Emergence, Complexity and Computation ECC

Hendrik Richter
Andries P. Engelbrecht *Editors*

Recent Advances in the Theory and Application of Fitness Landscapes

Springer

Emergence, Complexity and Computation

Volume 6

For further volumes:
http://www.springer.com/series/10624

About this Series

“The Emergence, Complexity and Computation (ECC) series publishes new developments, advancements and selected topics in the fields of complexity, computation and emergence. The series focuses on all aspects of reality-based computation approaches from an interdisciplinary point of view especially from applied sciences, biology, physics, or Chemistry. It presents new ideas and interdisciplinary insight on the mutual intersection of subareas of computation, complexity and emergence and its impact and limits to any computing based on physical limits (thermodynamic and quantum limits, Bremermann’s limit, Seth Lloyd limits...) as well as algorithmic limits (Gödel’s proof and its impact on calculation, algorithmic complexity, the Chaitin’s Omega number and Kolmogorov complexity, non-traditional calculations like Turing machine process and its consequences,...) and limitations arising in artificial intelligence field. The topics are (but not limited to) membrane computing, DNA computing, immune computing, quantum computing, swarm computing, analogic computing, chaos computing and computing on the edge of chaos, computational aspects of dynamics of complex systems (systems with self-organization, multiagent systems, cellular automata, artificial life,...), emergence of complex systems and its computational aspects, and agent based computation. The main aim of this series it to discuss the above mentioned topics from an interdisciplinary point of view and present new ideas coming from mutual intersection of classical as well as modern methods of computation. Within the scope of the series are monographs, lecture notes, selected contributions from specialized conferences and workshops, special contribution from international experts, as well as selected PhD theses."

Hendrik Richter · Andries P. Engelbrecht
Editors

Recent Advances in the Theory and Application of Fitness Landscapes

Springer

Editors
Hendrik Richter
HTWK Leipzig University of Applied Sciences
Faculty of Electrical Engineering and Information Technology
Leipzig
Germany

Andries P. Engelbrecht
Department of Computer Science
University of Pretoria
Pretoria
South Africa

ISSN 2194-7287 ISSN 2194-7295 (electronic)
ISBN 978-3-642-41887-7 ISBN 978-3-642-41888-4 (eBook)
DOI 10.1007/978-3-642-41888-4
Springer Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my daughter Malena Charlotte, who was conceived, carried, and born while this book was being prepared – Hendrik Richter

To Anneli Carien, who came to this world while the book was being prepared – Andries Engelbrecht

Foreword:
Statable and Non–prestatable Fitness Landscapes

Stuart Kauffman

It is an honor to be asked to write a foreword for this fine book on recent advances in the theory and application of fitness landscapes. The topic is vast, the book a superb review of much of its current status. My task is not to present a précis of the book itself, but, I hope, to place its topics in an even wider context as I, with my own limited biases, see that context.

Topic 1

The first context is that of this book. We are confronted, typically, with a well formulated continuous or discrete complex combinatorial optimization problem of a fixed fitness landscape and seek a search algorithm to find good optima or even the global optima. Here the landscape constitutes a fixed "potential function" whose peaks or valleys are the desired solutions. Derived from this, as discussed so well in this book, are co-evolutionary problems in which two or more landscapes are coupled, agents on each landscape making adaptive moves and thereby deforming the other landscape. Such systems are general dynamical systems and known to exhibit, as discussed in this book, two modes of behavior, one in which the agents reach mutually consistent local optima and the system stops changing. In the other, as agents move, their landscapes deform even more rapidly, resulting in chaotic behavior, sometimes called the Red Queen effect. Between these two regimes lies a "critical" phase transition. In general, these models are a subset of game theory in which each agent can, typically, only change to neighboring strategies in a genetically encoded strategy space. The mutually consistent local peak solution is the generalization of pure strategy Nash equilibria from games where any move can be made in strategy space, to those in which only neighboring moves can be made. Past work by this author has given initial results that optimal solutions are at least sometimes found at the critical phase transition [1]. Landscapes can occur without and with neutrality. The statistical structure of such landscapes is a major topic of concern, and with it, use of measures of that statistical structure, such as landscape correlation lengths, Stadler et al.'s "reachability topology" (as discussed in Chapter 6 of this book) and others, to attempt to predict how hard learning or adaptation may be on fixed, or, harder, co-evolving landscapes. Beyond this, in this book are efforts to consider time varying landscapes that may vary in stochastic or non-stochastic ways. This latter topic will become of major concern below in this foreword. The well known no free lunch theorem suggests that averaged over all landscapes, no search algorithm outperforms any other algorithm. This book contains efforts noted above to study the structure of a given landscape to choose algorithms that may be better than average or even very good on a given landscape. The problem is stated to be insoluble in general, but sometimes good algorithms can be found.

Topic 2

A second fundamental issue, which is outside the scope of this book, is the kinds of problems, the kinds of systems we or, say evolution, seeks to optimize, to generate what kinds of landscapes? Here little work, to my knowledge has been done. But it is of fundamental importance. I present a brief conjecture that some problems are not solvable by any local search algorithm in less than exponential time. Consider the shortest algorithm to produce a given output on a universal computer. Let its (unprovable) length be N. Gregory Chaitin has shown on the N dimensional Boolean hypercube there is on order of a single vertex that constitutes this program, where the binary string at that vertex is the program for the computer. Chaitin then shows that if one considers programs of length $N, N+1, N+2, \ldots, N+C$, there are in the order of 1, 2, 4,..., 2^C vertices on the hypercube that satisfy the requirement to be a program of that length that solves the problem. Now my conjecture, perhaps able to become a theorem: Take a fixed length input string and choose the single correct vertex on the N dimensional Boolean hypercube where only that single vertex yields the proper output on that input. Run the randomly chosen input string on the "correct" vertex and observe the output string taken as the proper solution. Now choose the N "one mutant neighboring points" to the "correct" vertex, and run the input string on each of these to obtain an output string. Use normalized compression distance to compare the correct output string and the output string of a given 1 mutant neighbor of the correct vertex. Normalized compression distance, *NCD*, a universal measure of, essentially, the mutual information between these two strings. Use $1-NCD$ as a measure of the "fitness" of the program at that one mutant neighbor of the correct vertex. Do this for all vertices on the N dimensional hypercube to obtain a "fitness landscape" of the fitness of each program, encoded at each vertex, to generate an output similar or dissimilar to the correct output. My bet is that for the minimal length program, length N, the resulting fitness landscape is *random*. That is, I bet, to be proven, that the fitness at neighboring vertices are random in value. It is known that such landscapes have on the order of $\frac{2^N}{N+1}$ local optima. Thus finding the global optimum with the correct minimal program is NP hard, requiring search of the entire space or at least, as the space is exponential in N, a fixed fraction of this exponentially large space; hence, for large N, not solvable in less than exponential time. My further guess is that if one started with a redundant program, longer than N, i.e. $N+C$, due to that redundancy, the fitness landscape on the $N+C$ Boolean hypercube, it would be a correlated landscape whose correlation length increases as C increases. My further bet is that no procedure can start with an $N+C$ landscape and evolve to ever smaller C approaching $C=0$ to achieve the minimal program, because as C decreases, the landscape becomes less correlated then uncorrelated. My guess is that as C decreases, correlations on the landscape may be related such that the evolving system is led into a region of the N dimensional space close to the correct vertex, but on the $C=0$ random landscape, no local search algorithm is likely to find the nearby correct solution before wandering off on the random landscape away from the correct solution. “Likely” may be quantifiable.

Topic 3

This discussion suffices to relate some problems, here finding the shortest algorithm to solve a problem and the structure of the induced fitness landscape. More it relates to the next issue, for the above problem, in my conjecture, cannot be solved by mutation and selection alone, and perhaps not by recombination, mutation and selection together, for recombination does not work on random landscapes [2]. If so, no evolutionary process in biology can evolve the shortest program. All this needs to become a set of theorems, relating to the issue of what problems induce what landscapes and why.

1. Do biological and economic and other evolutionary processes “tune” the very structure of the fitness landscapes upon which they evolve? This book does not address this topic. In [1], I had a first try at the problem, and believe I showed a model in which, with no group selection, “organisms” co-evolving on landscapes and invading one another’s niches could evolve the structure of the fitness landscapes upon which they evolved. In this model, an invading species, if successful in a new niche, carried with it the ruggedness of its own landscape. Thus landscape ruggedness itself becomes an evolving feature of the total evolving system, and landscape ruggedness itself can evolve. In this model, the system evolves from Red Queen, and from the stationary Nash equilibrium regime to the critical phase transition between the two, and, on average, the life time of species increases and their fitness increases. These results suggest that an evolutionary process can, in fact, evolve the very structure and couplings among co-evolving agents to the long term benefit of all the currently evolving agents. This topic remains very unexplored but is likely to be of major importance. It suggests that biological and economic evolution are “tuning” the statistical structure of the very problems and thus fitness landscapes over which they evolve to become more “evolvable” and “solvable”. Much remains to be learned.
2. Adaptive evolution in the biosphere and economy and elsewhere is probably not only occurring on time deforming and stochastically time deforming landscapes, it is far worse: Not only can we not characterize the stochastic process by which landscapes change, we cannot even prestate the possibility space, that is

the ever changing phase space of these evolutionary processes, hence we can neither mathematize actual evolution of the biosphere, biosphere and ecology, or the economy or culture or law. And even if we could write down laws of motion for this evolution, we would lack foreknowledge of the "niche" boundary conditions and so could not integrate the equations we do not have anyway. In short, if what I say with my colleagues, Giuseppe Longo, Mael Montevil, and myself [4, 5], is correct after careful examination, *no laws* entail the evolution of the biosphere or, a fortiori the economy or probably many aspects of life. Not only do we not know what *will* happen, we do not even know what *can* happen. Thus we cannot formulate a probability distribution over what "can happen" for we do not know the sample space beforehand. Therefore we cannot formulate a stochastic model of changing landscapes for, as we will see next, we cannot prestate the variables that will become relevant, so we do not know the space over which to formulate a fitness landscape.

Topic 4

I discuss the fourth topic in several sections below.

1. The Non Ergodic Universe Above the Level of Atoms
   Has the universe created in its 13.7 billion years all known atoms? Yes. But now consider proteins made of 20 kinds of amino acids strung together in a linear sequence by peptide bonds. A typical biological protein has a length of 300 amino acids. Consider, then, all possible proteins length 20 amino acids. There are 20^{200} or about 10^{260} such possible proteins. Now the universe has about 10^{80} particles. Its fastest time scale is the Planck time scale of 10^{-43} seconds. Ignoring space–like separation, if all the universe were doing in the past 13.7 billion years was constructing, in parallel, different proteins of length 200, it would require the current age of the universe raised to the 37th power to construct all possible proteins of length 200 just *once*. This has physical meaning. At levels above the atom in complexity, the universe is on a unique trajectory that cannot become ergodic in the lifetimes of many universes. Thus, most complex things will never exist, so those complex things that get to exist have a special status.

2. Kantian Wholes
   Kant said that in an "organized being the parts exist for and by means of the parts", that is, the whole exists by means of the parts and the parts exist by means of the whole. He was thinking of organisms.

3. Collectively Autocatalytic Peptide Sets as Minimal Kantian Wholes
   Gonen Ashkenasy [6] at the Ben Gurion University, has a set of 9 peptides, each of which catalyzes the formation of the next peptide by ligating two fragments of that peptide, around a 9 peptide circle of peptides. The set as a *whole* is collectively autocatalytic. Note that no peptide catalyzes its own formation; the set as a whole collectively catalyses its own formation. As a side comment, Ashkenasy's results demonstrate conclusively that molecular reproduction does not depend

upon DNA– or RNA–like template replication. Now, calling catalysis of a reaction a “catalytic task”, the 9 peptide collectively autocatalytic set achieves a *task closure*. All the reactions that must be catalyzed from within the set, are catalyzed from within the set. The system is also an open thermodynamic one deriving food from the two fragments of each peptide maintained at constant concentration.

Now note that a collectively autocatalytic set is a minimal model of a Kantian Whole, the parts, peptides, exist for and by means of the whole *task closure*, and the whole exists by means of the peptide parts. Note next, as an essential side point, that given the collectively autocatalytic set, we can define the *function* of a part by its causal consequences that serve to maintain the collectively autocatalytic whole; that is, catalyzing the appropriate next ligation reaction, not wiggling water in the petri plate. So functions that are real in the universe are a *subset* of the causal consequences of the parts. So Kantian wholes have parts with some causal consequences as functions and other causal consequences as irrelevant side effects in that environment.

