist us in the design of evolutionary search techniques that might exploit the generic evolutionary dynamics which we have identified. To aid us in this endeavour, we introduced *model landscapes* in several capacities: to sharpen our intuition, test our analysis and (hopefully) as useful models for real-world artificial fitness landscapes - "useful" in the sense that theoretical results for our models might apply, at least to some degree, to real landscapes - that we might, in other words, "fit" our models to real-world landscapes. This thesis, then, has been devoted to the discovery of statistics, models and techniques relevant to the design of effective evolutionary search in order to exploit the assumed landscape structure and concomitant evolutionary dynamics.

In a nutshell, the class of fitness landscapes we have addressed ourselves to are *discrete, correlated* landscapes with *large-scale neutrality* and the characteristic evolutionary dynamics identified for such landscapes involve *neutral drift* on *neutral networks*, punctuated by the sporadic discovery of *portals* to higher-fitness networks. Exploiting these features has involved identification of a balancing act between maximising neutral drift whilst retaining fitness correlation. Some casualties along the way have been recombination and an unquestioning assumption of the effectiveness of population-based search. In their place have appeared neutral drift, stochastic hill-climbing and (adaptive) optimisation of mutation rates.

8.1 Review of Results

In the introductory Chapter 1 we present some general discussion on optimisation and evolutionary search and attempt to lay out the rather specific type of evolutionary optimisation scenario addressed by this thesis. It is stressed that we are concerned with a particular class of (discretely encoded) fitness landscapes: those featuring some correlation and substantial neutrality. We examine why we might want to study optimisation on such landscapes and in particular why they

might arise in complex "real world" optimisation problems. We also present some discussion on the rôle(s) and usefulness of statistics and model landscapes.

Chapter 2 introduces fitness landscapes, in particular their structure with respect to *mutation* and *neutral networks*. While largely concerned with formalities, it also presents some novel statistical measures, the utility of which will be examined in later Chapters. These are divided into *fitness-independent* statistics, which depend only on the partitioning of a landscape into neutral networks and *fitness-dependent* statistics, which depend in addition on actual fitness values. The former include:

- The *mutation matrix* $\boldsymbol{m}(U)$ with respect to a mutation operator U for a neutral partitioning of a fitness landscape (Section 2.2.2), which encapsulates the "coarse-grained" structure of the landscape with respect to mutation, and which emerges as the basis for a *maximum entropy approximation* of an evolutionary process in the next Chapter. The algebraic structure of mutation matrices is examined and several approximations are presented for expressing a general mutation matrix in terms of the matrix for 1-point mutation (Section 2.2.4). These approximations are put into practice in Chapter 5.

  It is also proved (Section 2.2.5, Prop. 2.2.2) that the mutation mode/rate which maximises the probability of mutating from one neutral network to another is *constant* (n-bit) mutation at a rate that may be calculated from the mutation matrix.

- The *entropy* $H(U)$ of a neutral network (Section 2.2.3, Def. 2.2.4), which re-appears later in the Chapter as the basis for the *percolation index* $\mathcal{P}(U)$ (Eq. 2.49) of a neutral network - a measure of the "innovation rate" or "accessibility" of other networks via mutation (Section 2.2.6). It is also demonstrated how the neutral contribution may be "factored out" of these statistics.

- The *Markov index* $\mathcal{M}(U)$ of a neutral network (Section 2.2.3, Def. 2.2.5), a measure of the *homogeneity* of a neutral network, or the degree of "localisation" of mutational information on a network. This mutual information-based statistic is proposed as a measure of how well the maximum entropy approximation of the following Chapter is likely to work. It also emerges later in Chapter 2 as the basis for the *percolation drift factor* $\mathcal{D}^{perc}(U)$ (Eq. 2.58), a measure of how important *neutral drift* is likely to be as regards network accessibility (Section 2.2.6).

The necessity of fitness-dependent statistics is stressed as a consequence of the inadequacies of more traditional landscape statistics (such as fitness auto-correlation) which are based essentially on *uniform sampling* of the landscape. Statistics introduced, based on the full *mutant fitness distribution*, include:

- The *mean mutant fitness* $\mathcal{F}(U|w)$; i.e. the expected fitness of a mutant of a sequence of given fitness. It is demonstrated that the auto-correlation $\rho(U)$ depends (at least for uniform mutation operators) on mutation probabilities only through this function; implications for the usefulness of auto-correlation are discussed.

- *Linear correlation* (Section 2.3.3) - where the mean mutant fitness depends linearly on actual fitness. We show that for landscapes with this property mutation decreases fitness "on average" by a factor equal to the auto-correlation. In Chapter 5 and Chapter 6 we verify this property for our model landscapes and conjecture that it may be a ubiquitous and important statistical feature of artificial fitness landscapes in general.

- The *evolvability* statistic $\mathcal{E}(U|w)$ (Section 2.3.4) - the (fitness-conditional) probability of finding *fitness increasing* mutations. It is argued that, while evolvability - or perhaps more properly the *decay* of evolvability with increasing fitness - may be a more meaningful metric than auto-correlation as regards the behaviour of evolutionary processes, we might nonetheless expect to find some degree of evolvability on correlated landscapes.

  Analogous to Prop. 2.2.2 we show (Prop. 2.3.1) that, for a given fitness, the *optimal mutation mode/rate* - in the sense of that which maximises *evolvability* - is also *constant* (n-bit) mutation, now at a fitness-dependent rate. It is noted that this contradicts the common practice among GA practitioners of mutating on a *per locus* basis.

- The *evolvability drift factor* $\mathcal{D}^{evol}(U)$ (Eq. 2.72) - a measure of the extent to which the probability of discovering fitness-increasing mutations depends on the particular sequence of a neutral network. This mutual information-based statistic is proposed as a measure of the importance of drift to evolvability, a theme to be revisited in more detail in Chapter 4.

In the final Section 2.4 we extend our statistical techniques to families of *random fitness landscapes*. In particular, we draw the distinction between *ensemble* statistics and "averaged" per-landscape statistics; it is noted in particular that, since there will in general be no neutral partitioning valid across a family of random landscapes, neutral network statistics must necessarily be fitness-conditional.

The first couple of Sections of Chapter 3 are also concerned with formalities, aimed at capturing in quite precise terms what we mean by an *evolutionary process*; that is, encapsulating mathematically the notion of a *population* evolving on a fitness landscape via *fitness-based selection* and *heritable random variation*. Evolutionary processes are thus defined as Markov processes on the state space of populations on a sequence space, with transitions from one population to the next "generation" defined by an *evolutionary operator*. An evolutionary operator, in turn, is defined by a fitness-based, stochastic *selection operator* in combination with a mutation operator (as encountered in the previous Chapter) to supply the (heritable, random) variation. Capturing the notion of fitness-based selection in a suitably general sense turns out to be non-trivial and is thus relegated to the technical Appendix A; several practical examples are presented to substantiate the formalism. Our construction is designed to correspond to the intuitive notion that to form a new generation we create mutants of sequences from the current population and select for the next generation - stochastically, but solely on the basis of fitness - from mutants and un-mutated original sequences. The approach is general enough to cover most mutation-based search processes that would likely be accepted as constituting Genetic Algorithms, as well as including such (less manifestly "evolutionary") processes as *stochastic hill-climbing* and *simulated annealing*. Obvious non-contenders for our definitions are *recombination* and algorithms (such as "spatial" GA's) incorporating state information besides sequence/fitness; we note that our definitions might, if desired, be extended to incorporate both of these aspects without major conceptual complications. We remark too that if the formalism in the opening Sections to this Chapter appears unnecessarily pedantic, some degree of semantic (if not mathematical) precision will be required at least in Chapter 5 when we argue (Section 5.3) the optimality of the *netcrawler* process - *within the class of evolutionary processes* - for search on a particular family of fitness landscapes.

Section 3.3 outlines the *statistical dynamics* approach to the analysis of evolutionary processes, as propounded by (Nimwegen et al., 1997). Thus we review the *coarse-graining* of a landscape via

a neutral partitioning and the *maximum entropy approximation* to an evolutionary process, by way of which the state space of evolutionary search is "collapsed" to a (hopefully) more manageable form. The approximation is related *post hoc* to some statistical constructions already introduced in Chapter 2, in particular the Markov index of a neutral network. Section 3.4 reviews the typical *epochal dynamics* of evolutionary processes on fitness landscapes with neutral networks. The characteristic dynamics of evolution with neutral networks is contrasted with the more traditional GA picture of evolution (with recombination) on rugged landscapes. Section 3.4.1 discusses the dynamics behind the "punctuated" aspect of epochal dynamics with neutral networks, identifying the crossing of *entropy barriers* - the discovery of *portals* to higher fitness networks - rather than *fitness barriers* as the crucial factor in search effectiveness and the consequent importance (and implications) of *neutral drift*. The Chapter closes with a Section (3.5) on the measurement and comparison of search efficiency: it is emphasised that efficiency comparisons should be on a *per-fitness-evaluation* basis and introduces the notion of *time-* versus *fitness-critical* evaluation standards.

In Chapter 3 neutral drift was identified as an important feature of evolution with neutral networks. In Chapter 4 we investigate how we might *exploit* neutral drift - specifically, to maximise the efficiency of search for portals to higher neutral networks. We work towards the conclusion that - in lieu of specific structural information on the accessibility of portals and *independently of neutral network structure (with regard to mutation)* - our evolutionary process should always attempt to maximise drift in the sense that, when faced with a choice, *selection should generally choose a neutral mutant at the expense of its parent*. It is demonstrated (via counter-example) that if we *do* have further structural information on portal accessibility - if, for instance, we make a (not unreasonable) assumption that portal sequences are likely to suffer a higher than average probability of *fitness-decreasing* mutation (Example 4.1.1) - then maximising drift may not, in general, be the best strategy.