4. Task Closure in an Evolving Reproducing Bacterium
   A reproducing bacterium achieves a *task closure* that is much wider than mere catalysis. Membranes are formed, DNA replicated, chemiosmotic pumps built and vectored to proper membrane locations, receptors are constructed and located in membranes all for the bacterium to reproduce.

5. The Uses or Functions of a Screw Driver *Cannot be Algorithmically Enumerated*
   I now jump to a seemingly strange topic. Can you list all the uses of a screw driver? Screw in a screw, open a paint can, wedge open or closed a door, stab an assailant, prop up a piece of cardboard....The uses of a screw driver are *indefinite in number*. Next, the integers are orderable, $1,2,3,4,\ldots$, but are the uses of a screw driver orderable? Say beyond its “first use” to screw in screws? *No*. But this means that there is *no effective procedure, or algorithm, to list all the uses of a screw driver*. This is the famous unsolved*frame* problem of computer science.

6. Evolution *Find Unprestatable Uses of Molecular Screw Drivers in Evolving cells, Then Selected at the Level of the Kantian Whole Cells*
   In an evolving bacterium in, say a new environment, all that has to happen is that someone or more molecular or cellular component screw drivers *find a use* that enhances the fitness of the evolving cell. Then if there is heritable variation for that new or improved use, it will be grafted into evolution by Natural Selection. But we cannot list, hence cannot prestate the new use of the molecular screw driver selected at the level of the Kantian whole cell. Thus, we cannot prestate the way the very phase space, the space of possibilities, of evolution changes. (Note that this *is* the arrival of the fittest, never solved by the NeoDarwinian synthesis.)

7. The Evolution of the Biosphere is Not Mathematizable as Integrable Laws of Motion

   Since we cannot prestate the new functionality of the cellular or molecular screw driver, we cannot prestate the way the evolving phase space of evolution changes. Hence we *cannot write down equations of motion for this evolution. Nor, since we lack prestatement of niche boundary conditions, could we integrate those equations of motion, even if we were to have them!*

   This implies that the evolution of the biosphere is entailed by no law. If correct, reductionism end at the watershed of the evolution of life. In turn this implies that we cannot prestate the space of possibilities that constitute the space over which there is a prestatable fitness landscape. Further, because we cannot prestate that ever changing phase space, we do not know its sample space, so we cannot construct a probability measure. In turn, this implies that we cannot formulate stochastic processes for the changes in the fitness landscape.

8. The Evolving Biosphere, Without Selection "Acting to Achieve It", Persistently Creates Its Own Future Possibilities

   The last point I wish to make is beyond the subject of fitness landscapes themselves, but one which I find to be stunning. If true, as what I shall say appears to be, it changes our view of the reality we live in in the evolving living world. I need to define Darwinian Preadaptations. Were we to ask Darwin the function of the human heart, he would respond that it is to pump blood. But we might say hearts make heart sounds and jiggle water in the pericardial sac. Why are these causal consequences of the heart not its function? Darwin would answer that we have hearts because their pumping blood was of selective significance in our ancestors. Note that therefore, as with the peptide collectively autocatalytic set above, the function of the heart is a subset of its causal consequences. More Darwin is also implicitly answering the question of why a complex organ, the heart, exists in the non-ergodic universe: Because it plays a role in sustaining Kantian Whole organisms in existence in the non-ergodic universe.

   Next, Darwin noted that a causal consequence of the heart, or other organ, of no selective significance in the current environment, might have selective significance in a new environment and be selected for that new functional significance. A new function might arise. These are called Darwinian preadaptations, or by Gould, exaptations.

   I give but one example: Some fish have an organ called a swim bladder. The ratio of air and water in the bladder-sac determines neutral buoyancy in the water column. Paleontologists believe that the swim bladder evolved from the lungs of lung fish. Water got into some lung(s), now there were sacs partly filled with air and water, poised to evolve into swim bladders. Let's assume the paleontologists are right.

   I now ask three questions: First, did a new function come to exist in the biosphere? Yes, neutral buoyancy in the water column. Note that evolution here solves the *frame* problem which algorithmic computer science cannot solve. In my understanding, the *frame* problem is that, e.g. for a robot in a room, one provides a finite list of “affordances” for objects in the room, e.g. corner of the room, floor electric plug, where the affordance finite list is of “is a”, “does a”, “uses a”, “needs a”, etc. Then anything that can be deduced from this *finite* list is within the *frame* yielded by the finite list of affordances. *But uses*, as in the screw driver uses, cannot be captured by any finite list in the sense that no effective procedure can do so. *We* solve the frame problem as humans all the time. We find new uses for object and processes. An example: Engineers, the story/fact says, were trying to invent the tractor. They needed a huge engine block, got one, mounted it on a series of ever bigger chassis, all of which broke. Finally an engineer said, “You know, the *engine block itself* is so big and *rigid*, that we can use the engine block itself *as* the chassis, and hang everything else off the engine block.” And that was the invention of the tractor and how they are made. So too were formula racing cars. This It use of the engine block’s rigidity for a new function, is a technological Darwinian preadaptation and could not, in general be prestated. This *is* the solving of the frame problem. As I said we do it all the time, Turing machines cannot, hence I believe human mind is not algorithmic, see my speculative paper, Answering Descartes: Beyond Turing [3]. So too, I think the swim bladder solves the frame problem.

Second, did the swim bladder, once it exists, alter the future evolution of the biosphere? Yes, new species evolved with swim bladders, new proteins evolved. And particularly important, once the swim bladder exists, a worm or bacterium or both might evolve to live in swim bladders, so the existing swim bladder is what I’ll call an “Empty Adjacent Possible Niche”. Thus the swim bladder, once it exists, changes the future possible evolution of the biosphere. I return to this for it is the main point of this last section. Third, now that you know what preadaptations are, can you name all possible Darwinian preadaptations, just for human evolution in the next 4 million years? We all say *NO*. Why? Well, how would we name all possible selective environments? Now that we had listed all those environments? How would we list all the features of one or several organisms that might constitute preadaptations? We cannot. And the reason was given above, “the uses” of a screw driver are indefinite in number and unorderable, so no algorithm can be an effective procedure to list them all. And if we take one use of a screw driver, say to open a can of paint, the number of other objects/processes that can open a can of paint is indefinite and unorderable, so again, no effective procedure or algorithm can list them. Hence our *no* above. Hence we not only do not know what *will* happen, we do not even know what *can* happen.

Now return to the existing swim bladder as an “Empty Adjacent Possible Niche” that changes the future possible evolution of the biosphere. Do we think that natural selection acted on a population of lung fish to “craft” a well functioning swim bladder? Yes, of course. The swim bladder is a selected preadaptation, “achieved” by natural selection. But do we think that natural selection “acted” in any sense of “act” to “achieve” the swim bladder as a new Adjacent Possible Empty Niche? NO! Selection was involved in achieving a functioning swim bladder. But selection was not evolved in creating that swim bladder AS a new Adjacent Possible Empty Niche. Yet once that niche exists, it alters the future possible evolution of the biosphere, for the worm or bacterium really might come to evolve to live in the swim bladder.

But this means something profound: Without selection "acting" to do so, the biosphere is persistently creating its own future possibilities! The biosphere, beyond selection, persistently creates what it may become. If the above two sections are right, reductionism fails for the evolution of life and we are beyond Newton and Schrödinger. If the last section is right, we are beyond even Darwin.

May this fine book add to the growing discussion of all these topics.

Acknowledgements. This work is partially supported by the TEKES FOUNDATION for my position as a Finland Distinguished Professor.

References

[1] Kauffman, S.: At Home in the Universe. Oxford University Press, New York (1995)

[2] Kauffman, S.: The Origins of Order: Self–organization and Selection in Evolution. Oxford University Press, New York (1993)

[3] Kauffman, S.: Answering Descartes: Beyond Turing. In: Barry Cooper, S., Hodges, A. (eds.) The Once and Future Turing: Computing the World. Cambridge University Press, Cambridge (in press, 2013), http://mitpress.mit.edu/sites/default/files/titles/alife/0262297140chap4.pdf

[4] Longo, G., Montevil, M., Kauffman, S.: No entailing laws, but enablement in the evolution of the biosphere (2012), http://arxiv.org/abs/1201.2069

[5] Longo, G., Montevil, M., Kauffman, S.: No entailing laws, but enablement in the evolution of the biosphere. In: Soule, T., Moore, J.H. (eds.) Proc. GECCO Companion, pp. 1379–1392. ACM, New York (2012)

[6] Wagner, N., Ashkenasy, G.: Symmetry and order in systems chemistry. The Journal of Chemical Physics 130, 164907–164911 (2009)

Santa Fe, N.M. USA, March 6, 2013 Stuart Kauffman

Preface

Observing life on Earth, it is hard to ignore its overwhelming abundance, diversity and beauty, its finely–tuned structures and forms, and its imaginative as well as purposeful behaviors and functions. First and foremost this applies to all forms of natural life, meaning the molecular, carbon– and protein–based forms of life found on our planet. Clearly on a smaller scale but nevertheless, also instances and examples of artificial life forms created in digital computers can exhibit properties that surprise us in their beauty and complexity. For both cases, arguable for the former even more so than for the latter, it is as obvious as interesting to ask why life is as it is and how it came (or comes or will come) into being. It is scientific consensus and hence tempting to give a rather simple and in some ways self–explaining answer: by the mechanism of Darwinian evolution. This certainly is true but simply poses other questions. How is evolution working? How does it enable the development of life forms? Are evolutionary developments in some ways directed, or even forced and can be predicted within certain bounds? Or is evolution directionless, open–ended and indeterminate with respect to possible outcomes? What role does chance and randomness play in evolution in general? What can be realistically expected to be the outcome of a certain period of evolutionary development? What is a meaningful mathematization of evolutionary dynamics? What requirements and preconditions must be fulfilled for the emergence of complex biological forms and behaviors, may they be natural or artificial?

Admittedly, answering all these questions conclusively is far beyond the scope of this book; in fact, it is beyond the current understanding in the sciences in general. Nevertheless, we belief that the foundation for answering these questions is an understanding of evolution as a dynamical process. This goes along with (and is unthinkable without) describing the driving forces that enable evolutionary dynamics. Addressing the dynamics of evolution is the main scope of this book, and the approach we use is the framework of fitness landscapes.

Fitness landscapes are an abstract way for describing the relationship between the genetically possible (genotype), the actually realized (phenotype) and the survival/reproduction success (fitness). Differences in the fitness over genotypic space together with the Darwinian imperative to move into the direction of increasing fitness (codified by the notion of natural and sexual selection) results into the driving forces that are behind evolutionary processes. Undeniably, evolution is a defining feature of natural life. However, we take the view that evolutionary processes are much more general than biology. In particular, the dynamical processes in evolutionary computation are in their core and essence as much a manifestation of evolution as the processes observed in biological systems. For using evolutionary search algorithms to solve optimization problems, we are interested in basically the same questions as above. What algorithmic performance can be expected for a given evolutionary search algorithm and optimization problem (that is a given genotype–to–fitness map)? How will performance scale if either the algorithm's parameters or the setting of the optimization problem, or both, were to change? Also, answering these questions requires understanding the underlying evolutionary dynamics and can hence be approached by fitness landscape methods.

In theoretical biology, recent experimental and numerical works involving mutational evolution of molecular functions revealed far–reaching implications on possible evolutionary paths, which have renewed the interest in fitness landscapes. In computer science, studies of artificial evolution in the form of artificial life and evolutionary computation increasingly used fitness landscape methods to describe evolutionary dynamics. Using these methods is particularly aimed at increasing our knowledge about the working principles of the algorithm, its expected behavior as well as some aspects of performance. Both fields, theoretical biology and evolutionary computation, to a certain degree experienced a renaissance in using landscape methods with a significant number of recent works. However, the recent progress can be found in the literature only in a very decentralized manner. Hence, the motivation to write and edit this book came from two observations: the recent advances in understanding fitness landscapes in both theoretical biology and computer science and the lack of a book covering it. We have invited some of the leading researchers that drove the recent advances in the field to provide their views on various aspects of fitness landscapes.

One main aspect of the book is that research in fitness landscapes has been separated into many distinct fields and would benefit from some kind of unification. This is also to promote communication between the fields and cross–fertilization of ideas. Consequently, we have encouraged the authors to put emphasis not only on specific questions and methodological details, but also on fundamental questions as to what is the inner sense and meaning of the approach, what is the background and the underlying principles, how is it related to research around it, and where is it going to (or could it go to). Of course, we were not aiming at compiling a tutorial or textbook, but we advocated the chapters (in difference to usual journal or proceedings papers) to be able to stand alone and to be understandable in itself by scientists of roughly related fields not yet working on fitness landscapes. In other words, if it appeared necessary to compromise between a certain degree of redundancy between chapters and the stand–alone ability of a chapter, we promoted the former over the the latter.

The book has in total 20 chapters and a foreword. The chapters are not ordered by theory and application, as the book title might suggest, but by five groups of themes. Almost all chapters address both theoretical and application aspects. We have divided the chapters that follow into five parts:

- Part I: Principles and perspectives
- Part II: Topology, measures and problem hardness
- Part III: Coevolution and dynamics
- Part IV: Visualization and characterization
- Part V: Outlook and afterthoughts

Part I: Principles and Perspectives

In this part, we intend to set out basic principles of and different perspectives to fitness landscapes. Chapter 1 by Hendrik Richter gives an introduction to the book and an overview of concepts, notions and mathematical descriptions of fitness landscapes. The chapter discusses major motivations to use landscape paradigms and considers how topological features of the landscape give raise to evolutionary dynamics. It further introduces examples of computational and empirical landscapes. Herschel Rabitz, Re–Bing Wu, Tak–San Ho, Katharine Moore Tibbetts and Xiaojiang Feng review in Chapter 2 recent developments in considering the landscape's input variables as controls. The chapter further specifies three assumptions whose satisfaction permits a general analysis of the landscape topology and demonstrates that control landscapes may be devoid of suboptimal critical point traps. The chapter applies this analysis to control landscapes that arise in quantum mechanics, chemical and material science, and in natural and directed evolution. Takuyo Aita and Yuzuru Husimi study in Chapter 3 evolutionary processes using the concept of the information gaining process. As the evolutionary process can be modeled as a walk on a fitness landscape, the evolving entity collects biomolecular information. Using this information gaining approach the chapter draws a link between evolutionary dynamics and thermodynamics, introduces the concept of "free fitness" which is analogous to free energy, and proposes that evolution is driven in the direction in which the free fitness increases.

Part II: Topology, Measures and Problem Hardness

Non–trivial fitness landscapes originate from differences in fitness over genotypic space. These differences in fitness cast the landscape's topological features, which in turn shape possible evolutionary paths. This part is devoted to the relationships between the landscape's topology and the hardness of locating evolutionary paths. Crucial links between topology and problem hardness are formed by landscape measures which quantify the effects the topology has on search paths. Chapter 4 by Katherine M. Malan and Andries P. Engelbrecht considers metaheuristic search algorithms and their optimization performance. It focuses on the feasibility of predicting algorithm performance on unknown real-valued problems based on fitness landscape features. The chapter proposes normalized metrics for quantifying algorithm performance on known problems and shows that fitness landscape techniques are useful as a part–predictor of algorithm performance. A related but slightly different aspect is discussed in Chapter 5 by Guanzhou Lu, Jinlong Li and Xin Yao, which also deals with problem hardness and evolutionary search algorithms. A problem hardness measure is presented in this chapter which is derived from a theoretical difficulty measure widely used in complexity theory. It is shown how the measure can be incorporated with a machine learning algorithm for parameter tuning and hence contributes to the goal of constructing better suited algorithms for solving problems. In Chapter 6 by Konstantin Klemm, Jing Qin and Peter F. Stadler another fundamental issue is addressed with considering geometric notions for describing the structure of landscapes as well as the dynamics of local search on them. Focusing on discrete, combinatorial landscapes and emphasizing the complications arising from local degeneracies, the authors introduce combinatorial vector fields as a mathematical tool for understanding landscape features. Also the coarse graining of landscapes is studied from two perspectives. Chapter 7 by Kristina Crona resumes with a geometrically oriented study of landscapes and concentrates on graphs and polytopes. It introduces fitness graphs for describing coarse properties of landscapes such as mutational trajectories and the number of peaks. Triangulations of polytopes give raise to shapes that can replace the well established concepts of positive and negative epistasis for two mutations. Yoshiaki Katada deals in Chapter 8 with two important topological features of landscapes: ruggedness and neutrality. As the evolutionary dynamics on a fitness landscape with neutrality shows special characteristics, ruggedness alone might be insufficient. In the chapter, a neutrality measure called standard genetic distance is introduced, which originates from population genetics, for measuring neutrality of fitness landscapes. Numerical experiments are reported and demonstrate that genetic distance is a reliable method for estimating the degree of neutrality of real-world problems. Chapter 9 by Gabriela Ochoa, Sébastien Verel, Fabio Daolio and Marco Tomassini gives an overview of local optima networks (LON) which are a recently introduced network–based model of combinatorial landscapes. The model compresses the information given by the whole search space into a smaller mathematical object and yields a new set of metrics to characterize the structure of combinatorial landscapes. The approach is applied to two well–known combinatorial optimization problems and the experimental results show that the network features correlate with and even predict the performance of heuristic search algorithms operating on these problems.

Part III: Coevolution and Dynamics

Traditionally, the structure and topology of fitness landscapes are considered to be static. This part brings together chapters that consider different approaches dealing with dynamic landscapes, particularly dynamics that is environmental or occurs in and is caused by coevolution. Chapter 10 by Hendrik Richter gives an overview of landscapes whose fitness values change with time. The chapter studies these time–dependent landscapes in two contexts. One is evolutionary processes that take place in dynamic environments and result in dynamic fitness landscapes. Another is coevolutionary processes where the fitness of a given individual depends on the fitness and the genotype of other individuals in a temporal or spatial fashion and results in codynamic landscapes. The chapter gives an overview of issues in and problems of time–dependent fitness landscapes and particularly highlights several types of mathematical descriptions and their properties. Ricard V. Solé and Josep Sardanyés focus in Chapter 11 on coevolutionary dynamics. The chapter introduces the Red Queen hypothesis of evolution and comments on some theoretical aspects and empirical evidence. It further reviews key issues of evolution on simple and rugged fitness landscapes and presents modeling examples of coevolution on different fitness landscapes at different scales using examples from RNA viruses to complex ecosystems and macroevolution. Chapter 12 by Richard A. Watson and Marc Ebner treats another aspect of dynamic landscapes with the interactions of evolutionary and ecological dynamics. To understand these interactions as coupled processes leads to eco–evolutionary dynamics that can be modeled by deformable fitness landscapes. The chapter reports numerical experiments and observes that the model of deformable landscapes can exhibit either of the two behavioral modes: evolutionary stasis or continued evolutionary change (also known as Red Queen dynamics). Wim Hordijk present an overview of a statistical analysis to measure and express the correlation structure of fitness landscapes in Chapter 13. The correlation analysis is applied to both static and coupled fitness landscapes. The experimental results presented show that the correlation analysis gives a direct and useful link to the actual search performance of evolutionary algorithms that use a coevolutionary approach. Chapter 14 by Krzysztof Trojanowski concludes this part with a discussion about dynamic real–valued landscapes and methods of evaluating the efficiency of (meta–)heuristic optimization algorithms operating on these landscapes. The chapter introduces measures for dynamic performance evaluation and associated measurement methods, gives dynamic benchmarks and different types for implementing changes, and considers the role of time and uncertainty originating from the measurement method.

Part IV: Visualization and Characterization

As the structure and topology of a fitness landscape offers to gain insight into evolutionary dynamics, it is vital to have methods and tools for visualizing and characterizing landscape's properties. The chapters in this part discuss such questions. Chapter 15 by Ivan Zelinka, Oldrich Zmeskal and Petr Saloun is devoted to fitness landscapes with fractal characteristics. The main topic of this chapter is to use elements from fractal geometry to measure attributes of fractal landscapes. These attributes are taken to characterize fractal properties of basic artificial test functions as well as cost functions of real application problems that appear in experimental chaos control and synchronization. Daniel Ashlock, Justin Schonfeld, Wendy Ashlock and Colin Lee describe in Chapter 16 three important tools that were recently suggested to explore fitness landscapes: agent-case embeddings, fitness morphs, and nonlinear projection. These techniques are examined using fitness landscapes for a variety of discrete problems including finding self avoiding walks, finding features for DNA sequence classification, the Tartarus AI test problem, locating cellular automata rules, and a novel real optimization problem connected with the Mandelbrot set. The results show that the techniques discussed transform information about discrete fitness into real-valued spaces enabling both analysis and visualization. Another approach to visualize fitness landscapes is presented in Chapter 17 by Sebastian Volke, Simon Bin, Dirk Zeckzer, Martin Middendorf and Gerik Scheuermann. This approach is applied to the question of how and why changes in the design of a particle swarm optimization (PSO) algorithm influence its optimization behavior. The visual approach discussed in the chapter combines a terrain representation of the fitness landscape topology with configuration-local, time-dependent statistical measures of PSO runs and is implemented in a visualization tool called dPSO-Vis. It is demonstrated how dPSO-Vis can be used to analyze and compare the optimization behavior of PSO algorithms designed for solving the RNA folding problem. Bjørn Østman and Christoph Adami finally give in Chapter 18 a concise overview of the relationship between visualization of fitness landscapes and potential predictability of evolutionary dynamics. To know whether evolution is predominantly taking paths that move upwards in fitness and along neutral ridges, or else entails a significant number of valley crossings, there is the need to visualize these landscapes. For instance it must be determined whether there are peaks in the landscape, where these peaks are located with respect to one another, and whether evolutionary paths can connect them. The chapter focuses on the predictability of evolution on rugged genetic fitness landscapes and presents numerical results to answer the question of whether evolutionary trajectories towards the highest peak in the landscape can be achieved via a series of valley crossings.

Part V: Outlook and Afterthoughts

The final part of this book contains two shorter chapters devoted to the prospects of fitness landscape research. Possible future issues are discussed in Chapter 19 by Hendrik Richter. The chapter addresses challenges to fitness landscape approaches that result from recent experimental and theoretical findings about the information transfer in biological systems. It further sets out opportunities these results may open up and speculates about directions that landscape research may take. Chapter 20 by Edward D. Weinberger concludes the book with afterthoughts and discusses the past, the present and the future of the topic. In a personal reminiscence fueled by his involvement in fitness landscape research for over 20 years, he particularly highlights the need to rethink our concepts of fitness, the relevance of coevolutionary effects and the importance of information used by biological systems. The chapter also concludes that a main topic to be addressed by future research is to advance our understanding of biological evolution as a dynamical process. Landscape methods, if extended and adapted, have the potential to achieve this.

It is often said that there are more contributors to a book who should be acknowledged than the people whose names appear on the cover page. This literally applies to this book. So, our foremost thank goes to the in total 42 authors of the chapters who entrusted their contributions to this book. We are also very thankful for both the foreword and the afterthoughts. Stuart Kauffman, who moved the topic of fitness landscapes significantly beyond the scope of evolutionary biology with his seminal works in the late 1980s, provided a foreword that is rich in substance, even controversial and discusses limitations of current fitness landscape research as well as points at possible directions of further development. We are equally grateful that Edward Weinberger, who similarly and partly collaborating with Stuart Kauffman significantly extended the scope of fitness landscapes 20 years ago, provided afterthoughts.

To edit and write a publication collaboratively seems to be possible even over large distances nowadays due to communication devices such as email and chat. This book is also proof of that. Nonetheless, we also noted that some discussion is much more fruitful and efficient face to face by visiting each other. So, we thank the HTWK Leipzig University of Applied Sciences for providing a travel grant and the University of Pretoria for cordial hospitality that enabled this visit. Special thanks go to Dr. Thomas Ditzinger of Springer-Verlag for his support during the preparation of this book and to Prof. Ivan Zelinka, the series editor for the Springer Series Emergence, Complexity and Computation (ECC), for inviting the book to this series. Also thanks to all the reviewers who have provided very valuable inputs to improve the quality of the chapters within this book.

Last in order but clearly not in importance, our most heartfelt thanks goes to our families and friends for their support, love, encouragement and patience.