The argument in favour of maximising drift works roughly on the basis that the more "independent" are two sequences on a neutral network, the greater the probability that at least one of them find a portal (Prop. 4.0.1). The *evolvability drift factor* $\mathcal{D}^{evol}$ (Eq. 2.58) of Chapter 2 appears as a measure of this effect. A more sophisticated argument - taking into account *genealogies* of sequences - uses a novel variety of neutral walk with tunable drift, which we dub a *nervous ant* walk (Example 3.2.5 and Section 4.1), to investigate the dependency of the distribution of time to portal discovery on drift rate. A neutral ant walk moves to neutral mutants with a fixed probability that controls the drift (or diffusion) rate of the process on a neutral network. We prove the (Weak) Neutral Drift Theorem (Theorem 4.1.1), which states roughly that, in the absence of specific information regarding portal distribution, increasing the drift rate (at least slightly) from zero always improves the chances of portal discovery in the sense that the probability of discovery of a portal within any given time is increased. This result is quite general - it holds true regardless of network topology or distribution of portals on the network. We conjecture (Conjecture 4.1.1) that, on any given time scale, the portal discovery probability in fact increases monotonically with increasing drift rate, up to maximum drift. We also examine the behaviour of the nervous ant in the long time limit (Section 4.1.1) and calculate its *diffusion coefficient* in terms of network neutrality, mutation rate and drift parameter (Eq. 4.48, 4.49). Finally (Section 4.2), we discuss briefly how our conclu-

sions might apply to *populations*, particularly with regard to known results on the phenomenon of *mutational buffering*, whereby a population diffusing on a neutral network preferentially samples network sequences of higher neutrality.

Chapter 5 introduces the first of our families of statistical models for correlated fitness landscapes with neutral networks, ε-*correlated* landscapes. These are characterised by a "ladder-like" structure, where fitness-increasing mutation probability - evolvability - is controlled by a small order parameter (the "ε") such that, to first order in ε, the only fitness-increasing mutations that may occur with better-than-negligible probability are to the "next network up". Working in general to leading order in ε, it is shown (Eq. 5.2) that (maximal) neutral network size decays (at least) exponentially with increasing fitness. Using the *multiplicative mutation approximations* (Section 2.2.4) of Chapter 2 we are able to calculate explicitly the *n*-point mutation matrix (Eq. 5.3, 5.4), which we use to calculate (Prop. 5.2.1) the *optimal (constant) mutation rate* for ε-correlated landscapes as predicted by Prop. 2.3.1 of Chapter 2. We also calculate the optimal Poisson mutation rate.

In Section 5.3 we deploy the evolutionary process formalism of Chapter 3 and results from Chapter 4 regarding the utility of neutral drift to argue that the optimal search process on an ε-correlated landscape - within the class of mutation-based evolutionary processes as we have defined them - is the *netcrawler* process of Example 3.2.5, a stochastic hill-climber which always accepts neutral moves. In Section 5.3.1 we derive the $1/e$ *Neutral Mutation Rule* (Prop. 5.3.1) for optimising (constant or Poisson) mutation rates which says that, to a first approximation, if our mutation rate is optimal then the observed fraction of neutral mutations should be $1/e$. The rule is proposed as a general heuristic for setting mutation rates on correlated fitness landscapes with neutral networks and we describe the *adaptive netcrawler* evolutionary process, which uses online monitoring of observed neutrality to self-tune its mutation rate. The remainder of the Section examines the random search and netcrawler processes in detail on ε-correlated landscapes, deriving analytic expressions for mean first passage times to achieve a given fitness and expected fitness within a given number of fitness evaluations.

Section 5.4 introduces Royal Road landscapes within the context of ε-correlated landscapes. Many of the statistics introduced in Section 2.2 and Section 2.3 are calculated analytically and implications of results discussed. In particular, we find that:

- Evolvability decays *linearly* with fitness (Eq. 5.32), so that Royal Road landscapes are comparatively "easy" to optimise.
- Percolation of neutral networks is low.
- The Markov index scales as $\boldsymbol{O}(\varepsilon \log \varepsilon)$ (Eq. 5.46) so that the maximum entropy approximation should work well.
- The evolvability drift factor is high, particularly for large block size - i.e. small ε (Eq. 5.48) - so that neutral drift should be a significant factor in locating portals.
- Royal Road landscapes are *linearly correlated* (Eq. 5.51) and approximately *elementary* (Eq. 5.52).

Experiments were also performed to test the (analytic) Prop. 5.2.1 on optimum mutation rates for Royal Road landscapes. Results confirm the theoretically predicted optimum mutation rates for constant (*n*-bit) and Poisson mutation (for small ε) and also confirm Prop. 2.3.1 of Chapter 2

insofar as constant mutation out-performs Poisson mutation. A practical implementation of an *adaptive netcrawler* is described and it is confirmed experimentally that it is able to track the optimum mutation rate quite accurately, corroborating the effectiveness of the $1/e$ Neutral Mutation Rule.

Section 5.4.2 examines evolutionary search performance on Royal Road landscapes. Firstly, performance comparison criteria are discussed (including a statistical methodology for estimating fitness-critical performance) and experimental performance results for the netcrawler are found to be in good agreement with theoretical predictions of Section 5.3.3. The remainder of the Section describes and trials a wide range of GA's (with and without recombination) on Royal Road landscapes. Results are analysed in detail and are found, in particular, to support our arguments (Prop. 5.2.1, Conjecture 5.3.1) proposing the netcrawler with constant optimal mutation, as the optimal evolutionary search process for ε-correlated landscapes. Results also support known results (Forrest & Mitchell, 1993; Mitchell et al., 1992) on problems with *recombination* on Royal Road landscapes, a topic to which we return in Chapter 7.

Chapter 6 presents our second model for landscapes with neutrality, the NKp family of random landscapes originally described by the author in (Barnett, 1997). NKp landscapes feature "independently tunable" correlation and neutrality. The Chapter opens with a Section on the historical background and some (physical and philosophical) motivations for the model, places the NKp model within the context of *generalised NK landscapes* and proceeds to details of the actual construction. The following Section examines the "global" statistics of the model. The auto-correlation function is calculated for generalised NK landscapes and its independence of the underlying fitness distribution - and hence in particular the independence of correlation and neutrality - is proved (Prop. 6.2.1). The independence of auto-correlation from the *number of features* also follows directly from the result. Generalised NK landscapes are also shown to be (at least approximately) *elementary*. Specialising to NKp landscapes, we introduce the notion of *contributing features* (Section 6.2.2) as an analytic tool and calculate several statistics conditional on the distribution of contributing features (and thence their global counterparts), including *mean fitness* and *fitness variance*, *neutrality*, *lethality* and the distribution of *neutral degree*. The dependence of these statistics on the model parameters is discussed.

Section 6.3 addresses the *fitness-dependent* statistics of generalised NK landscapes. We calculated the *mean mutant fitness* (Section 6.3.1) and prove that generalised NK landscapes have the *linear correlation* property (Prop. 6.3.1). As a corollary, we obtain another proof of the independence of correlation from the underlying distribution, although (as we stress) the result is in fact a more stringent statistical requirement. Specialising once more to NKp landscapes, we proceed to calculate fitness-conditional neutrality, lethality and neutral degree variance in terms of the underlying fitness distribution (Section 6.3.2). We also detail how to calculate the full *mutant fitness distribution* (Section 6.3.3) via its moment generating function and proceed to an explicit calculation for the case of a *Gaussian* underlying distribution. This is used to calculate the *(ensemble) evolvability* statistic. It is (correctly) predicted, on the basis of the scaling of evolvability that NKp landscapes are in general "hard" to optimise, at least in comparison with, say, the Royal Road landscapes of the previous Chapter. The evolvability is also deployed to calculate an (ensemble, fitness-dependent) optimal mutation rate. This rate is found to compare favourably with a

mutation rate predicted as optimal by the $1/e$ Neutral Mutation Rule as applied to the (ensemble, fitness-dependent) neutrality calculated previously. We supply an alternative derivation of the $1/e$ Rule based on some simple assumptions regarding local neutral network structure.

In section Section 6.4 we discuss the modelling of real-world artificial fitness landscapes by the NKp scheme. We propose a "baseline" set of parameters, intended to capture some aspects of real landscapes (neutrality, lethality, correlation, *etc.*), for experimentation. We address the problem of (uniform/fitness-dependent) sampling of large NKp landscapes - specifically the rarity of higher fitness sequences - and present a sampling approach using *simulated annealing*, which proved satisfactory in practice. Section 6.4.1 investigates how, given a real-world correlated fitness landscape with neutrality, we might (using statistical features of the model derived in earlier Sections) attempt to fit parameters to an NKp model. Section 6.4.2 raises some issues regarding computer implementation of NKp landscapes. Section 6.4.3 investigates some (fitness-dependent) statistical properties of actual neutral networks on NKp landscapes; since analysis proved intractable (to the author!) for many of the properties examined, much of the Section is of an empirical nature. Properties examined include neutral network size, number of networks, network connectivity and network percolation/innovation. Results are sometimes counter-intuitive: eg. it is found that as we ascend an NKp landscape, while neutral networks shrink rapidly in size, they also percolate more (in the sense of having more accessible neighbouring networks). It is found that this effect cannot be ascribed simply to a reduction of percolation by neutrality.

Finally, Section 6.4.4 presents some preliminary results on optimisation on NKp landscapes via stochastic hill-climbing. Results corroborate previous analysis regarding evolvability, optimal mutation rates and the $1/e$ Neutral Mutation Rule; they also suggest that, despite previous caveats, *ensemble* statistics (eg. for neutrality and evolvability) may indeed be useful. The picture presented in Chapter 3 of evolution with neutral networks - i.e. of neutral drift punctuated by the crossing of entropy barriers - is confirmed, although there is a suggestion that fitness barriers exist on a "global" scale, much in the manner of the original NK landscapes (Kauffman, 1993). Preliminary research also indicates that the most effective evolutionary search process on NKp landscapes is likely to be simulated annealing, rather than any population-based algorithm.

In Chapter 7 we finally address the issue of *recombination*, which we have hitherto rejected as a useful search mechanism for the class of optimisation problems with which this thesis concerns itself. In Section 7.1 we review some known (and some novel) problematic issues with the so-called *Building Block Hypothesis*; in particular, we question the likelihood of suitable building blocks existing in the first place and also the ability of recombination to assemble building blocks effectively. Section 7.2 reviews the well-known (related) phenomena of *genetic drift*, *premature convergence* and *hitch-hiking* as regards their impact on the effectiveness of recombination.

Section 7.3 presents new research by the author (inspired by a model for retrovirus infection from population genetics), identifying a *bi-stability barrier* to the fixation of beneficial mutations - as well as a lowering of the mutational *error threshold* - as a result of the interaction of recombination with certain local features of a fitness landscape. This work represents in some sense the obverse to the well-known benevolent effect (the lowering of the *mutational load*) of recombination in the presence of *synergistic epistasis* (Kondrashov, 1982). More specifically, it is found (Section 7.3.3) that if there is *non*-synergistic epistasis in the neighbourhood of a fitness peak, then

the error threshold takes the form of a *continuous* phase transition (as opposed to the discontinuous transition found with mutation alone). Within the error threshold, the long-term behaviour of the population distribution around the fitness peak - the (equilibrium) *quasi-species* distribution - splits into a stable solution representing a population converged around the peak and a solution representing a *delocalisation* of the quasi-species - i.e. loss of the fitness peak. These two stable equilibria are separated by an unstable equilibrium which, we argue, represents a *barrier* to the fixation of a sequence discovering the fitness peak, in the sense that, below the barrier concentration, recombination (and mutation) will tend to destroy peak sequences faster than selection can replicate them. Although the analysis is based on the (infinite population limit) quasi-species formalism, it is indicated how finite-population sampling effects might exacerbate the problem. The quasi-species distribution, error thresholds and barrier height are calculated explicitly in the limits of strong and weak selection (Section 7.3.4). In the limit of weak selection the barrier height is found to approach $1/e$ and error thresholds are found always to be (substantially) *lower* than the comparable scenario without recombination. Stability of the equilibria is also analysed in some detail (Section 7.3.5).

8.2 Directions for Further Research

This thesis has been concerned explicitly with correlated fitness landscapes in artificial evolution with neutral networks; yet, it might be said that remarkably little is actually known about the structure of fitness landscapes that arise in complex real-world optimisation problems such as robot control, hardware evolution, *etc.* There is indeed accumulating evidence that such problems do not necessarily (or in general) give rise to landscapes that fit the more traditional rugged, multimodal, non-neutral conception (Goldberg, 1989; Kauffman, 1993; Michaelewicz, 1996) but more research is required. At the very least, it is hoped that GA practitioners might be persuaded to investigate more thoroughly the structure of their fitness landscapes rather than assuming a traditional rugged structure.

Areas concerning neutrality that this thesis *hasn't* addressed include:

- *"Near neutrality"*: It seems reasonable that if, for example, there are extensive regions of sequence space for which fitness varies but slightly, then the behaviour of evolving populations might be as if those regions were truly equi-fit. But some thought turns up problems with such an assumption:

  - Might not even a small fitness differential - a gentle "shelving" of the landscape, perhaps - be enough to cause a population to "ratchet up" a slight gradient?
  - Does the neutral network concept carry over at all? If we were, for instance, to decide on an arbitrary "quantisation" of a landscape into equi-fit "bands" (*cf.* (Newman & Engelhardt, 1998)), why should evolutionary dynamics respect this banding?

  It is known from population genetics (Ohta, 1992; Kimura, 1983; Kimura & Ohta, 1969) that there is a relationship between the smallest fitness differential - the smallest selective advantage - that evolution "sees" (as non-neutral), population size and mutation rate. A more comprehensive "nearly-neutral network theory" might take this relationship as a starting point.

- *Neutral dynamics*: The dynamics and structure of populations evolving even on a perfectly flat landscape are surprisingly complex and difficult to analyse. What is known reveals a rich "clustered" structure deriving from the formation and extinction of genealogies of sequences (Derrida & Peliti, 1991), in the manner of *coalescent* statistics. More detailed knowledge of this structure - perhaps using a *diffusion approximation* (Kimura, 1964) - would be useful for more accurate analysis of eg. portal discovery times in a drifting population (Nimwegen et al., 1997). Other questions begging clarification include:

  - How does network topology affect evolutionary dynamics? That it is significant is not in doubt, as the phenomenon of *mutational buffering* demonstrates (A. Wagner & Stadler, 1999; Nimwegen et al., 1999; Wilke, 2001); but it also seems likely that the relationship between network topology and population structure/dynamics (Bullock, 2002) is likely to be a complex one.
  - May we prove our Strong Neutral Drift conjecture (Conjecture 4.1.1)? How might such results on the "benefit" of drift be extended to eg. more complex population structures and more relaxed assumptions on *a priori* knowledge of landscape/network structure?
  - Why do we (in the absence of recombination) need *populations* at all? Given a scenario where we are dealing with entropy rather than fitness barriers, does the plausible justification for population-based search (on correlated landscapes) - that a population may search the neighbourhood of a peak without losing the current (local) optimum - still apply? Or, as our results with simulated annealing on NKp landscapes (Section 6.4.4) seem to suggest, might we be better off with some variety of population-of-one hill-climber?

- *Fitness noise*: Fitness evaluation in many (if not most) complex, real-world optimisation scenarios is essentially *stochastic* (Jakobi et al., 1995) - fitness evaluation of the same sequence will not yield the same value twice. As for near neutrality this poses problems for neutral network theory: when do two sequences belong to the same network? Again, arbitrary banding of fitness may be un-elucidating and it may, furthermore, be difficult to spot gradients up which populations might "ratchet".

  The situation, it might be remarked, is far from clear. As an example, some rather counter-intuitive results by the author (in preparation) analyse a situation where (fitness-proportional) selection is "blind" to noise with an *arbitrary degree of variance* - so long as the noise is the right "shape" and scales appropriately with fitness.

- *Linear correlation*: We have noted the apparent ubiquity of linear correlation (Section 2.3.3, 5.4.1, 6.3.1), where the relationship between parent and mutant fitness is (perhaps approximately) linear. Can we use this property directly to help predict evolutionary dynamics?

- *Coding*: Several issues arise regarding the relationship between the *coding* of an optimisation problem - the sequence $\rightarrow$ fitness mapping - and landscape statistical properties such as neutrality, correlation and evolvability. For example:

  - To what extent might landscape structure - eg. correlation, neutral network topology, scaling of evolvability, *etc.* - be "inherent" in a problem as opposed to being "coding-dependent"? How for example, is linear correlation affected by a re-coding of the fitness function? What classes of fitness landscape are re-codable into a linearly correlated landscape?
  - May we "exploit" neutrality/neutral drift by deliberately coding for neutral networks - perhaps with a view to replacing fitness barriers with the "easier" entropy barriers

(Shipman, Shackleton, M. Ebner, & Watson, 2000)? Or is this a vain pursuit (Bullock, 2001) - are there, perhaps, "No Free Lunch" theorems for neutrality?

- We have, it might seem, pretty much written off recombination for optimisation on our correlated, neutral landscapes. Yet, as remarked, GA researchers do frequently report better results with recombination. Is this because of the types of evolutionary algorithm they use (and might they not be better off with one of our mutation-based processes...?) or is there indeed a useful rôle for recombination? If so, how might the problems identified in Chapter 7 be addressed? Do, eg. the bi-stability barriers of Section 7.3 arise in realistic artificial optimisation scenarios?

- We have been at pains to point out that results should not be extrapolated beyond the *discrete* sequence spaces addressed in this thesis to the optimisation of *continuous* parameters (such as frequently arise eg. in the encoding of neural networks). Might any of our research in fact have application to continuous parameter optimisation - or will we be again run into awkward problems with gradients and definitions of neutral networks? Does discrete encoding (eg. Gray coding) of continuous parameters give rise to (discrete) landscapes with correlation and/or neutral networks? Are there, indeed, any good reasons for discrete encoding of continuous parameters?

8.3 Closing Remarks

It may appear from the somewhat lengthy list above that we have, ultimately, succeeded in asking more questions than we have answered. This we view as to the good: it is the hallmark of an area rich and fertile for new research and one that we hope may, in the best scientific tradition, continue to challenge orthodoxies and reveal fresh insights into old problems.

References

Altenberg, L. (1994). The evolution of evolvability in genetic programming. In K. E. Kinnear (Ed.), *Advances in genetic programming* (p. 47-74). Cambridge, MA: MIT Press. 28, 31, 32, 141

Altenberg, L. (1995a). NK fitness landscapes. In T. Bäck, D. Fogel, Z. Michaelewicz (Eds.), *Handbook of evolutionary computation* (chap. B2.7.2). Oxford University Press. 105, 129

Altenberg, L. (1995b). The Schema Theorem and Price's Theorem. In D. Whitley M. D. Vose (Eds.), *Foundations of genetic algorithms 3* (p. 23-49). San Mateo, CA: Morgan Kaufmann. 31, 32, 143

Anderson, P. W. (1985). Spin glass Hamiltonians: A bridge between biology, statistical mechanics and computer science. In D. Pines (Ed.), *Emerging synthesis in science: Proceedings of the founding workshops of the santa fe institute.* Santa Fe , NM: Santa Fe Institute. 103

Baake, E., Wiehe, T. (1997). Bifurcations in haploid and diploid sequence space models. *J. Math. Biol.*, *35*, 321. 152

Babajide, A., Hofacker, I., Sippl, M., Stadler, P. F. (1997). Neutral networks in protein space: A computational study based on knowledge-based potentials of mean force. *Folding Design*, *2*, 261-269. 7

Back, T., Hoffmeister, F., Schwefel, H.-P. (1991). A survey of evolution strategies. In L. B. Belew R. K. Booker (Eds.), *Proc. 4th int. conf. on genetic algorithms* (p. 2-9). San Diego, CA: Morgan Kaufmann. 76

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In J. J. Grefenstette (Ed.), *Genetic algorithms and their applications: Proceedings of the second international conference on genetic algorithms.* Erlbaum. 93

Barnett, L. (1997). *Tangled webs: Evolutionary dynamics on fitness landscapes with neutrality*. Unpublished master's thesis, COGS, University of Sussex. (ftp://ftp.cogs.susx.ac.uk/pub/users/inmanh/lionelb/FullDiss.ps.gz) 8, 48, 68, 103, 104, 105, 109, 167

Barnett, L. (1998). Ruggedness and neutrality - the NKp family of fitness landscapes. In C. Adami, R. K. Belew, H. Kitano, C. Taylor (Eds.), *Alife vi, proceedings of the sixth international conference on artificial life* (p. 18-27). The MIT Press. (http://www.cogs.susx.ac.uk/users/lionelb/download/alife6_paper.ps.gz) 8, 48, 103, 104, 111, 133

Barnett, L. (2001, 27-30). Netcrawling - optimal evolutionary search with neutral networks. In *Proceedings of the 2001 congress on evolutionary computation cec2001* (pp. 30–37). COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea: IEEE Press. (http://www.cogs.susx.ac.uk/users/lionelb/download/netcrawl.ps.gz) 8, 43, 70, 71, 72, 76

Bastolla, U., Roman, H. E., Vendruscolo, M. (1998). *Neutral evolution of model proteins: Diffusion in sequence space and overdispersion* (Technical report, LANL preprint archive: No. cond-mat/9811267). LANL. 7

Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behaviour*, *3*(4), 471-511. 115

Boerlijst, M. C., Bonhoeffer, S., Nowak, M. A. (1996). Viral quasi-species and recombination. *Proc. R. Soc. Lond. B.*, *263*, 1577-1584. 147, 151, 161