Leipzig, Germany — Hendrik Richter
Pretoria, South Africa — Andries P. Engelbrecht

List of Contributors

Christoph Adami
Department Microbiology and Molecular Genetics & BEACON Center for the Study of Evolution in Action, Michigan State University,
East Lansing, MI 48824, U.S.A.
e-mail: adami@msu.edu

Takuyo Aita
Graduate School of Science and Engineering,
Saitama University, Saitama 338-8570, Japan.
e-mail: taita@mail.saitama-u.ac.jp

Daniel Ashlock
Department of Mathematics and Statistics, University of Guelph,
50 Stone Rd. E, Guelph, Ontario, N1G 2R4, Canada.
e-mail: dashlock@uoguelph.ca

Wendy Ashlock
Department of Computer Science and Engineering,
York University, 4700 Keele Street, Toronto, Ontario, M3J 1P3, Canada.
e-mail: washlock@sympatico.ca

Simon Bin
Department of Computer Science, University of Leipzig,
Augustusplatz 10, 04109 Leipzig, Germany.
e-mail: sbin@informatik.uni-leipzig.de

Kristina Crona
University of California at Merced, 5200 Lake Road, Merced, CA 95343, U.S.A.
e-mail: kcrona@ucmerced.edu

Fabio Daolio
Department of Information Systems, University of Lausanne,
1015 Lausanne, Switzerland.
e-mail: fabio.daolio@unil.ch

Marc Ebner
Ernst Moritz Arndt Universität Greifswald, Institut für Mathematik und Informatik,
Walther-Rathenau-Strasse 47, 17487 Greifswald, Germany.
e-mail: `marc.ebner@uni-greifswald.de`

Andries P. Engelbrecht
Department of Computer Science, University of Pretoria, Pretoria 0002,
South Africa.
e-mail: `engel@cs.up.ac.za`

Xiaojiang Feng
Kadmon Corporation, 450 East 29th Street, New York, NY 10016, U.S.A.
e-mail: `fengxiaojiang@yahoo.com`

Tak–San Ho
Department of Chemistry, Princeton University, Washington Road,
Princeton, NJ 08544, U.S.A.
e-mail: `tsho@princeton.edu`

Wim Hordijk
SmartAnalytiX.com, Lausanne, Switzerland.
e-mail: `wim@WorldWideWanderings.net`

Yuzuru Husimi
Innovative Research Organization, Saitama University, Saitama 338-8570, Japan.
e-mail: `husimi.@mail.saitama-u.ac.jp`

Yoshiaki Katada
Department of Electrical and Electronic Engineering, Setsunan University,
17-8 Ikeda-nakamachi, Neyagawa, Osaka 572-8508, Japan.
e-mail: `katada@ele.setsunan.ac.jp`

Stuart Kauffman
University of Vermont, 85 South Prospect Street, Burlington, VT 05405, U.S.A.
University of Pennsylvania, 3451 Walnut St, Philadelphia, PA 19104, U.S.A.
Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM87501, U.S.A.
e-mail: `stukauffman@gmail.com`

Konstantin Klemm
Bioinformatics Group, Department of Computer Science, and Interdisciplinary
Center for Bioinformatics, University of Leipzig, Härtelstrasse 16-18, 04107
Leipzig, Germany.
e-mail: `klemm@bioinf.uni-leipzig.de`

Colin Lee
Department of Mathematics and Statistics, University of Guelph,
50 Stone Rd. E, Guelph, Ontario, N1G 2R4, Canada.
e-mail: `clee04@uoguelph.ca`

Jinlong Li
Joint USTC-Birmingham Research Institute in Intelligent Computation
and Its Applications
The Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China, Hefei, 230027, China.
e-mail: jlli@ustc.edu.cn

Guanzhou Lu
Joint USTC-Birmingham Research Institute in Intelligent Computation
and Its Applications
The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
e-mail: g.lu@cs.bham.ac.uk

Katherine M. Malan
Department of Computer Science, University of Pretoria,
Pretoria 0002, South Africa.
e-mail: kmalan@cs.up.ac.za

Martin Middendorf
Department of Computer Science, University of Leipzig,
Augustusplatz 10, 04109 Leipzig, Germany.
e-mail: middendorf@informatik.uni-leipzig.de

Gabriela Ochoa
Department of Computing Science and Mathematics, School of Natural Sciences, University of Stirling, Stirling, FK9 4LA, Scotland, UK.
e-mail: gabriela.ochoa@cs.stir.ac.uk

Bjørn Østman
Department Microbiology and Molecular Genetics & BEACON Center for the Study of Evolution in Action, Michigan State University,
East Lansing, MI 48824, U.S.A.
e-mail: ostman@msu.edu

Jing Qin
Max Planck Institute for Mathematics in the Sciences,
Inselstraße 22, 04103 Leipzig, Germany.
e-mail: qin@bioinf.uni-leipzig.de

Herschel Rabitz
Department of Chemistry, Princeton University, Washington Road,
Princeton, NJ 08544, U.S.A.
e-mail: hrabitz@princeton.edu

Hendrik Richter
HTWK Leipzig University of Applied Sciences, Faculty of Electrical Engineering and Information Technology, D–04251 Leipzig, Germany.
e-mail: `richter@eit.htwk-leipzig.de`

Petr Saloun
VSB-TUO, Faculty of Electrical Engineering and Computer Science,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic.
e-mail: `petr.saloun@vsb.cz`

Josep Sardanyés
ICREA-Complex Systems Lab, Universitat Pompeu Fabra, Parc de Recerca Biomèdica de Barcelona (PRBB), Dr. Aiguader 88, 08003 Barcelona, Spain.
Institut de Biologia Evolutiva (CSIC-Universitat Pompeu Fabra), Passeig Marítim de la Barceloneta 37, 08003 Barcelona, Spain.
e-mail: `josep.sardanes@upf.edu`

Gerik Scheuermann
Department of Computer Science, University of Leipzig,
Augustusplatz 10, 04109 Leipzig, Germany.
e-mail: `scheuermann@informatik.uni-leipzig.de`

Justin Schonfeld
Department of Integrative Biology and Biodiversity Institute of Ontario,
University of Guelph, 50 Stone Rd. E, Guelph, Ontario, N1G 2R4, Canada.
e-mail: `jschonfe@uoguelph.ca`

Ricard V. Solé
ICREA-Complex Systems Lab, Universitat Pompeu Fabra, Parc de Recerca Biomèdica de Barcelona (PRBB), Dr. Aiguader 88, 08003 Barcelona, Spain.
Institut de Biologia Evolutiva (CSIC-Universitat Pompeu Fabra), Passeig Marítim de la Barceloneta 37, 08003 Barcelona, Spain.
Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87501, U.S.A.
e-mail: `ricard.sole@upf.edu`

Peter F. Stadler
Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstrasse 16-18, 04107 Leipzig, Germany.
Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, 04103 Leipzig, Germany.
Fraunhofer Institut for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany.
Institute for Theoretical Chemistry, University of Vienna, Währingerstrasse 17, A-1090 Vienna, Austria.
Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM87501, U.S.A.
e-mail: `studla@bioinf.uni-leipzig.de`

Katharine Moore Tibbetts
Department of Chemistry, Temple University, 1901 N. 13th St.,
Philadelphia, PA 19122, U.S.A.
e-mail: katharine.moore@temple.edu

Marco Tomassini
Department of Information Systems, University of Lausanne, 1015 Lausanne,
Switzerland.
e-mail: marco.tomassini@unil.ch

Krzysztof Trojanowski
Institute of Computer Science, Polish Academy of Sciences, Jana Kazimierza 5,
01-248 Warsaw, Poland.
Cardinal Stefan Wyszyński University, Faculty of Mathematics and Natural
Sciences, Wóycickiego 1/3, 01-938 Warsaw, Poland.
e-mail: trojanow@ipipan.waw.pl

Sébastien Verel
INRIA Lille - Nord Europe and University of Nice Sophia-Antipolis,
06903 Sophia Antipolis Cedex, France.
e-mail: verel@3s.unice.fr

Sebastian Volke
Department of Computer Science, University of Leipzig,
Augustusplatz 10, 04109 Leipzig, Germany.
e-mail: volke@informatik.uni-leipzig.de

Richard A. Watson
University of Southampton, School of Electronics and Computer Science, Institute
for Life Sciences, Institute for Complex Systems Simulation, Agents Interaction
and Complexity, Highfield, Southampton, SO17 1BJ, UK.
e-mail: raw@ecs.soton.ac.uk

Edward D. Weinberger
Weinberger Post-Quantitative, Inc., 370 Central Park West,
#110, New York, NY 10025, U.S.A.
Department of Finance and Risk Engineering,
Polytechnic Institute of New York University,
6 Metrotech Center, Brooklyn, NY 12201, U.S.A.
e-mail: edw@wpq-inc.com

Re–Bing Wu
Department of Automation, Tsinghua University and Center for Quantum
Information Science and Technology, TNLIST,
Beijing, 100084, China.
e-mail: rbwu@tsinghua.edu.cn

Xin Yao
Joint USTC-Birmingham Research Institute in Intelligent Computation and Its Applications
The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK.
e-mail: x.yao@cs.bham.ac.uk

Dirk Zeckzer
Department of Computer Science, University of Kaiserslautern, Postfach 3049, 67653 Kaiserslautern, Germany.
e-mail: zeckzer@informatik.uni-kl.de

Ivan Zelinka
VSB-TUO, Faculty of Electrical Engineering and Computer Science,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic.
e-mail: ivan.zelinka@vsb.cz

Oldrich Zmeskal
Faculty of Chemistry, Brno University of Technology,
Purkyňova 118, 612 00 Brno, Czech Republic.
e-mail: zmeskal@fch.vutbr.cz

Contents

Part I: Principles and Perspectives

Part II: Topology, Measures and Problem Hardness

Part III: Coevolution and Dynamics

Part IV: Visualization and Characterization

Part V: Outlook and Afterthoughts

Part I
Principles and Perspectives

Chapter 1
Fitness Landscapes: From Evolutionary Biology to Evolutionary Computation

Hendrik Richter

Abstract. This chapter gives an introduction to the book and an overview of fundamental concepts, notions and mathematical descriptions of fitness landscapes. Based on a brief discussion of the origin of landscape paradigms, major motivations to use fitness landscapes are presented. It is further considered how topological features of the landscape give raise to evolutionary dynamics. Also, examples of computational and empirical landscapes are introduced.

1.1 Introduction

This book is concerned with recent advances in the theory and application of fitness landscapes and this chapter intends to give an overview of the fundamental concepts and notions. Fitness landscapes are an abstract way to express the relationship between the genetically possible (genotype), the actual realized traits (phenotype), and the phenotype's survival/reproduction success (fitness). This fitness is the main driving force behind evolutionary processes, and evolution, in turn, is a defining ingredient of life. Section 1.1.1 sets out conceptional background and motivation for using fitness landscapes in studies of natural and artificial evolutionary processes. For understanding and appreciating the role fitness landscapes play in recent advances in evolutionary biology and evolutionary computation, it might be in order to briefly look at the history of using landscape models, which is given in Section 1.1.2. This Introduction is concluded with a more detailed discussion about life, evolution and fitness. Section 1.2 formally introduces fitness landscapes and provides mathematical notions and concepts. Fitness landscapes originated from evolutionary biology. Section 1.3 overviews fitness landscapes for studying how evolutionary development proceeds and how this may offer clues about predicting evolutionary paths in biological systems. The three major sources that fueled this discussion

Hendrik Richter
HTWK Leipzig University of Applied Sciences,
Faculty of Electrical Engineering and Information Technology, D–04251 Leipzig, Germany
e-mail: richter@eit.htwk-leipzig.de

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_1, © Springer-Verlag Berlin Heidelberg 2014

are reviewed, namely (i) geometrical intuition and conception, (ii) computational experiments with designed models of fitness landscapes, and (iii) microbiological (microbial) experiments which give raise to empirical fitness landscapes. It is also briefly described how insights from fitness landscapes in evolutionary biology were received, modified and adapted in evolutionary computation. This chapter only considers fitness landscapes for a single species that are constant over time. However, changing environments and the coexistence of several species alter the rules for staying alive. For an introduction of dynamic fitness landscapes and coevolution refer to Chapter 10.