Bollobás, B. (1985). *Random graphs*. London: Academic Press. 23

Bonhoeffer, L., Stadler, P. (1993). *Error thresholds on correlated fitness landscapes.* (In press)

146

Bullock, S. (2001). Smooth operator? Understanding and visualising mutation bias. In J. Kelemen P. Sosik (Eds.), *Sixth european conference on artificial life (ecal2001)*. Heidelberg: Springer-Verlag. 171

Bullock, S. (2002). Will selection for mutational robustness significantly retard evolutionary innovation on neutral networks? In *Artificial life viii: Proceedings of the eighth international conference on artificial life*. Cambridge, MA.: MIT Press. 69, 170

Catoni, O. (1996). Metropolis, simulated annealing, and iterated energy transformation algorithms: Theory and experiments. *Journal of Complexity*, *12*(4), 595–623. 40, 45

Charlesworth, B. (1990). Mutation-selection balance and the evolutionary advantage of sex and recombination. *Genet. Res.*, *55*, 199-221. 146, 148

Cliff, D., Husbands, P., Harvey, I. (1993). Evolving visually guided robots. In J.-A. Meyer, H. Roitblat, S. Wilson (Eds.), *From animals to animats 2: Proc. of the second intl. conf. on simulation of adaptive behaviour, (sab92)* (p. 374-383). Cambridge, MA.: MIT Press/Bradford Books. 8, 113

Crow, J. F., Kimura, M. (1970). *An introduction to population genetics theory*. New York, N. Y.: Harper and Row. 7, 11, 41, 43, 148, 152

Dennett, D. C. (1995). *Darwin's dangerous idea*. London: Penguin Books. 3, 8

Derrida, B., Peliti, L. (1991). Evolution in a flat fitness landscape. *Bull. Math. Biol.*, *53*(3), 355-382. 50, 51, 170

Eigen, M. (1971). Selforganization of matter and the evolution of biological macromolecules. *Natuurwissenschaften*, *10*, 465-523. 49, 146, 147

Eigen, M., McCaskill, J., Schuster, P. (1989). The molecular quasispecies. *Adv. Chem. Phys.*, *75*, 149-263. 29, 49, 146, 147

Eigen, M., Schuster, P. (1979). *The hypercycle - a principle of natural selforganization*. New York, N. Y.: Springer. 147

Ekland, E. H., Bartel, D. P. (1996). RNA-catalysed RNA polymerization using nucleotide triphosphates. *Nature*, *383*, 192. 8

Feller, W. (1966). *An introduction to probability theory and its applications*. New York: John Wiley. 84, 105

Fisher, R. A. (1930). *The genetical theory of natural selection*. Oxford: The Clarendon Press. 49

Fontana, W., Stadler, P. F., Bornberg-Bauer, E. G., Griesmacher, T., Hofacker, I. L., Tacker, M., et al. (1993). RNA folding and combinatory landscapes. *Phys. Rev. E*, *47*, 2083-2099. 7

Forrest, S., Mitchell, M. (1993). Relative building block fitness and the building block hypothesis. In D. Whitely (Ed.), *Foundations of genetic algorithms 2*. Morgan Kaufmann, San Mateo, CA. 44, 49, 78, 99, 141, 167

Forst, C. V., Reidys, C., Weber, J. (1995). Evolutionary dynamics and optimization: Neutral networks as model-landscapes for RNA secondary-structure folding-landscapes. *Lecture notes in Artificial Intelligence, 929: Advances in Artificial Life*. 8

Fu, Y. T., Anderson, P. W. (1986). Application of statistical mechanics to NP-complete problems in combinatorial optimization. *J. Phys. A: Math. Gen.*, *19*, 1605-1620. 103

Gantmacher, F. R. (1959). *Applications of the theory of matrices*. New York, N. Y.: Interscience Publishers, Inc. 17

Gantmacher, F. R. (1960). *The theory of matrices (vol. 1)*. New York, N. Y.: Chelsea Publishing Company. 67

Gavrilets, S., Gravner, J. (1997). Percolation on the fitness hypercube and the evolution of reproductive isolation. *J. Theor. Biol.*, *184*, 51-64. 143, 146

Glover, F., Laguna, M. (1993). Tabu search. In C. Reeves (Ed.), *Modern heuristic techniques for combinatorial problems*. Oxford, England: Blackwell Scientific Publishing. 40

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization & machine learning*. Addison-Wesley. 92, 145, 169

Goldschmidt, R. B. (1933). Some aspects of evolution. *Science*, *78*, 539-547. 3

Goldschmidt, R. B. (1940). *The material basis of evolution*. Seattle: University of Washington Press. 3

Grefenstette, J. J., Baker, J. E. (1989). How genetic algorithms work: A critical look at implicit parallelism. In *Proc. 3rd intl. conf. on genetic algorithms* (p. 20-27). San Mateo, CA: Morgan Kaufmann. 141

Grüner, W., Giegerich, R., Strothman, D., Reidys, C., Weber, J., Hofacker, I. L., et al. (1996). Analysis of RNA sequence structure maps by exhaustive enumeration: I. neutral networks; ii. structure of neutral networks and shape space covering. *Monatsh Chem.*, *127*, 355-389. 7, 13, 23

Harvey, I. (1997). Artificial evolution for real problems. In T. Gomi (Ed.), *Evolutionary robotics: From intelligent robots to artificial life (er'97)* (p. 127-149). Kanata, Ontario Canada: AAI Books. (5th Intl. Symposium on Evolutionary Robotics, Tokyo, April 1997, Invited paper) 113

Harvey, I., Thompson, A. (1996). Through the labyrinth evolution finds a way: A silicon ridge. In *Proc. 1st internatl. conf. evol. sys.: From biology to hardware (ices 96)*. Springer Verlag. 8, 48, 113, 144

Higgs, P. G. (1994). Error thresholds and stationary mutant distributions in multi-locus diploid genetic models. *Genet. Res.*, *47*, 2083-2099. 146

Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, S., Tacker, M., Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatsh Chem.*, *125(2)*, 167-188. 7, 114

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge, Massachusetts: The MIT Press/Bradford Books. 49, 100, 142, 143

Hughes, B. D. (1996). *Random walks and random environments* (Vol. II). Oxford: Clarendon Press. 23, 44, 67

Huynen, M. A. (1996). Exploring phenotype space through neutral evolution. *J. Mol. Evol.*, *63*, 63-78. 23

Huynen, M. A., Hogeweg, P. (1994). Pattern generation in molecular evolution: Exploitation of the variation in RNA landscapes. *J. Mol. Evol.*, *39*, 71-79. 28

Huynen, M. A., Konings, D., Hogeweg, P. (1993). Multiple codings and the evolutionary properties of RNA secondary structure. *J. Theor. Biol.*, *165*, 251-267. 28

Huynen, M. A., Stadler, P. F., Fontana, W. (1996). Smoothness within ruggedness: The role of neutrality in adaptation. *Proc. Natl. Acad. Sci. (USA)*, *93*, 397-401. 8, 23, 44, 48, 49, 68, 114

Jakobi, N., Husbands, P., Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In F. Moran, A. Moreno, J. J. Merelo, P. Chacon (Eds.), *Proc. 3rd european conference on artificial life, springer-verlag, lecture notes in artificial intelligence 929* (Vol. XVI, p. 704-720). Springer Verlag. 52, 113, 170

Jakobi, N., Husbands, P., Smith, T. (1998). Robot space exploration by trial and error. In J. Koza et al. (Eds.), *Proceedings of the third annual conference on genetic programming: GP98* (p. 807-815). Morgan Kaufmann. 4

Jakobi, N., Quinn, M. (1998). Some problems and a few solutions for open-ended evolutionary robotics. In P. Husbands J.-A. Meyer (Eds.), *Proceedings of evorob98*. Springer Verlag. 4

Kampen, N. G. van. (1992). *Stochastic processes in physics and chemistry*. North-Holland. 50, 90

Kauffman, S. A. (1989). Adaptation on rugged fitness landscapes. In D. Stein (Ed.), *Lectures in the sciences of complexity* (p. 527-618). Redwood City: Addison-Wesley. (SFI Sciences in the Sciences of Complexity, Lecture Volume I) 103

Kauffman, S. A. (1993). *The origins of order - orginization and selection in evolution*. Oxford University Press, New York. 30, 103, 104, 105, 138, 143, 168, 169

Kauffman, S. A., Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, *128*, 11-45. 103

Kimura, M. (1962). On the probability of fixation of mutant genes in a population. *Genetics*, *47*, 713-719. 49

Kimura, M. (1964). Diffusion models in population genetics. *J. Appl. Prob.*, *1*, 177-232. 49, 55, 170

Kimura, M. (1983). *The neutral theory of molecular evolution*. Cambridge University Press. 7, 49, 55, 169

Kimura, M., Maruyama, T. (1966). The mutational load with epistatic gene interactions in fitness. *Genetics*, *54*, 1337-1351. 146, 148, 188

Kimura, M., Ohta, T. (1969). The average number of generations until fixation of a mutant gene in a finite population. *Genetics*, *61*, 763-771. 49, 169

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671-680. 40, 45

Kondrashov, A. S. (1982). Selection against harmful mutations in large sexual and asexual populations. *Genet. Res.*, *40*, 325-332. 146, 148, 168

Lande, R. (1985). Expected time for random genetic drift of a population between stable phenotypic states. In *Proc. natl. acad. sci. usa* (Vol. 82, p. 7641-7645). 49

Landweber, L. F., Pokrovskaya, I. D. (1999). Emergence of a dual catalytic RNA with metal-specific cleavage and ligase activities. In *Proc. natl. acad. sci. usa* (Vol. 96, p. 173-178). 8

Lawler, E., Lenstra, J., Kan, A. R., Shmoys, D. (1985). *The travelling salesman problem. a guided tour of combinatorial optimization*. New York: John Wiley & Sons. 103

Layzell, P. (2001). *Hardware evolution: On the nature of artificially evolved electronic circuits*. Unpublished doctoral dissertation, COGS, University of Sussex. (http://www.cogs.susx.ac.uk/users/adrianth/lazwebpag/web/Publications/THESIS/ssxdphil.pdf) 8, 113, 144

Matsumoto, M., Nishimura, T. (1998, Jan.). Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modelling and Computer Simulation*, *8*(1), 3-30. 94, 129

Maynard Smith, J. (1978). *The evolution of sex*. Cambridge, UK: Cambridge University Press. 148

Maynard Smith, J. (1998). *Evolutionary genetics (2nd. ed.)*. Oxford, UK: Oxford University Press. 11, 40, 41, 49, 55, 148, 152

McIlhagga, M., Husbands, P., Ives, R. (1996a). A comparison of optimization techniques for integrated manufacturing planning and scheduling. In H.-M. Voigt, W. Eberling, I. Rechenberg, H.-P. Schwefel (Eds.), *Proceedings of the fourth conference on parallel problem solving from nature* (p. 604-613). Berlin: Springer. 8, 39

McIlhagga, M., Husbands, P., Ives, R. (1996b). A comparison of search techniques on a wing-box optimisation problem. In H.-M. Voigt, W. Eberling, I. Rechenberg, H.-P. Schwefel (Eds.), *Proceedings of the fourth conference on parallel problem solving from nature* (p. 614-623). Berlin: Springer. 39

Michaelewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs (2nd. ed.)*. Berlin: Springer-Verlag. 43, 94, 99, 169

Mitchell, M., Crutchfield, J. P., Das, R. (2000). Evolving cellular automata to perform computations. In T. Bäck, D. Fogel, Z. Michalewicz (Eds.), *Handbook of evolutionary computation*. Oxford University Press. 115

Mitchell, M., Forrest, S., Holland, J. H. (1992). The Royal Road for genetic algorithms: Fitness landscapes and GA performance. In F. J. Varela P. Bourgine (Eds.), *Proceedings of the first european conference on artificial life* (p. 245-254). Cambridge, MA: MIT Press. 49, 70, 78, 92, 141, 143, 145, 167

Mitchell, M., Holland, J. H., Forrest, S. (1994). When will a genetic algorithm outperform hill-climbing? In J. D. Cowan, G. Tesauro, J. Alspector (Eds.), *Advances in neural information processing systems 6.* Morgan Kaufmann. 44, 100

Moran, P. A. P. (1958). The effect of selection in a haploid genetic population. *Proc. Camb. Phil. Soc.*, *54*, 463-467. 42

Newman, M., Engelhardt, R. (1998). Effect of neutral selection on the evolution of molecular species. *Proc. R. Soc. London B*, *256*, 1333-1338. 169

Nimwegen, E. van, Crutchfield, J. P. (1998a). *Optimizing epochal evolutionary search: Population-size dependent theory* (Technical report No. 98-10-090). Santa Fe Institute. 8, 51, 71

Nimwegen, E. van, Crutchfield, J. P. (1998b). *Optimizing epochal evolutionary search: Population-size independent theory* (Technical report No. 98-06-046). Santa Fe Institute. 8, 31, 78, 100

Nimwegen, E. van, Crutchfield, J. P. (1999). *Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths?* (Technical report No. 99-07-041). Santa Fe Institute. 8, 49, 50

Nimwegen, E. van, Crutchfield, J. P., Huynen, M. (1999). *Neutral evolution of mutational robustness* (Technical report No. 99-03-021). Santa Fe Institute. 28, 60, 69, 112, 170

Nimwegen, E. van, Crutchfield, J. P., Mitchell, M. (1997a). Finite populations induce metastability in evolutionary search. *Phys. Lett. A*, *229*, 144-150. 8, 46, 48, 50, 115, 164

Nimwegen, E. van, Crutchfield, J. P., Mitchell, M. (1997b). *Statistical dynamics of the Royal Road genetic algorithm* (Technical report No. 97-04-035). Santa Fe Institute. 8, 9, 48, 51, 70, 145, 170

Nowak, M., Schuster, P. (1989). Error thresholds of replication in finite populations: Mutation frequencies and the onset of Muller's ratchet. *J. Theor. Biol.*, *137*, 375-395. 49, 146, 161

Ochoa, G., Harvey, I. (1999). Recombination and error thresholds in finite populations. In *Proceedings of foga 5 (foundations of genetic algorithms).* (ftp://ftp.cogs.susx.ac.uk/pub/users/inmanh/fogat.ps.gz) 147, 161

Ochoa, G., Harvey, I., Buxton, H. (1999). On recombination and optimal mutation rates. In *Proceedings of genetic and evolutionary computation conference (gecco'99).* 147

Ohta, T. (1992). The nearly neutral theory of molecular evolution. *Annual Review of Ecology and Systematics*, *23*, 263-286. 169

Price, G. R. (1970). Selection and covariance. *Nature*, *227*, 520-521. 143

Reidys, C. (1995). *Neutral networks of RNA secondary structures*. Unpublished doctoral dissertation, Friedrich-Schiller-Universität, Jena. 103

Reidys, C., Forst, C. V., Schuster, P. (1998). *Replication and mutation on neutral networks of RNA secondary structures* (Technical report No. 98-04-036). Santa Fe Institute. 8, 48

Reidys, C., Stadler, P. F. (2001). Neutrality in fitness landscapes. *Applied Mathematics and Computation*, *117*(2-3), 321-350. 34, 104, 109, 113, 114, 147, 161

Reidys, C., Stadler, P. F., Schuster, P. (1997). Generic properties of combinatory maps: Neutral networks of RNA secondary structures. *Bull. Math. Biol.*, *59*(2), 339-397. 23, 103

Rendel, J. M. (1979). Canalization and selection. In J. N. Thompson Jr. J. M. Thoday (Eds.), *Quantitative genetic variation* (p. 139-156). New York: Academic Press. 28

Sarma, J., Jong, K. D. (1999). The behavior of spatially distributed evolutionary algorithms in non-stationary environments. In W. Banzhaf et al. (Eds.), *Proceedings of the genetic and evolutionary computation conference (gecco'99)* (p. 572-578). San Francisco CA.: Morgan Kauffman. 39

Schuster, P., Fontana, W., Stadler, P. F., Hofacker, I. (1989). From sequences to shapes and back: A case study in RNA secondary structures. *Proc. Roy. Soc (London) B*, *255*, 279-284. 7, 13

Schuster, P., Stadler, P. (1994). Landscapes: Complex optimization problems and biopolymer structures. *Computers Chem.*, *18*, 295-314. 146

Seneta, E. (1973). *Non-negative matrices - an introduction to theory and applications*. London: George Allen & Unwin Ltd. 17, 59

Sherrington, D., Kirkpatrick, S. (1975). Solvable model of a spin glass. *Phys. Rev. Lett.*, *35*, 1792-1796. 103

Shipman, R., Shackleton, M., M. Ebner, M., Watson, R. (2000). Neutral search spaces for artificial evolution: A lesson from life. In M. A. Bedau, J. S. McCaskill, N. H. Packard, S. Rasmussen (Eds.), *Artificial life vii* (p. 162-169). Cambridge, MA: MIT Press. 171

Smith, T., Husbands, P., Layzell, P., O'Shea, M. (2002). Fitness landscapes and evolvability. *Evolutionary Computation*, *10*(1), 1-34. 28, 31, 122

Smith, T., Husbands, P., O'Shea, M. (2001a). Neutral networks in an evolutionary robotics search space. In *Proceedings of the 2001 congress on evolutionary computation (CEC'2001)* (p. 136-145). Piscataway, New Jersey: IEEE Press. 8, 48

Smith, T., Husbands, P., O'Shea, M. (2001b). *Not* measuring evolvability: Initial exploration of an evolutionary robotics search space. In *Proceedings of the 2001 congress on evolutionary computation (CEC'2001)* (p. 9-16). Piscataway, New Jersey: IEEE Press. 28, 31, 113, 122

Smith, T., Philippides, A., Husbands, P., O'Shea, M. (2002). Neutrality and ruggedness in robot landscapes. In *Proceedings of the 2002 congress on evolutionary computation (CEC'2002)* (p. 1348-1353). Piscataway, New Jersey: IEEE Press. 8

Stadler, P. F. (1996). Landscapes and their correlation functions. *J. Math. Chem.*, *20*, 1-45. 29, 30, 59, 144

Stirzaker, D. (1994). *Elementary probability*. Cambridge, UK: Cambridge University Press. 188

Swetina, J., Schuster, P. (1982). Self replicating with error, a model for polynucleotide replication. *Biophys. Chem.*, *16*, 329-340. 49, 146

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In J. D. Schaffer (Ed.), *Proceedings of the third international conference on genetic algorithms and their applications* (p. 2-9). Morgan Kauffman. 151

Tacker, M., Fontana, W., Stadler, P. F., Schuster, P. (1994). Statistics of RNA melting kinetics. *Eur. Biophys J.*, *23*(1), 29-38. 7

Thompson, A. (1996). Silicon evolution. In J. R. Koza, D. E. Goldberg, D. B. Fogel, R. L. Riolo (Eds.), *Genetic programming 1996: Proc. 1st annual conf. (gp96)* (p. 444-452). Cambridge, MA: MIT Press. 8, 113, 144

Thompson, A. (1998). *Hardware evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution*. Springer-Verlag. 113, 144

Thompson, A., Layzell, P. (1999, apr). Analysis of unconventional evolved electronics. *Communications of the ACM*, *28*(4), 71-79. 8

Thompson, A., Layzell, P. (2000). Evolution of robustness in an electronics design. In J. Miller, A. Thompson, P. Thomson, T. Fogarty (Eds.), *Proc. 3rd intl. conf. on evolvable systems (ices2000): From biology to hardware* (Vol. 1801, p. 218-228). Springer-Verlag. 100

Thompson, A., Wasshuber, C. (2000). Design of single-electron systems through artificial evolution. *Int. J. Circ. Theor. Appl.*, *28*(6), 585-599. 8

Wagner, A., Stadler, P. F. (1999). *Viral RNA and evolved mutational robustness* (Technical report No. 99-02-010). Santa Fe Institute. 28, 69, 112, 170

Wagner, G. P., Altenberg, L. (1996). Complex adaptations and the evolution of evolvability. *Evolution*, *50*(3), 967–976. 28

Weinberger, E. D. (1990). Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biol. Cybern.*, *63*, 325-336. 29, 30

Wilke, C. O. (2001). Selection for fitness versus selection for robustness in RNA secondary structure folding. *Evolution*, *55*(12), 2412-2420. 28, 69, 112, 170

Wolpert, D. H., Macready, W. G. (1997, April). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67-82. iv, 3

Wright, M. C., Joyce, G. F. (1997). Continuous in vitro evolution of catalytic function. *Science*,

276, 614-617. 8

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. In D. F. Jones (Ed.), *Proc. 6th intl. congress. on genetics* (Vol. 1, p. 356-366). 1, 103

Wright, S. (1982). Character change, speciation and the higher taxa. *Evolution, 36*, 427-443. 49

Zuker, M., Sankoff, D. (1984). RNA secondary structures and their prediction. *Bull. Math. Biol.*, *46*, 591-621. 7

Appendix A

Evolutionary Operators

Throughout this Appendix we use the following notation and conventions: if A is any set then A^∞ denotes the set of countable sequences $(a_1, a_2, a_3, \ldots)$ of elements of A. For any symbol a, $a^{(1)} = a'$, $a^{(2)} = a''$ and in general $a^{(n)}$ denotes a with n primes; we also adopt the convention that $a^{(0)}$ stands for just a (no prime).