1.1.1 Motivation

One of the central research questions of the 21th century is how to advance our understanding about what life is in itself and what organizational, structural and developmental principles may be connected with it. Here, life means in the same way the state of being alive as well as the process of exhibiting life–like properties. Posing these questions is at least accompanied by, but rather interacts with, is partly driven by, and partly drives, two other major trends in modern science. One is the substantial progress in our knowledge about the biochemistry of living structures on a molecular (microscopic) level. This starts with sequencing whole genomes and decoding them, goes on with understanding the mechanisms of the transfer of genetic information from the genome to proteins, reaches as far as the dynamics of protein structuring and folding, and concludes with the functions a protein (or a set of proteins) has in an organism and how this function might affect reproduction success and longevity. Each of these topics has reached a certain degree of maturity and is more or less solidly based, while at the same time some fundamental questions still await to be answered. One of the main research questions within this trend is to establish a sound theoretical understanding of how this chain of topics can be consistently connected.

The other trend is the ubiquity of computing devices in the form of cheap, reliable, fast and increasingly networked digital computers. This is accompanied by attempts to make more and more structures and processes available for mathematical and algorithmic description with the general aim to make these structures and processes computable. This naturally applies to innate fields of scientific computing, such as physics and chemistry, but also extends (at least partially) to biological, neuropsychological and social phenomena. This touches on the question of what is computable in general, but also what kind of mathematical description is most suitable for catching the essentials of what is studied, and how this description can be obtained. It goes on with the question of what is the appropriate computational environment for the mathematical/algorithmic description of structures and processes and how to put the description to the computational environment, and concludes with designing, conducting, evaluating, visualizing and interpreting the numerical (computational) experiments. Arguably, for working on detailed questions in each of these two trends, it might be possible to take little notice of the other one.

However, some of the most intriguing and fundamental issues exist in the border territory between both trends, most notably in computational biology and bioinformatics, but also in artificial life and evolutionary computation. Life in itself can be understood as propagation and transfer of information and hence in its core and substance is a computational issue [1, 2, 98]. It seems to be obvious that this is the case for all attempts to model and compute life and/or life–like behavior artificially. However, this explicitly applies to all known forms of natural life, that are the molecular, carbon– and protein–based forms of living beings on our planet.

The explanatory and predictive power of natural sciences is closely related to the possibility of meaningful mathematization. Immanuel Kant's famous dictum that

> "any particular natural theory contains only so much true science as it contains mathematics"[1]

catches the spirit of this thinking as well as gives it a pragmatic outlook at what is (or even should be) the developmental direction of a branch of science. Particularly in physics, but also in chemistry, this became the way to proceed after the dictum was first uttered near the end of the 18th century. The remarkable success in explaining and predicting natural phenomena of inanimate, non–living matter that has set in ever since is unthinkable without the powerful mathematical penetration we find in physics and chemistry. The same is not exactly true for biology and the study of living matter. It cannot be said that all fields of biology at all times openly embraced mathematics. There has been (and still is) much argument about biology being different from other sciences, particularly from physics and chemistry, and therefore defying, at least partly, meaningful mathematization. Even thinkers in biology as influential and progressive as Ernst Mayr sometimes detected in mathematics "a harmful influence on the development of biology" [73], p. 304. There are several reasons for this state of affairs: for the harsh coexistence of potentials and limitations, of promises and pitfalls that are connected with the fundamental progress towards algorithmization, computization and mathematization in biology; see e.g. [19, 57, 65, 106] for some recent discussion. One group of reason surely is that biological systems exhibit an enormous amount of complexity, inhomogeneity and diversity that seemingly defies to be harnessed to fundamental, all–compassing natural laws that can be expressed by rather simple, equation–like mathematical descriptions [6, 92]. In physics the presence of simple balance principles such as conservation of energy/mass or momentum enables expression of fundamental relationships by equations. Frequently, these equation–like mathematical descriptions of physical laws can be solved analytically (or meaningfully simplified to become analytically solvable) and give raise to easily calculable and evaluable formulae for timely evolutions. This property sometimes puzzled even the profoundest thinkers in physics, and Eugene Wigner even went as far as to acknowledge an "unreasonable effectiveness of mathematics" [135] in describing physical reality. Quite the

[1] The original quotation is: "Ich behaupte aber, daß in jeder besonderen Naturlehre nur so viel eigentliche Wissenschaft angetroffen werden könne, als darin Mathematik anzutreffen ist." [50], Preface, p. XIV.

contrary seems to be true in biology. Here, complex interactions and interdependencies on and between many spatial and temporal scales are found, also between causes and effects, structures and behaviors, forms and functions. All these facts apparently hinder to set up rules and laws that apply to all instances of a well–defined group of phenomena. In biology, it seems, the occurrence of important exceptions from an established rule is the only rule without exceptions.

A possible way out of this difficulty occurring in biology might be to put much more emphasis on case–based, local algorithmization and computization as on the forms of mathematization frequently found in physics and chemistry. This goes conveniently along with the aforementioned trend of computization in other branches of science. To describe biological phenomena, it appears, an algorithmic and computational approach is frequently much more suitable than looking for a set of equations that can explain all. Moreover, to understand the information flow in biological systems and processes by making it computable appears to be a more promising option to overcome the limitation of a theorem–and–derived–fundamental–law approach. Studying living matter most likely will not be based on a small number of fundamental laws expressible by rather simple equations.

The quotation mentioned above not only gives programmatic direction for a field of science, but also defines an axiomatic baseline that is, once accepted, generally agreed upon without further proof or argument. Also in the life sciences there are basic convictions forming some foundation for the field. One is expressed by Theodosius Dobzhansky and states:

> "Nothing makes sense in biology except in the light of evolution," [24] p. 449.

Our knowledge about a biological system is only superficial if its evolutionary processes are not included and not understood; it could even be said that the degree of understanding a biological system scales to the understanding of its evolution. So, in consequence of the discussion above, it becomes natural to seek for a mathematical or algorithmic or computational framework for describing evolutionary processes. Because the topic is of considerable significance for biological phenomena, there are several approaches to it. One is the mathematical theory of population genetics [20, 32, 64], also see [89] for a computational perspective. This theory intends to treat the Darwinian process of inheritance, genetic drift, variation and selection by considering allele frequency distributions. This may result in formulae that describe, for instance, the probability that an allele connected to a particular advantage goes into fixation, how long that would take, and how different rules of inheritance would affect these processes.

Another way to describe the dynamics of evolutionary processes is fitness landscapes. The conceptional framework of fitness landscapes origins from theoretical biology and intends to address some of the most intriguing and fundamental questions in natural and artificial evolution: what way is evolution going, to what extend is it predictable, what can be realistically expected to be the outcome of a certain period of evolutionary development? For approaching these questions it appears to be useful to have a mathematical notion about how the genetically possible (genotype), the actual existing biological instance (phenotype) and the survival/reproduction success (fitness) relate to each other. This is what a fitness landscape in an abstract way does. It offers a framework for analyzing and visualizing the relationships between genotype, phenotype and fitness. Interestingly, addressing these questions by the framework of fitness landscapes is valuable and well–established practice in both evolutionary biology and evolutionary computation. Naturally, the above mentioned fundamental questions take slightly different forms in both fields.

1.1.2 Historical Background

Fitness landscapes were first proposed by Sewall Wright [140] in the 1930s as a mapping from a set of genotypes to fitness. The genotypes were countable and neighboring in such a way that genotypes next to each other can mutate from one to another. The fitness landscape framework axiomatically assumes that for every genotype there is (or can be assigned) a fitness value[2,3]. This fitness value can be viewed as an elevation over the genotypic space. In other words, fitness is a property of the genotype expressed as its orthogonal projection. Based on this convention, the fitness landscape metaphor becomes apparent: that of a mountainous region with peaks, valleys, ridges and plateaus (see Figure 1.1). These topological features of the fitness landscape are critical for understanding the evolutionary dynamics that is laid out by the landscape's structure. Using this understanding it becomes explainable why some genotypes are more often realized as phenotypes than others as evolutionary time goes by. Also it is explicable how likely it is that one phenotypic realization of a genotype origins from another. So, another important part of the fitness landscape thinking is that of evolutionary paths that are engraved on

[2] Wright also advocated a second type of fitness landscape that is conceptually related, but methodologically different to the one considered in this chapter. In this second type, we have a mapping from the allele frequency in a given population (replacing genotype) to the population mean fitness (replacing genotypic fitness). It can been argued that the population mean fitness is continuous for population size going to infinity. Hence, such continuous population mean fitness landscapes can be seen as filling the gaps between discrete phenotypes that inevitable occur in granular (discrete) genotypic fitness landscapes because of this discreteness. Moreover, the genotypic landscape is obtained from the population mean landscape for population size going to unity [133]. On the other hand, there has been substantial criticism on population mean fitness landscapes, mainly on the ground of these landscapes being an entirely geometrical concept inaccessible to computational or empirical treatment [74]. For the same reason these landscapes are not considered further in this chapter.

[3] In the literature, next to the term fitness landscape, also *adaptive landscape* can be found. Both terms adaptive and fitness landscape are sometimes used synonymously (for instance by Wright himself [140, 142]), which may cause some confusion. More often, they are taken to discriminate between landscapes that express relations between collective quantities (allele or phenotype frequencies and mean fitness) and landscapes that express relations between individual quantities (genotypes, phenotypes and fitness). Adaptive landscape refers to the former, while fitness landscape refers to the latter. By this convention, a population mean fitness landscape is an adaptive landscape. According to the reasons given in the footnote above, the term fitness landscape is used and promoted in this chapter.

the landscape's surface. These paths are taken by a succession of phenotypic realizations and utilize the population's genotypic makeup from low–fitness regions to higher–fitness regions.

The concept of fitness landscapes introduced by Wright [140] attracted interest particularly because it offered an approach to conceptionalize and visualize evolutionary dynamics in various population genetics modes. The created diagrams were considerably influential for thinking in an abstracted way about possible paths evolutionary processes can follow and fostered an appropriate intuition for understanding evolutionary dynamics; see e.g. [85, 99, 108] for more detailed historical remarks. However, as a conceptualization and visualization tool alone fitness landscapes were mainly an illustration of certain settings and stages in the evolutionary process. In other words, the landscapes were used retrospective to show how a certain known type of evolutionary dynamics could be explained. In their original form, fitness landscapes did not allow a computational approach to experiment with possible evolutionary scenarios.

A first instance for a more computational approach to understand evolutionary dynamics was proposed by Stuart Kauffman collaborating with Simon Levin [53] and Edward Weinberger [54], called NK fitness landscape. The NK fitness landscape, formally introduced in Equation (1.3) in Section 1.3.2, is a computational model where the ruggedness (or smoothness) of the landscape can be tuned by parameters, called N and K. The concept of ruggedness (or smoothness) intend to generalize the features in the landscape metaphor. The genotypes are coded as a string of length N over a given alphabet. Each string is attributed with a fitness. Fixing the way a string is built from the alphabet and establishing a distance measure between strings define which genotypes are sitting next to each other. In addition, K epistatic interactions modify the fitness such that the fitness of each string is the sum of contributions from K other strings. This allows tuning between completely smooth ($K = 0$) to most rugged ($K = N - 1$). There exists an extensive and sophisticated body of work, e.g. [26, 54, 55, 128, 131, 132], analyzing various aspects of this landscape and providing important insight into evolutionary dynamics. In particular, these studies have shown the importance of the landscape's topology for possible evolutionary pathways, and became the starting point for developing numerically evaluable quantities, so called landscape measures, that can be seen as to define some types of metric for fitness landscapes. These measures quantify a landscape by (possible a set of) numbers and allow comparison and assessment of different types of topologies. Furthermore, the studies introduced concepts such as modality, ruggedness, neutrality, long–path problems. These concepts characterize different types of evolutionary scenarios. In addition, the works on NK landscapes triggered and inspired thinking about whether such concepts would also be valuable for understanding the working principles and behaviors of evolutionary computation methods such as genetic algorithms, evolutionary programming and evolution strategies (see e.g. [7, 36] for a brief history of evolutionary algorithms). Referring explicitly to the works of Kauffman and Weinberger, a first attempt to explain the working of a genetic algorithm using landscape analysis methods was by Bernard Manderick and co–workers [72]. Also here much work has been done between then

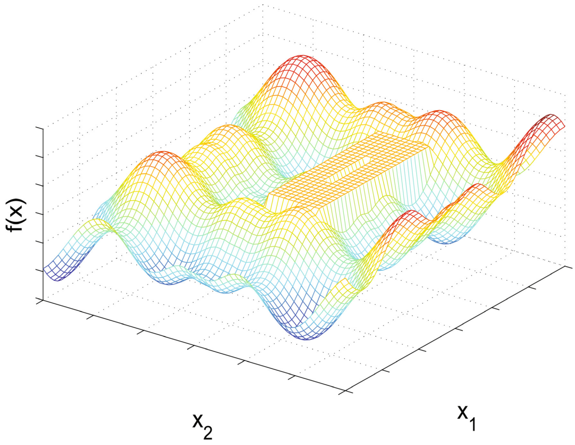


Fig. 1.1 Static fitness landscape in $\mathbb{R}^2$ as a mountainous region with peaks, valleys, ridges and plateaus

and now, and particularly the work of Terry Jones [46] turned out to be markedly influential on applying landscape methods for analyzing, understanding and designing evolutionary search algorithms.

It appears almost a little surprising that such a rather naive picture as a fitness landscape has meaning in evolutionary biology and evolutionary computation. Even more so as there are conceptual similar mathematical constructs in branches of sciences as illustrious as physics and chemistry [38, 110]. In this context they are called energy landscapes. These energy landscapes occur for instance in the theory of spin glasses [12, 79, 105, 143], in describing plastic dissipation processes in metals [81], and surface patterns in alloys and ceramics [126]. In biochemistry the folding and energy relaxation in proteins [82, 83, 109] and nucleic acids [8, 10, 47] can be described by energy landscapes. Furthermore, there is a conceptually close relationship to the potential energy surfaces of theoretical chemistry. The main difference between energy landscapes and fitness landscapes is that the elevation quantity in the evolutionary context is fitness, which is meant to be maximized, while the energy that specifies the elevation in energy landscapes is to be minimized. Clearly, this simple inversion does not alter the intrinsic mathematical meaning and content, but as the landscape metaphor originates from a geographical conceptualization it also molds the way we perceive the potential for such an abstraction. Next to the static topological features in the landscape this particularly applies to any form of dynamics that may occur in or on the landscape. The dynamic behavior can be thought of as that of motion on the landscape's surface. In terms of the landscape metaphor, next to the static mountains and valleys that characterize topology, there are lakes and flows that are the dynamic consequences of the topology and define potentials for and actions of movement. Of course, the heuristic value of fitness landscapes does not stem from such two-dimensional images or picturing two–dimensional surfaces. The value of the landscape metaphor is what can be mathematically derived from it.

The main motivation to employ a landscape approach is that it offers a framework for a computational treatment. This treatment becomes geometrically interpretable in a meaningful way for the aforementioned simple two–dimensional case, but not for landscapes with a genotypic space with higher dimension. As any interesting evolutionary process most likely has genotypes with more than two degrees of variation, this poses a serious limitation and a problem to overcome. This is known since the beginning of employing fitness landscapes to understand evolutionary dynamics. No matter whether the landscapes may originate from modelling biological processes or computational evolutionary search, the problem is present. The tension and mixture between intuitive usefulness and apparent limitation is for instance highlighted by Terry Jones in the aforementioned work laying out the potential of landscape methods for describing computational evolutionary search, [46], p. 45–46:

> "The term 'landscape' has something powerfully seductive about it. The imagery it evokes is so appealing that further thought can be completely suspended. An important question to ask is why we would want to use such a term. The answer is presumably that we hope to use the imagery (e.g., peaks, ridges, valleys etc.) to enhance our understanding of some process to develop new ideas for exploring spaces and to stimulate questions about processes operating on these structures. All of this tends to rely rather heavily on the simple properties that we see in physical three dimensional landscapes. It is not clear just how many of the ideas scale up to landscapes with tens or thousands of dimensions."

Similar remarks can also be found in biological literature on fitness landscapes, e.g. [99]. However, to overcome the dimensionality issue is one of the major topics in recent advances in the theory and application of fitness landscapes. Two major directions are employing landscape measures [94, 115] and visualization schemes [74, 136]. Almost all chapters of this book address these topics to some extend.

There is an extended body of literature on theoretical and applicational problems and issues of fitness landscapes in both fields, evolutionary biology and evolutionary computation. Particularly notable is the book by Gavrilets [41], who reviews the state–of–the–art of a computational approach in evolutionary biology and aims at summarizing and generalizing mathematical models that describe the dynamics of speciation. A slightly different approach is applied by McGhee [75], who uses graphic concepts, called theoretical morphospaces, for describing the process of evolution. Morphospaces are geometrical spaces of both existent and non–existent biological form.

1.1.3 Life, Evolution, and Fitness

Up to now, there is no final scientific agreement about what exactly defines life, and it even has been argued that attempts to define life addressing the needs from different fields of science are all in vain [18, 17, 69]. On the other hand, there is, in fact, also a certain amount of consensus as to what the defining ingredients of life are. Naturally, defining life is inextricably intertwined with the aforementioned task of identifying what life is and how its emergence from the physical world has taken (or takes) place. To put it into other words, defining life should be catching the essence of understanding life. So apart from the fields of evolutionary biology and artificial life (where fitness landscapes are frequently used and hence are the focus of this chapter – and this book), the question of defining life also plays a major role in molecular biology, synthetic biology and astrobiology. Reviewing the different views in these fields, we generally notice three major components that are independently necessary and jointly sufficient for defining life: self–reproduction, metabolism, and evolution. Based on this understanding, a commonly used working definition of life is [48] "a self–sustained chemical system capable of undergoing Darwinian evolution", which has also been adopted by the Exobiology program at NASA. It appears to be interesting in our context that a paramount importance is put on Darwinian evolution and its mechanism for understanding life.

Key components of this Darwinian mechanism are that each and every living being is specified by a genetic code that codes for the individuals' features and abilities. These features and abilities determine (at least on average) the individuals' success in survival and reproduction by the process of natural and sexual selection. To make this distinction even more clear, genetic coding is associated with the individuals' genotypes, the abilities and features of the individuals with the phenotypes. For expressing the (relative) amount of success quantitatively, the concept of fitness is employed. In general, fitness measures the individual's viability as the likelihood to survive to the age of reproducing and actually reproduce. In other words, the larger the reproduction success of an individual is, the higher is its fitness. As the genetic code (or at least snippets of it) is propagated to the next generation by reproducing, features and abilities become inheritable. On average, this applies disproportionately for features and abilities that enhance fitness.

It should be mentioned that there is occasional criticism on the importance of Darwinian evolution for defining what life is. This has much to do with the question of why such a definition is important and what the intended (or possible) use of the definition is. Some of the fields of science mentioned above, namely artificial life, synthetic biology and astrobiology, are frequently faced with the following task: they deal with phenomena (either observed extra–terrestrially as in astrobiology, or programmed in software, as in artificial life, or bio–chemically set up in wetware, as in synthetic biology) and are to decide if the phenomena qualify or not as constituting life or being alive. There are several issues with including Darwinian evolution in such a definition. A first is in astrobiology, where an in situ search for life on another celestial body is hindered by the need to find evidence for Darwinian evolution as it may take a considerable period of time to establish such an observation [16]. In synthetic biology it is conceivable to have naked RNA molecular life forms, which would conflate phenotype with genotype and hence violating some principles of Darwinian evolution [49, 66]. Artificial life models with metabolic properties have been studied that show chemical and spatio–temporal self–organization similarly to living entities but forgo evolutionary processes in a strict sense [30, 33, 120]. However, in all these examples the absence of Darwinian evolution is caused by operational or methodological objections, not as a result of principle considerations. Hence, it is sensible to say that every realistic understanding of life should incorporate evolution as a leading and defining factor, and that thinking about life is incomplete without evolutionary processes. Moreover, understanding the mechanisms and dynamics of evolution is a necessity to explain the emergence of complex biological systems, may they be natural or artificial.

1.2 Fitness and Fitness Landscapes

A (static) fitness landscape Λ_S can be expressed by [51, 119]

$$\Lambda_S = (\mathbb{X}, n, f), \tag{1.1}$$

where $\mathbb{X}$ is a configuration space, $n(x)$ is a neighborhood structure that assigns to every $x \in \mathbb{X}$ a set of (more or less distant) neighbors, and $f(x) : \mathbb{X} \to \mathbb{R}$ is a fitness function that gives to every $x \in \mathbb{X}$ a proprietary quantity to be interpreted as a 'quality' information. In other words, the configuration space in connection with the neighborhood structure expresses a (possibly multi–dimensional) 'location', while the fitness is an orthogonal projection from location, defining an 'elevation' or 'height' and at the same time giving a location its most important property. Fitness is usually considered a single parameter but it seems perfectly possible to have a height measure with several dimensions.

The origin and character of a configuration space, neighborhood structure and fitness function differ, naturally, in evolutionary biology and evolutionary computation, and also in different application contexts in each of these fields. Moreover, in the literature on fitness landscapes in general, and also in the different chapters of this book, we find additional terms for the space over which the fitness landscape is defined. Next to the term configuration space, there is also genotypic space, sequence space, search space, or representation space. All these terms are synonymous in that they specify how the location is set out for which the fitness allocated is its constituting property. The terms differ in the origin and meaning of the location, and hence of the landscape itself. Configuration space can be seen as the most general term. It is made up of a finite or infinite number of configurations the genetic description of the natural or artificial biological systems can have. For natural biological systems, the genetic description is associated with the genotype, giving raise to the genotypic space. Sequence space can often be found in discussions of the relations between genotypes and fitness of (micro-)biological entities whose genetic specification and coding is based on DNA or RNA sequences, or on strings of their binary genomic equivalents. Search space is frequently used to underline the aspect of evolutionary search and evolutionary optimization, and it is hence mainly found in approaches to employ fitness landscapes to understand evolutionary computation. Representation space is often employed to characterize the situation where different representations of the evolutionary search algorithm redefine the search space and its neighborhood structure. In this view, the representation space is imposed by the way the search algorithm codes possible solutions. For the same configuration space, therefore, we can define several representation spaces.

In evolutionary biology, the configuration space consists of the genotypes of the biological system under study [55, 140]. The genotype characterizes the genetic make–up of a generic individual. It comprises of the sum (or union) of all genetically possible individuals and hence is the total genetic information. The neighborhood of a genotypic location is defined by which other genotypes are adjacent, near, close, accessible or reachable. Adjacency, nearness, or closeness and accessibility or reachability in a genetic context is measured by simple one–step evolutionary variations. These variations might be asexual, which specifies a mutational neighborhood, or sexual, which leads to a recombinational neighborhood. In a rather static view, expressed by the terms adjacency, nearness, and closeness, neighborhood implies which genotypes are next to each other. In a more dynamic view, expressed by the terms accessibility and reachability, neighborhood defines which genotypes can be explored from a given starting genotype by a possible one–step evolutionary trajectory. In other words, the dynamic properties intrinsic to evolutionary processes invoke a collection of possible transitions between genotypes. A succession of evolutionary transitions is also known as a *walk*.

Depending on the evolutionary variation considered (asexual or sexual) there may be a mutational trajectory or a recombinational trajectory. For describing mutational trajectories through genotypic space, Gillespie's strong selection/weak mutation (SSWM) model [42, 133] allows a simple mathematical treatment. This model assumes that the population is genetically monomorphic and considers that one–step transitions between genotypes take place as occasional random point mutations. SSWM further postulates that natural selection and genetic drift instantly lead either to fixation or elimination of the mutant genotype. As the configuration space that codes for natural biological genotypes is made up of strings over a finite alphabet, point mutations are flips in the letters of the alphabet at single points on the string. Hence, a mutational trajectory originates from subsequently executing one of the several possible single point changes in the string describing the genotype. As a consequence, the static and dynamic aspects of the genotypic space expressed by the notion of adjacency and accessibility, respectively, reflect each other.

For defining recombinatorial trajectories, the SSWM model is not suitable. To begin with, the evolutionary one–step transitions are not longer necessarily random, as the genetic material that undergoes recombination has been selected because of the high fitness of parental genotypes [133]. Also, polymorphism is required to generate recombinatorial evolutionary transitions. However, introducing polymorphism qualitatively changes the genotypic space and particularly the character of possible evolutionary transitions. As pointed out above, for mutational trajectories adjacency and accessibility are equivalent concepts. For recombinatorial trajectories, it is not. Recombination allows jumps through the genotypic space [133]. Moreover, it has been shown that for homologous recombination (exchanged between two similar or identical molecules of DNA) accessibility can be defined topologically similar to the adjacency of point mutation [43, 117, 118], but for non–homologous recombination the resulting genotypic space seems to be no longer metric [111, 114]. For these reasons, recombinatorial trajectories and hence recombinatorial fitness landscapes are an open research topic in theoretical biology. In evolutionary computation, mutational as well as recombinatorial landscapes have been intensively studied (see [46], but refer also to Chapter 13 of this book).