A.1 Definitions

Definition A.1.1. A ***selection operator for population size*** M is a mapping $\mathcal{S}$ from $(\mathbf{R}^\infty)^M$ to the set of jointly distributed random variables on $(\mathbf{N}^\infty)^M$; i.e. for:

$$\omega = \begin{pmatrix} w_1, & w_2, & \ldots, & w_M \\ w'_1, & w'_2, & \ldots, & w'_M \\ w''_1, & w''_2, & \ldots, & w''_M \\ \vdots & \vdots & & \vdots \end{pmatrix} \in (\mathbf{R}^\infty)^M \tag{A.1}$$

with $w_\alpha^{(n)} \in \mathbf{R}$, we have:

$$\mathcal{S}(\omega) = \begin{pmatrix} S_1(\omega), & S_2(\omega), & \ldots, & S_M(\omega) \\ S'_1(\omega), & S'_2(\omega), & \ldots, & S'_M(\omega) \\ S''_1(\omega), & S''_2(\omega), & \ldots, & S''_M(\omega) \\ \vdots & \vdots & & \vdots \end{pmatrix} \tag{A.2}$$

where the $\boldsymbol{S}_\alpha^{(n)}(\omega)$ are *jointly distributed (non-negative) integer-valued random variables*[1]. $\mathcal{S}$ is invariant under re-ordering, in the sense that for any permutation σ of $\{1, 2, \ldots, M\}$ and any $\omega \in (\mathbf{R}^\infty)^M$, $\boldsymbol{s} \in (\mathbf{N}^\infty)^M$:

$$\mathbf{P}\left(\mathcal{S}(\sigma \cdot \omega) = \sigma \cdot \boldsymbol{s}\right) = \mathbf{P}\left(\mathcal{S}(\omega) = \boldsymbol{s}\right) \tag{A.3}$$

where permutations act on the left on $(\mathbf{R}^\infty)^M$ and $(\mathbf{N}^\infty)^M$ in the natural way.

[1] $\mathcal{S}$ may be defined only on a *subset* of $(\mathbf{R}^\infty)^M$; see e.g. fitness proportional selection below.

The intuitive justification for this definition is as follows: if $\boldsymbol{x} = (x_1, x_2, \ldots, x_M) \in \left(\mathcal{A}^L\right)^M$ is a sequence of M sequences on a fitness landscape $\mathcal{L} = (\mathcal{A}, L, f)$ and U a mutation operator on $\mathcal{A}^L$ we can form the random variable $\boldsymbol{x}_U$ taking values in $\left((\mathcal{A}^L)^\infty\right)^M$ by:

$$\boldsymbol{x}_U = \begin{pmatrix} x_1, & x_2, & \ldots, & x_M \\ X'_1, & X'_2, & \ldots, & X'_M \\ X''_1, & X''_2, & \ldots, & X''_M \\ \vdots & \vdots & & \vdots \end{pmatrix} \tag{A.4}$$

where for each $\alpha = 1, 2, \ldots, M$ the $\mathcal{A}^L$-valued random variables $X_\alpha^{(n)}$ are iid as $U(x_\alpha)$ for $n = 1, 2, \ldots$; thus $\boldsymbol{x}_U$ consists, for each α, of the sequence x_α plus countably many (independently generated) mutants $X'_\alpha, X''_\alpha, \ldots$ of x_α. We then define:

$$f(\boldsymbol{x}_U) = \begin{pmatrix} f(x_1), & f(x_2), & \ldots, & f(x_M) \\ f(X'_1), & f(X'_2), & \ldots, & f(X'_M) \\ f(X''_1), & f(X''_2), & \ldots, & f(X''_M) \\ \vdots & \vdots & & \vdots \end{pmatrix} \tag{A.5}$$

so that $f(\boldsymbol{x}_U)$ is a random variable with values in $(\mathbf{R}^\infty)^M$. If $\mathcal{S}(f(\boldsymbol{x}_U)) = \boldsymbol{s} \in (\mathbf{N}^\infty)^M$ and (for each α, n) we have $X_\alpha^{(n)} = x_\alpha^{(n)} \in \mathcal{A}^L$, then $s_\alpha^{(n)}$ represents the number of copies of sequence $x_\alpha^{(n)}$ to be selected for a new population:

Definition A.1.2. The ***evolutionary operator*** $\mathcal{G} = \mathcal{G}(\mathcal{S}, U)$ associated with the selection operator $\mathcal{S}$ and the mutation operator U is the mapping that takes a population $\boldsymbol{x} \in \mathcal{P}^M(\mathcal{A}^L)$ to the new (random) population $\mathcal{G}(\boldsymbol{x})$ formed as follows: conditional on $\mathcal{S}(f(\boldsymbol{x}_U)) = \boldsymbol{s} \in (\mathbf{N}^\infty)^M$, for each $\alpha = 1, 2, \ldots, M$ we select:

$$\begin{array}{ccc} s_\alpha & \text{copies of} & x_\alpha \\ s'_\alpha & \text{copies of} & X'_\alpha \\ s''_\alpha & \text{copies of} & X''_\alpha \\ & \textit{etc.} & \end{array} \tag{A.6}$$

The invariance of $\mathcal{S}$ under re-ordering as according to Eq. (A.3) guarantees that $\mathcal{G}(\boldsymbol{x})$ is well-defined. $\mathcal{G}(\mathcal{S}, U)$ thus maps populations of size M on $\mathcal{A}^L$ to the set of random variables taking values in the set $\mathcal{P}(\mathcal{A}^L)$ of populations on $\mathcal{A}^L$.

We have, however, disregarded two important points: firstly, we have not guaranteed that Eq. (A.6) will select only a *finite* number of sequences for the new population. Secondly (and somewhat less obviously), we would like to ensure that only a finite number of *fitness evaluations* are required to generate a new population. We address these issues in the following manner: let us say that a (real-valued) function $\phi(\omega)$ defined on $(\mathbf{R}^\infty)^M$ ***depends on*** (α, n) iff there exist $\omega_1, \omega_2 \in (\mathbf{R}^\infty)^M$ with $(w_1)_\beta^{(m)} = (w_2)_\beta^{(m)}$ for all β, m *except* $\beta = \alpha$ and $m = n$, and such that $\phi(\omega_1) \neq \phi(\omega_2)$. Stated more simply, $\phi(\omega)$ depends on (α, n) iff $w_\alpha^{(n)}$ appears in the expression for $\phi(\omega)$ - it is necessary to know the value of the (α, n)-th entry of ω in order to calculate $\phi(\omega)$.

Now given $\boldsymbol{s} \in (\mathbf{N}^\infty)^M$ we have the real-valued function: $\omega \mapsto \mathbf{P}(\mathcal{S}(\omega) = \boldsymbol{s})$. Suppose we have a population $\boldsymbol{x}$ and a mutation operator U. Intuitively, if the function $\mathbf{P}(\mathcal{S}(\omega) = \boldsymbol{s})$ depends

on (α, n) then we need to know $f\left(X_\alpha^{(n)}\right)$ in order to establish whether the number of copies of sequences and mutants required according to Eq. (A.6) is given by $\boldsymbol{s}$; that is, *we need to evaluate the fitness of the n-th mutant of* x_α.

Furthermore, we *always* evaluate fitness of a new mutant if it is selected for inclusion in a population, regardless of whether we actually *need* to know its fitness in order to select it. Thus For $\boldsymbol{s} \in (\mathbf{N}^\infty)^M$ we define the Boolean:

$$\varepsilon_\alpha^{(n)}(\boldsymbol{s}) = \left(s_\alpha^{(n)} > 0 \; or \; \mathbf{P}\left(\mathcal{S}(\omega) = \boldsymbol{s}\right) \text{ depends on } (\alpha, n)\right) \tag{A.7}$$

for $\alpha = 1, 2, \ldots, M,\ n = 1, 2, \ldots$ (note that the *or* on the RHS above is *inclusive*). Intuitively, $\varepsilon_\alpha^{(n)}(\boldsymbol{s})$ is *true* iff the fitness of the n-th mutant of x_α needs to be evaluated when selecting a new population by $\boldsymbol{s}$ - either because that mutant is itself to be selected for the new population, or because its fitness needs to be evaluated to decide which sequences to select - or both. Note that we do not define $\varepsilon_\alpha^{(n)}(\boldsymbol{s})$ for $n = 0$; it is assumed that the fitnesses of sequences in the *current* population are always known. We also impose the restriction on $\mathcal{S}$ that for any $\boldsymbol{s}$ and for $\alpha = 1, 2, \ldots, M$:

$$\varepsilon_\alpha^{(n)}(\boldsymbol{s}) \Rightarrow \varepsilon_\alpha^{(m)}(\boldsymbol{s}) \quad \text{for } 1 \leq m < n \tag{A.8}$$

That is, if a mutant needs to be evaluated for fitness, then it is implicit that all "prior" mutants of the same sequence also needed to be evaluated - mutants don't have to be "generated" unless they are actually needed! We may thus think of mutants as being generated sequentially "as required" by the selection operator.

Now for $\omega \in (\mathbf{R}^\infty)^M$ we define the ***(target) population size*** of $\mathcal{S}(\omega)$ to be the random variable:

$$|\mathcal{S}(\omega)| = \sum_{\alpha=1}^{M} \sum_{n=0}^{\infty} S_\alpha^{(n)}(\omega) \tag{A.9}$$

and the ***number of fitness evaluations*** of $\mathcal{S}(\omega)$ to be the random variable:

$$\|\mathcal{S}(\omega)\| = \sum_{\alpha=1}^{M} \sum_{n=1}^{\infty} \varepsilon_\alpha^{(n)}\left(\mathcal{S}(\omega)\right) \tag{A.10}$$

We wish to ensure that these random variables are finite. We thus impose the requirement on any selection operator $\mathcal{S}$ that:

$$\forall \boldsymbol{s}\ \exists n(\boldsymbol{s}) \text{ such that } n > n(\boldsymbol{s}) \Rightarrow \forall \alpha,\ \varepsilon_\alpha^{(n)}(\boldsymbol{s}) = false \tag{A.11}$$

This condition says that for any given $\boldsymbol{s}$ - i.e. whichever combination of copies of original sequences and new mutants are actually selected to form the new population - only a finite number of mutants need to be evaluated for fitness. Then, since mutants are only added to a new population if they have been evaluated for fitness, this implies in turn that the new population is finite. We have (a.s.) for any ω; i.e. for any sequence/mutant fitnesses:

$$|\mathcal{S}(\omega)| < \infty \tag{A.12}$$

$$\|\mathcal{S}(\omega)\| < \infty \tag{A.13}$$

We shall say that the selection operator $\mathcal{S}$ (of population size M) is of ***fixed population size*** M iff $\mathbf{P}\left(|\mathcal{S}(\omega)| = M\right) = 1$ for any ω. We shall say that it has a ***fixed number of fitness evaluations*** r iff $\mathbf{P}\left(\|\mathcal{S}(\omega)\| = r\right) = 1$ for any ω.

A further subtlety is the following: in order to measure the performance of evolutionary processes, we will want to know the *fittest* sequence in a population, in particular so as to maintain a record of "fittest-sequence-so-far-discovered" during an optimisation run (*cf.* Section 3.5). However, it is conceivable that in creating a new population via some evolutionary operator, a mutant of higher fitness than any sequence in the current population is evaluated - but then not selected for the new population! We consider that in this (unlikely) scenario the "transient" fit mutant should be recorded as a candidate for fittest-so-far; it has, after all, been created and evaluated for fitness. Therefore, to maintain a best-so-far record we cannot simply note the fittest mutant in a new population; we should also include (possibly fitter) transient mutants. This motivates the following: for $\omega \in (\mathbf{R}^\infty)^M$ let define the ***best fitness*** of $\mathcal{S}(\omega)$ to be the r.v.:

$$\mathcal{S}(\omega)^* = \max_{\alpha,n} \left\{ w_\alpha^{(n)} \;\middle|\; S_\alpha^{(n)}(\omega) > 0 \right\} \tag{A.14}$$

i.e. the best fitness of the new population. We then define the ***best evaluated fittness*** of $\mathcal{S}(\omega)$ to be the random variable:

$$[\mathcal{S}(\omega)] = \max \left\{ \mathcal{S}(\omega)^*, \max_{\alpha,n} \left\{ w_\alpha^{(n)} \;\middle|\; \varepsilon_\alpha^{(n)}(\mathcal{S}(\omega)) = \mathit{true} \right\} \right\} \tag{A.15}$$

That is, $[\mathcal{S}(\omega)]$ is the fitness of the fittest out of the new population *and any mutant evaluated in creating the new population*. In general there would not seem to be much reason *not* to select the fittest mutant discovered during creation of a new population, so that (certainly for all selection operators we shall consider) we have simply[2] $[\mathcal{S}(\omega)] = \mathcal{S}(\omega)^*$.

We extend our definitions also to evolutionary operators: for an evolutionary operator $\mathcal{G} = \mathcal{G}(\mathcal{S}, U)$ based on the selection operator $\mathcal{S}$ of population size M and any $\boldsymbol{x} \in \mathcal{P}^M(\mathcal{A}^L)$ we define the random variables $|\mathcal{G}(\boldsymbol{x})| = |\mathcal{S}(f(\boldsymbol{x}_U))|$, $\|\mathcal{G}(\boldsymbol{x})\| = \|\mathcal{S}(f(\boldsymbol{x}_U))\|$, $[\mathcal{G}(\boldsymbol{x})] = [\mathcal{S}(f(\boldsymbol{x}_U))]$ and $[\mathcal{G}(\boldsymbol{x})] = [\mathcal{S}(f(\boldsymbol{x}_U))]$ where the random variable $f(\boldsymbol{x}_U)$ is given by Eq. (A.5). Note that $\mathcal{S}(f(\boldsymbol{x}_U))^*$ is just $f^*(\mathcal{G}(\boldsymbol{x}))$.

The formalism is perhaps clarified by a simple example:

Example A.1.1. Consider the following procedure for forming a new population from an existing population of size $M = 1$: suppose the current population comprises a single sequence x with fitness $f(x) = w$, say. We generate a mutant $x' = U(x)$ with fitness $f(x') = w'$. If $w' > w$ we replace x by x' so that our new population - the "next generation" - comprises the single sequence x'. If $w' \leq w$, we repeat the procedure: let $x'' = U(x)$ be *another* mutant of x (not of x' !) with fitness $f(x'') = w''$. If now $w'' > w$ we replace x by x'' so that the new population comprises just the sequence x''. If, however, $w'' \leq w$ we give up and retain x; the new population is the same as the old, comprising just the original sequence x.

The selection operator $\mathcal{S}(w, w', w'')$ is evidently of fixed population size 1 and there are three possible outcomes of selection, which we label $\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3$:

$$\begin{array}{lllll} \boldsymbol{s}_1: & s = 1 & s' = 0 & s'' = 0 \\ \boldsymbol{s}_2: & s = 0 & s' = 1 & s'' = 0 \\ \boldsymbol{s}_3: & s = 0 & s' = 0 & s'' = 1 \end{array} \tag{A.16}$$

[2] Note that this is not the same as saying that selection is *elitist* (see below).

We then find:

$$\begin{aligned}\mathbf{P}\left(\mathcal{S}(w,w',w'')=\boldsymbol{s}_1\right) &= h(w-w')\,h(w-w'')\\ \mathbf{P}\left(\mathcal{S}(w,w',w'')=\boldsymbol{s}_2\right) &= h(w'-w)\\ \mathbf{P}\left(\mathcal{S}(w,w',w'')=\boldsymbol{s}_3\right) &= h(w-w')\,h(w''-w)\end{aligned} \tag{A.17}$$

where $h(z)$ is the step function:

$$h(z)=\begin{cases}1 & z>0\\ 0 & z\leq 0\end{cases} \tag{A.18}$$

Thus $\mathbf{P}\left(\mathcal{S}(w,w',w'')=\boldsymbol{s}_2\right)$ depends on just w,w' while the other selection probabilities depend on w,w',w''. We thus have from Eq. (A.7):

$$\begin{aligned}&\varepsilon'(\boldsymbol{s}_1)=\varepsilon''(\boldsymbol{s}_1)=true\\ &\varepsilon'(\boldsymbol{s}_2)=true,\ \ \varepsilon''(\boldsymbol{s}_2)=false\\ &\varepsilon'(\boldsymbol{s}_3)=\varepsilon''(\boldsymbol{s}_3)=true\end{aligned} \tag{A.19}$$

Thus either 1 or 2 fitness evaluations need be performed and from Eq. (A.10) using Eq. (A.17) and Eq. (A.19) we may calculate the distribution of the number of fitness evaluations to be:

$$\begin{aligned}\mathbf{P}\left(\left\|\mathcal{S}(w,w',w'')\right\|=1\right) &= h(w'-w)\\ \mathbf{P}\left(\left\|\mathcal{S}(w,w',w'')\right\|=2\right) &= h(w-w')\end{aligned}$$

which states (the obvious) that we only need evaluate one mutant if the first mutant is fitter than the original sequence, otherwise we need to evaluate a second mutant. Similarly, we can calculate the probability that $[\mathcal{S}(w,w',w'')]$ is equal to w, w' or w''.

We remark that the only reason for demanding that a selection operator comprise a *(countable) infinity* of selections is to allow for the possibility of an arbitrary (albeit finite) number of fitness evaluations or target population size. In practice, almost all selection operators we shall encounter will have a fixed number of fitness evaluations and be of fixed population size, so that a derived evolutionary operator maps from $\mathcal{P}^M(\mathcal{A}^L)$ into the set of random variables on $\mathcal{P}^M(\mathcal{A}^L)$ for some population size $M>0$. Of course for biological selection we might not get away with this restriction.

We now relate our definition of selection to some familiar properties of evolutionary operators. $\mathcal{S}$ is ***fitness-proportional*** if for any real $c>0$ and $\omega\in(\mathbf{R}^\infty)^M$ we have $\mathcal{S}(c\cdot\omega)=\mathcal{S}(\omega)$ where $c\cdot\omega$ denotes the element of $(\mathbf{R}^\infty)^M$ specified by multiplying each component $w_\alpha^{(n)}$ of ω by c. Fitness-proportional selection operators are only defined for non-negative fitness. The selection operator $\mathcal{S}$ is said to be ***ranked*** if $rank(\omega)=rank(\omega')\Rightarrow\mathcal{S}(\omega)=\mathcal{S}(\omega')$; i.e. if selection probabilities depend only on the rank ordering of fitness components $w_\alpha^{(n)}$. $\mathcal{S}$ has ***discrete generations*** (or is ***generational***) iff $S_\alpha(\omega)=S_\alpha^{(0)}(\omega)=0\,\forall\omega,\ \forall\alpha$ (a.s.); i.e. no current sequences survive into the new population, which is made up entirely of new mutants - otherwise $\mathcal{S}$ has ***overlapping generations*** (or is ***steady-state***). Finally, $\mathcal{S}$ is ***elitist*** iff $\mathcal{S}(\omega)^*\geq\max\{w_\alpha\}$; that is, best fitness never decreases[3].

[3]Elitism is sometimes defined by the property that some existing sequence of current best fitness is always selected. By our definition, if a mutant of fitness *greater than or equal to* the current best fitness is selected, then we need not select an existing sequence of current best fitness.

A.2 Examples

For completeness we demonstrate how the example selection operators of Chapter 3 may be expressed by the above formalism. In the following, $\omega = \left(w_\alpha^{(n)}\right)$ and $\boldsymbol{w}$ denotes the current population fitnesses $(w_1, w_2, \ldots, w_M)$.

Example A.2.1. ***Birth-and-death selection:*** for each $\boldsymbol{w} \in \mathbf{R}^M$ there are jointly distributed random variables $B(\boldsymbol{w})$ and $D(\boldsymbol{w})$ taking values in $1, 2, \ldots, M$ such that for $\alpha = 1, 2, \ldots, M$:

1. $S_\alpha(\omega) = \begin{cases} 0 & \alpha = D(\boldsymbol{w}) \\ 1 & \text{otherwise} \end{cases}$
2. $S'_\alpha(\omega) = \begin{cases} 1 & \alpha = B(\boldsymbol{w}) \\ 0 & \text{otherwise} \end{cases}$
3. $S''_\alpha(\omega) = S'''_\alpha(\omega) = \ldots = 0$

Intuitively, the sequence $x_{B(\boldsymbol{w})}$ replicates, while the sequence $x_{D(\boldsymbol{w})}$ dies and is replaced by the new mutant $U\left(x_{B(\boldsymbol{w})}\right)$. Note that we might have $D(\boldsymbol{w}) = B(\boldsymbol{w})$; i.e. the same sequence is chosen to replicate and die. To ensure invariance under re-ordering we also require that, for σ a permutation of $1, 2, \ldots, M$, we have $B(\sigma \cdot \boldsymbol{w}) = \sigma(B(\boldsymbol{w}))$ and $D(\sigma \cdot \boldsymbol{w}) = \sigma(D(\boldsymbol{w}))$.