As pointed out above, the neighborhood structure is usually defined by the property of which genotypes can mutate from one to another [55] and genotypes are coded as a string over a finite alphabet. Most faithful to biological reality is the alphabet $\mathbb{A} = (A, T, C, G)$ of the DNA space or the RNA alphabet $\mathbb{A} = (A, U, C, G)$. Of course, strings using these alphabets can be recoded by using a binary alphabet $\mathbb{A} = (0, 1)$. This recasts the genotype as a binary genome and is also known as a digital genome approach. By fixing the length N of the string (or the number of base pairs), we obtain the configuration space $\mathbb{X} = \mathbb{A}^N$. A specific position on the string is called a locus. Each of the alternative values the locus (or several loci) can have is an allele. An instance of such a binary configuration space over the binary alphabet is a sequence (or string)

$$\mathbf{S} = (s_1 s_2, \ldots, s_i, \ldots, s_\ell), \tag{1.2}$$

which consist of ℓ loci with two possible alleles (0 or 1) at each locus s_i.

Assigning fitness to each element of the genotypic space requires additional considerations. Until recently, the question of what fitness a genotype actually has was answerable only purely theoretically and also required (and still requires) defining an intermediate level between genotype and fitness, i. e. the phenotypic space. The reason for that is that it is complicated or even infeasible to assign a fitness value to the ‘microscopic’ genotype. Fitness, at least in any sensible (marco–)biological sense, is connected to longevity and fertility and ultimately to reproduction success of a specific individual acting in a particular environment. Such a phenotypic individual can be thought of as an instance of the generic individual specified by a genotype. Hence, such a fitness landscape Λ_S is, strictly speaking, the product of a genotype–to–phenotype–to–fitness mapping and such landscapes have been the subject of much theoretical work on evolutionary dynamics [53, 104, 127, 133]. Anyway, assigning fitness to an actual biological (or even microbiological) entity is a highly debatable matter [137]. Given the understanding that fitness is the phenotype’s viability expressed as the fact of surviving to the age of reproduction and actually reproducing, it can only be assigned after the life time of the individual. Moreover, if higher fitness not only means a high number of offspring, but also offspring that for itself survives and reproduces disproportionally successful, fitness for a given phenotypic realization can only be accredited after observing a larger time window of (possibly overlapping) generations, while the environmental conditions are to remain the same. In short: assigning fitness is controversial. To circumvent these debates (and equal difficulties of how genotype maps to phenotype), it is standard in theoretical approaches to evolutionary biology to implicitly define fitness as an axiomatic property of the landscape.

In evolutionary computation, by contrast, the configuration space is made up by the search space obtained from encoding all possible solutions of the optimization problem. The neighborhood structure is a consequence of the search space and hence the objects to be optimized over, but also of the genetic operators the evolutionary search employs [51]. If the search space is metric (as for instance if the search space elements are real or integer numbers, and the genetic operators act on these numbers), then the neighborhood structure is inherent by the ordering of numbers. If the search space is not metric (or can have several different kinds of metrics), the neighborhood structure needs to be defined additionally. For examples, the neighborhood structure of binary coding can be a Hamming distance of different length, or the neighbors of tree representation are branches that differ by a (smaller or larger) variation in the subtrees. Fitness in evolutionary computation is usually also assigned via the associated optimization problem. As genotype codes for a possible solution, the problem's objective function for this solution can be calculated. This calculation may be as easy as inserting a variable into a mathematical formula, or as complicated as providing several parameters to a complex simulation model that in turn produces an output which needs to be evaluated. Anyway, at the end of the process there is a single and unique value that constitutes the genotype's fitness. If there is more than one value, they need to be aggregated (or treated as a multi–objective optimization problem). In case the value changes by being recalculated the fitness landscape is dynamic.

1.3 Fitness Landscapes in Evolutionary Biology

As considered in the previous sections, fitness landscapes are a geometrically motivated tool for discussing how the genetically possible (genotype), the actually realized (biological) 'life–form' (phenotype) and the survival/reproduction success (fitness) relate to each other. These relations immediately suggest the question of how an evolutionary development proceeds and hence evoke asking about the predictability of evolutionary paths. Up to most recently there were two major sources that fueled this discussion. A first is geometrical intuition and conception as in Wright's landscapes [140], a second is computational experiments with designed models for fitness landscapes as for instance with Kauffman's NK landscapes [55]; hence both are, in essence, theoretical. Lately, a third source appeared that is empirical by being based on biochemically altering biomolecular properties of organisms (or single genes of organisms) and assigning quality information as fitness to these properties. This approach using microbiological (microbial) experiments gives raise to empirical fitness landscapes [67, 96, 121]. In this view, empirical fitness landscapes connect molecular biology with evolutionary biology. These results have led to a renewed interest in the framework of fitness landscapes, as for the first time the question of the predictability of real evolutionary processes became addressable. Clearly, these three sources interactively drive recent advances in understanding fitness landscapes in evolutionary biology. Nevertheless, they historically and methodologically appeared in the given order, and are detailed in the following.

1.3.1 Topological Features of Fitness Landscapes

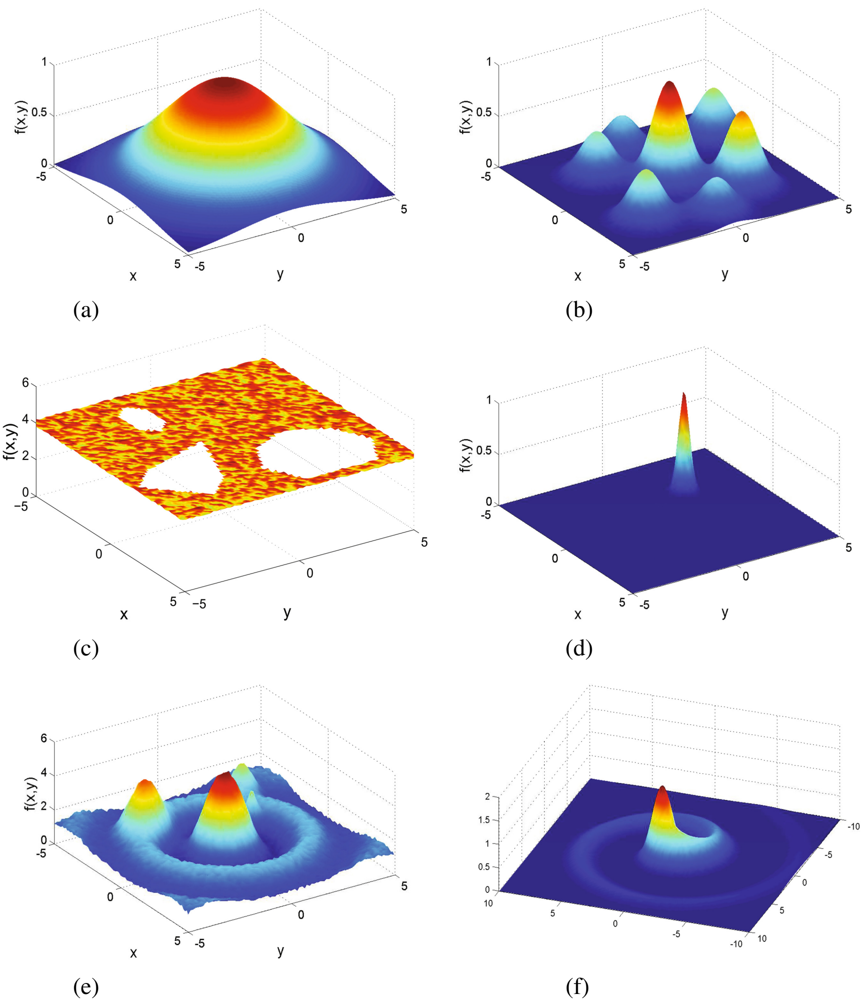
As discussed in the introduction, fitness landscapes are a geometrically motivated tool to visualize and evaluate how an evolving population may change over time.

The changes come into effect by one phenotypic realization of the genotype mutating into another phenotypic realization[4]. Hence, as mentioned before, genotypic locations in the landscape are distanced by the property that a single mutation leads from one genotype to another genotype. We (axiomatically or empirically) add that each genotype has a fitness and observe that the changes are driven by the Darwinian imperative to move into the direction of increasing fitness, which immediately imposes evolutionary dynamics. Note that so far we have a purely geometrical conception of how topological features of the landscape (the hills, valley, plains and ridges in the metaphor) result in potential for driving forces and dynamics (the lakes and flows in the landscape). By assuming that there are differences in fitness over genotypic space, the movement goes towards an aim. This defines possible evolutionary pathways that are directed and produces evolutionary outcomes that are more likely than others. The question, therefore, is not whether the topology and structure of a fitness landscape is a predictor of a likely evolutionary path, for on a conceptional level, the one defines the other. The question is whether the topology and structure of the fitness landscape at hand really matches the biological reality it supposes to study.

An important component in addressing this question is to collect which topological and structural features in a landscape are possible and allocate them to potential and observed mechanisms of evolutionary dynamics. For two–dimensional landscapes, there are a limited number of features. Figure 1.2 gives a catalog of possible topological features. The most basic feature is a single smooth peak as in Figure 1.2a. As discussed in Section 1.2 the landscape's topology imposes a succession of mutational transitions, which in turn define an evolutionary path with a corresponding evolutionary process. All paths of increasing fitness in Figure 1.2a go to the top. Hence, the corresponding process is evolutionary hill climbing. Due to the similarity to the same–named geographical landmark, a landscape with a single smooth peak as in Figure 1.2a is also known as a Fuji landscape (or Mt. Fuji landscape, or Fujiyama landscape). As a Fuji landscape consists of only one global optimum (and no other peaks where the evolutionary path could get stuck and hence trapped), it can also be consider as trap-free. To obtain a Fuji landscape, accessibility of possible mutational trajectories requires that the genotype's fitness increases monotonically with each mutation. In other words, each in a succession of mutations must be beneficial. It is an interesting question if this is a realistic assumption for a large group of biological systems.

A landscape with more complicated features has more than one peak and valleys between them. It is called a rugged landscape, as is shown in Figure 1.2b. A rugged landscape consists of a number of peaks that can be interpreted as local or global evolutionary optima. All peaks that are smaller than the largest one are local optima, which can be interpreted as traps. The degree of ruggedness may vary, usually with the number and the distribution of the peaks. An interesting question here is whether the peaks are distributed more or less uniformly in the landscape, or whether they rather group in distinct regions. The latter case is also known as the massif central

[4] For reasons given in Section 1.2, we restrict our discussion on mutational trajectories.



(a) (b) (c) (d) (e) (f)

Fig. 1.2 Geometrical features of fitness landscapes in 2D. (a) Single smooth peak (Mt. Fuji landscape), (b) Rugged landscape with multiple peaks, (c) Holey landscape, (d) Neutral landscape with single narrow peak, (e) Barrier landscape, (f) Detour landscape (long path problem).

hypothesis. While a minimum degree of ruggedness can be defined as a smooth Fuji landscape, a maximum degree of ruggedness is a landscape where the fitness of each of the neighbors of each genotype in the landscape is an independent (uncorrelated) realization of a random variable. This is known as the house–of–cards model of landscapes [53, 60]. The term comes from the understanding that a mutational transition from a functional (high–fitness) genotype is comparable to pulling a card from a house of cards, namely that the outcome is purely random. A single mutation may malfunction the phenotypic realization of the genotype (proverbially collapse the house of cards), and therefore eventuate in a low fitness value, or it may have no or little effect. Anyway, the result is unpredictable. Hence, fitness over a genotypic space are independent realizations of a random variable with a given distribution.