Example A.2.2. ***Moran selection:*** Without regard to waiting times between events, a selection operator $\mathcal{S}$ describing the process may be constructed as follows: let $Q(\boldsymbol{w})$ be a Boolean random variable with $\mathbf{P}(Q(\boldsymbol{w})) = q(\boldsymbol{w})$ and let $B(\boldsymbol{w})$ and $D(\boldsymbol{w})$ be random variables on $\{1, 2, \ldots, M\}$ with $\mathbf{P}(B(\boldsymbol{w}) = \alpha) = \lambda_\alpha(\boldsymbol{w})/\lambda(\boldsymbol{w})$ and $\mathbf{P}(D(\boldsymbol{w}) = \alpha) = \mu_\alpha(\boldsymbol{w})/\mu(\boldsymbol{w})$. Then $\mathcal{S}$ is given by:

1. $S_\alpha(\omega) = 1$
2. $S'_\alpha(\omega) = \begin{cases} 1 & \alpha = B(\boldsymbol{w}) \\ 0 & \text{otherwise} \end{cases}$
3. $S''_\alpha(\omega) = S'''_\alpha(\omega) = \ldots = 0$

for $\alpha = 1, 2, \ldots, M$ if $Q(\boldsymbol{w}) = \mathit{true}$ and by:

1. $S_\alpha(\omega) = \begin{cases} 0 & \alpha = D(\boldsymbol{w}) \\ 1 & \text{otherwise} \end{cases}$
2. $S'_\alpha(\omega) = S''_\alpha(\omega) = \ldots = 0$

for $\alpha = 1, 2, \ldots, M$ if $Q(\boldsymbol{w}) = \mathit{false}$. As noted, Moran selection is not of fixed population size or fixed number of fitness evaluations - we have:

$$\begin{aligned} \mathbf{P}(|\mathcal{S}(\omega)| = M+1) &= Q(\boldsymbol{w}) \\ \mathbf{P}(|\mathcal{S}(\omega)| = M) &= 1 - Q(\boldsymbol{w}) \end{aligned}$$

for population size M and:

$$\begin{aligned} \mathbf{P}(\|\mathcal{S}(\omega)\| = 1) &= Q(\boldsymbol{w}) \\ \mathbf{P}(\|\mathcal{S}(\omega)\| = 0) &= 1 - Q(\boldsymbol{w}) \end{aligned}$$

Example A.2.3. ***Multinomial selection:*** We define $\mathcal{S}(\omega)$ by:

$$S'_\alpha(\omega) = S''_\alpha(\omega) = \ldots = S_\alpha^{(R_\alpha(\boldsymbol{w}))}(\omega) = 1 \tag{A.20}$$

for $\alpha = 1,2,\ldots,M$, with all other $S_\alpha^{(n)}(\omega) = 0$, where the $R_\alpha(\boldsymbol{w})$ are given by Eq. (3.9). We note that re-ordering invariance requires that the selection probabilities $p_\alpha(\boldsymbol{w})$ satisfy $\boldsymbol{p}(\sigma \cdot \boldsymbol{w}) = (\sigma \cdot \boldsymbol{p})(\boldsymbol{w})\ \forall \boldsymbol{w}, \sigma$.

Example A.2.4. ***Stochastic hill-climber:*** Let $\omega = (w, w', w'', \ldots)$. $\mathcal{S}$ is then given by:

$$S'(\omega) = \begin{cases} 1 & Y_\alpha(w,w') = true \\ 0 & \text{otherwise} \end{cases} \tag{A.21}$$

$$S(\omega) = {}^\sim S'(\omega) \tag{A.22}$$

where $Y(w,w')$ is the Bernoulli random variable of Example 3.2.5. All other selections are 0.

Example A.2.5. ***Multiple independent stochastic hill-climbers:***

$$S'_\alpha(\omega) = \begin{cases} 1 & Y_\alpha(w_\alpha, w'_\alpha) = true \\ 0 & \text{otherwise} \end{cases} \tag{A.23}$$

$$S_\alpha(\omega) = {}^\sim S'_\alpha(\omega) \tag{A.24}$$

for $\alpha = 1,2,\ldots,M$, all other selections being 0.

A.3 "Lifting" the Selection Operator

Suppose we are given a neutral partitioning $\mathcal{A}^L = \bigcup_{i=1}^N \Gamma_i$ of $\mathcal{L}$ and a (compatible) mutation operator U. We proceed as follows: for each network index i let the (independent) r.v. X_i be *uniform random* on neutral network Γ_i. Let $\boldsymbol{i} =< i_1, i_2, \ldots, i_M > \in \mathcal{P}^M(\widetilde{\mathcal{A}^L})$ be a population of neutral network indices. We then define the random variable $\boldsymbol{i}_U$ with values in $\left((\mathcal{A}^L)^\infty\right)^M$ (*cf.* Eq. (A.4)) to be:

$$\boldsymbol{i}_U = \begin{pmatrix} i_1, & i_2, & \ldots, & i_M \\ I'_1, & I'_2, & \ldots, & I'_M \\ I''_1, & I''_2, & \ldots, & I''_M \\ \vdots & \vdots & & \vdots \end{pmatrix} \tag{A.25}$$

where for each $\alpha = 1,2,\ldots,M$ the r.v.'s $I_\alpha^{(n)}$ (which take values in $\widetilde{\mathcal{A}^L}$) are iid as $\widetilde{U}(X_{i_\alpha})$. Since by definition fitness is the same for all sequences in a neutral network we may, as in Eq. (A.5), form the r.v. $f(\boldsymbol{i}_U)$ which takes values in $(\mathbf{R}^\infty)^M$.

Again, if $\mathcal{S}(f(\boldsymbol{i}_U)) = \boldsymbol{s} \in (\mathbf{N}^\infty)^M$ and (for each α, n) $I_\alpha^{(n)} = i_\alpha^{(n)} \in \widetilde{\mathcal{A}^L}$ then $s_\alpha^{(n)}$ represents the number of copies of index $i_\alpha^{(n)}$ to be selected for a new population of network indices. Thus, as in Def. A.1.2, the "lifted" evolutionary operator $\tilde{\mathcal{G}}$ associated with selection operator $\mathcal{S}$, mutation operator U and the given neutral partitioning is the mapping that takes a population $\boldsymbol{i} \in \mathcal{P}^M(\widetilde{\mathcal{A}^L})$ to the new (random) population $\tilde{\mathcal{G}}(\boldsymbol{i}) \in \mathcal{P}(\widetilde{\mathcal{A}^L})$ formed (conditional on $\mathcal{S}(f(\boldsymbol{i}_U)) = \boldsymbol{s}$) by selecting, for each $\alpha = 1,2,\ldots,M$:

$$\begin{array}{lll} s_\alpha & \text{copies of} & i_\alpha \\ s'_\alpha & \text{copies of} & I'_\alpha \\ s''_\alpha & \text{copies of} & I''_\alpha \\ & etc. & \end{array} \tag{A.26}$$

It is clear that the above definition is invariant with respect to re-ordering of the i_α and thus well-defined on populations. We note that mutation enters our definition solely through the *mutation matrix* for the partitioning, since $\mathbf{P}\left(\widetilde{U}(X_j) = i\right) = \boldsymbol{m}_{ij}(U)$ by definition.

Appendix B

Transformations of the Quasi-species Generating Functions

B.1 Transformation of $g(z)$ by Mutation

From (7.9) we have, for fixed j, for any z and fixed $U \equiv Lu$:

$$\begin{aligned}
\sum_i m_{ij}(1-z)^i &= \sum_{\alpha,\beta} \binom{j}{\alpha}\binom{L-j}{\beta} u^{\alpha+\beta}(1-u)^{L-(\alpha+\beta)}(1-z)^{j-\alpha+\beta} \\
&= \sum_{\alpha} \binom{j}{\alpha} u^{\alpha}(1-u)^{j-\alpha}(1-z)^{j-\alpha} \\
&\qquad \times \sum_{\beta} \binom{L-j}{\beta} u^{\beta}(1-z)^{\beta}(1-u)^{L-j-\beta} \\
&= (1-z+uz)^j(1-uz)^{L-j} \\
&= \left(1-\frac{1}{L}Uz\right)^L \left(\frac{1-z+\frac{1}{L}Uz}{1-\frac{1}{L}Uz}\right)^j \\
&\to e^{-Uz}(1-z)^j \quad \text{as } L\to\infty
\end{aligned}$$

where in the last step we have used $\left(1-\frac{1}{L}Uz\right)^L \to e^{-Uz}$ as $L\to\infty$. The result follows immediately.

B.2 Transformation of $g(z)$ by Recombination

Let us set:

$$c_{jk,\alpha} \equiv \binom{j}{\alpha}\binom{L-j}{k-\alpha}\binom{L}{k}^{-1} \tag{B.1}$$

Note that $c_{jk,\alpha}$ is symmetric in j,k. Then from (7.18) we have:

$$r_{ijk} = \sum_{\alpha} c_{jk,\alpha}\binom{j+k-2\alpha}{i-\alpha}\left(\frac{1}{2}\right)^{j+k-2\alpha} \tag{B.2}$$

Now using Stirling's formula (Stirzaker, 1994) it is not difficult to show that, holding j,k and α fixed we have $\lim_{L\to\infty} c_{jk,\alpha} = \delta_{\alpha 0}$. Thus, holding i, j and k fixed, we have:

$$\lim_{L\to\infty} r_{ijk} = \binom{j+k}{i}\left(\frac{1}{2}\right)^{j+k} \tag{B.3}$$

(*cf.* (Kimura & Maruyama, 1966) - this is equivalent to neglecting the probability of *homozygous* mutant alleles occurring at any locus during recombination, which is a reasonable approximation for long sequence lengths). In the limit:

$$\sum_i r_{ijk}(1-z)^i = (1-\tfrac{1}{2}z)^{j+k} \tag{B.4}$$

and the result follows.