A rugged landscape also introduces a hill climbing evolutionary process, but whether a smaller local optimum is reached or the larger global optimum depends on the point in genotypic space where the walk starts from. All the genotypic starting points that evoke evolutionary paths going uphill to a local or global optimum form the optimum's *basin of attraction*. This sensitivity of the evolutionary outcome on starting points immediately poses a much more fundamental question. How can a walk that is stuck on a hill escape that local optimum every again? As this apparently requires a transition that loses fitness, such evolutionary paths appear to be unaccessible. The evolutionary process that may allow escaping local optima is known as valley crossing. The problem of valley crossing is long–standing and fundamental in evolutionary biology. It could even be said that without convincing solutions it is difficult to keep fitness landscape arguments alive. There are two types of approaches to the valley crossing problem. A first tries to construct evolutionary processes that circumvent the fact that natural selection does not tolerate deleterious mutations. A second group even questions if there are indeed valleys in relevant fitness landscapes that need to be crossed.

The first group of arguments for the valley crossing problem has classical roots. Wright's shifting balance theory [139, 141] suggests that the needed process to cross a valley in a fitness landscape starts with an exploratory phase characterized by random genetic drift. This leads to population subdivision with the subdivided small populations (with small genetic variance) randomly exploring neighboring genotypes. In other words, each of the small populations acts as a scout in genotypic space. If they encounter superior fitness peaks, they hill climb these peaks. Once established on a peak of higher fitness, migrants are send off that cause the other subpopulations to shift, eventually collecting the whole population on the higher peak and thus completing the valley crossing. An alternative, equally classical explanation comes from Fisher [34] (also see [129] for a further discussion). Fisher argues that valley crossing is mainly unnecessary as ecological and geological factors would dynamically change environmental conditions frequently, thus regularly recasting the hills–and–valleys structure of the fitness landscape. In other words, a population on a local peak could simple sit and wait until the peak becomes the starting point of an accessible uphill path to higher fitness. If, on the other hand, the local peak becomes a deep valley, this may simply lead to extinction. In this view, any realistic fitness landscape is a dynamic fitness landscape.

Next to these classical solutions of the valley crossing problem, there are further possible mechanisms. For instance, the crossing could be contrived by compensatory mutations [59, 134]. The compensation happens by the co–occurring (or consecutive occurring) of two mutations, each of which is individually deleterious or neutral, but combined are beneficial enough to bridge the valley. This process is closely related to *sign epistasis* [133]. Epistasis in general refers to the effect that the fitness of a genotype after a mutational transition depends not only on the genotypes before and after the transition, but also on the genetic context in which this transition takes place. Sign epistasis in particular means that the genetic context decides whether a mutation is beneficial or deleterious. In other words, sign epistasis is a property of the fitness landscape and a prerequisite for the landscape's ruggedness [63, 97] and hence the need for valley crossing. Compensatory mutations describe the order of mutations and hence the process by which the landscape's rugged features are transversed.

The second type of argument to the valley crossing problem, namely that valley structures in fitness landscapes are not frequent and therefore crossing is not necessary, decomposes into two further variants. One is to show that fitness landscapes of relevant biological reality have no valleys as they are trap–free in the sense that they resemble Fuji landscapes. There is some evidence for trap-freeness from empirical fitness landscapes discussed below and from other applicational fields (see for instance Chapter 2 for further discussion). The second variant maintains that all (or at least a large proportion of) high fitness peaks are connected by ridges of equal (or nearly equal) fitness. Mutational trajectories could evolve along these ridges. This is known as holey fitness landscapes [40, 41] (see also Figure 1.2c). This model holds that landscapes, particularity for higher dimensions, are rather holey than rugged. This assessment relies upon three postulates. Firstly, the landscape is mostly formed by networks of genotypes with similar high fitnesses. Secondly, potential lower fitness regions are not targeted by mutational transitions because of the deleterious effects these mutations would have. Thirdly, even higher fitness is hardly found in the landscape or it is difficult to access, for instance because of barriers or detours. Both types of regions apart from the network of similarity are hence hardly traversed and therefore appear as embedded holes. There is some affirmation for holey landscapes from analytical [41] and computational [88] models, but also substantial criticism. In particular, holey landscape conjectures that for the dimension of the landscape getting higher, more peaks are connected by ridges, thus reducing the relative number of peaks. Computational experiments with the NK landscape have not confirmed this [84] (also see Chapter 18). Apart from these mechanisms to accomplish valley crossings by mutational pathways, there are strong arguments that sexual genetic variation and recombination give even more meaningful potential to escape from local optima (see for instance [80, 130] for further discussions).

On a conceptional level, holey landscapes are closely related to neutral fitness landscapes (see Figure 1.2d). Here, almost all fitness values are equal (or nearly equal). Hence, mutational transitions happen, but have no effect in terms of change in fitness. This evolutionary process is also called neutral drift. For landscapes over a configuration space with a finite number of configurations, mutually accessible genotypes of equal fitness form a neutral network. Evolutionary dynamics takes place by population drift along these neutral networks. If peaks define aims for evolutionary developments, drift is seemingly aimless. However, this appears to be an oversimplification. Experiments with natural and artificial systems have demonstrated that there are frequently long periods of evolutionary movement that reveal no or little progress. On the other hand, these periods of drift are interrupted by sporadic, sudden and substantial increases in fitness. In other words, we have rare leaps between one level of neutrality and another. Recent results both in evolutionary biology as well as evolutionary computation have shown that neutrality is a common feature in fitness landscapes [102, 125, 128]. On a theoretical level these works are for instance Kimura's neutral theory of molecular evolution [58] or Eigen und Schuster's works on molecular quasispecies [27, 28, 29]. Moreover, neutrality plays an important role in RNA structures [101, 103], but also in evolutionary robotics [52, 112, 113] (also see Chapters 8 and 9). Figure 1.2d not only contains a neutral region, but also another landscape feature that makes finding the optimum very difficult: a single narrow peak with virtually no surrounding of increasing fitness serving as its basin of attraction. This is also called isolation or a needle–in–the–haystack–problem. As there is no information guiding the evolutionary process, there is no advantage compared to a purely random search.

There are further geometrical features of fitness landscapes that can be seen as obstructions to the hill climbing process. One is barriers (see Figure 1.2e). A barrier encloses and separates local or global optima and prevents uphill evolutionary paths that could connect them [35, 125]. In terms of the geometrical concepts, barriers combine valleys and basins of attraction. In this sense, valleys surround peaks, which in turn possess basins of attraction and barriers encloses these basins of attraction. In other words, barrier partition genotypic space in the peak's basins of attraction. Why this is an obstruction to hill climbing becomes understandable by looking at Figure 1.2e. If the high peak in the center is surrounded by a circular valley and another hill-like toroidal structure, this toroidal structure acts as a barrier for the high peak for all evolutionary paths that start from outside the barrier. Of course, some paths starting from the inner valley may be uphill walks to the high peak. On the other hand, paths beginning from some other points in the inner valley and from outside the barrier will climb the toroid, which is a local optimum. To now reach the high peak is difficult as it requires a valley crossing. For potential solutions to the valley crossing problem, see the discussion above.

Another geometrical feature that hinders uphill paths is a detour landscape [95, 96], which is also known as the long path problem [45] (see Figure 1.2f). The process of hill climbing is exacerbated by the path of increasing fitness being small, exclusive (illustrated in the graph by the Archimedean spiral in genotypic space), and sensitive to perturbations. For a mutational trajectory, these properties of long path problems mean that a walk has to find a precise sequence of mutations, which is difficult or takes a considerable amount of time. In evolutionary computation, it has been shown [39, 107] that this problem can be tackled by tailored mutation schemes.

1.3.2 Computational and Empirical Landscapes

Figure 1.2 provides a catalog of possible topological features in fitness landscapes. These pictures as instructive as they might be, at the same time also clearly show the limitation of geometrical conceptualization: it is bound to two dimensions. So, the natural question to ask is if these topological features and the conclusions drawn for evolutionary pathways and predictable outcomes will remain valid if the genotypic space is higher than two dimensions. From what we know now it seems that this is not answerable as a yes–no question. It rather appears that the features are present for all dimensions but increasing dimension may change what the features mean. This becomes geometrically imaginable by going down in dimension and considering one-dimensional slices through the two-dimensional genotypic space in Figure 1.2. For instance it is straightforward to see that the concepts of single peaks, ruggedness and neutrality have the same meaning in one-dimensional genotypic spaces as they have for two-dimensional landscapes. The geometrical features imply the same transitional movements. On the other hand, ridges between high fitness peaks are not possible in one-dimensional landscapes. The same applies for long-path problems. So, it makes sense to ask if the two-dimensional picture of a single peaked, or hilly and rugged, or flat and neutral landscape is relevant for higher dimension. Answers to this question can not come form considering what is geometrically imaginable. We need to observe the geometry of (higher–dimensional) computationally created landscapes and observe how empirical landscapes extracted from biological processes with real informational (information–carrying) molecules and processes look like.

There is a large number of publications with results from experiments with computationally created fitness landscapes. On the one hand, there are the fitness landscapes associated with optimization problems, which can be called problem–induced landscapes. Alternatively, there are artificially designed landscapes. Particularly for combinatorial optimization, problems-induced fitness landscapes have been analyzed intensively. Examples are landscapes for the traveling salesman problem (TSP) [78, 116], graph partitioning [76], graph coloring [13], quadratic assignment [23, 77], MAX-SAT [100] or the knapsack problem [122]. Although these studies have the main focus of giving information for evolutionary solving the problem most effectively, the results also showed a large variety of landscape features as cataloged in Figure 1.2. For instance, for the TSP it was demonstrated that the symmetric TSP (traveling cost from point i to point j are the same as from j to i) is rather smooth, while the asymmetric TSP is highly rugged [102, 115, 116], while for the quadratic assignment problem we obtain a rather flat landscape with a significant degree of neutrality [5, 23] (see also Chapter 9). For continuous (real-valued) problems, similar numerical results were obtained for a wide class of benchmark problems [14, 70] (also see Chapter 4). All these results have shown that usually the landscape features do not appear as disjunct as in the conceptualization and often a mix of features can be observed. This mix of features and therefore a blend of ingredients for search hardness becomes plausible by considering the no free lunch (NFL) theorem [21, 25, 138]. The theorem states that over all search problems, there is no algorithm that performs better than others. If search hardness is expressed by landscape features, then over all fitness landscapes, we should have a blend of features. Also, for problem-induced landscapes, there is no clear relation between features and dimension. For some problems, the features become more prominent as dimension increases, for others not. This is one reason for attempts to capture the essence of landscape features and their interplay in so–called landscape measures, refer to [71] for a comprehensive recent survey and outlook on open issues. Next to these problem-induced landscapes, there are specifically designed landscapes.

Most notable and almost predominant among artificially designed computational landscapes is Kauffman's NK landscape [53, 54, 55] (also see [4, 84, 123, 128] and the Chapters 9 and 18 for discussions of some recent results). The model has two parameters, N and K. The parameter N is the string length of a string defined according to Equation (1.2) and hence sets the dimension of the genotypic space. The parameter K describes the degree of epistatic interactions among the loci s_i of the string $\mathbf{S}$. The epistatic interactions are specified by setting K neighbors for each loci s_i via a neighborhood function $n(s_i,K)$ and defining a fitness contribution $f_i(s_i,n(s_i,K))$ for each locus s_i and the K neighbors. The fitness $f(\mathbf{S})$ of a genotype $\mathbf{S}$ is calculated by

$$f(\mathbf{S}) = \frac{1}{N}\sum_{i=0}^{N-1} f_i(s_i,n(s_i,K)). \tag{1.3}$$

The fitness contributions $f_i(s_i,n(s_i,K))$ are defined as realizations of a random variable uniformly distributed on the interval $[0,1]$ and hence also depend on the neighborhood and thus on the interaction parameter K. By varying the interaction parameter K between $0 \leq K \leq N-1$, different degrees of ruggedness can be adjusted. It can be shown that for $K=0$ we obtain a smooth single peaked Fuji like landscape, while for $K=N-1$ we have maximal ruggedness as in a house–of–cards landscape. The NK landscape can be modified to include neutrality, which is called NKp landscapes [9]. For the NK landscape there also exists some analytic results regarding the number of local optima [26, 56].

Another computational model of fitness landscapes is the rough Mt. Fiji (RMF) model [3, 37, 121]. This model has three parameters: a reference string $\mathbf{S}_0$, a drift parameter c and a random distribution η. The fitness $f(\mathbf{S})$ of a genotype $\mathbf{S}$ is

$$f(\mathbf{S}) = \eta(\mathbf{S}) - c \cdot d_H(\mathbf{S},\mathbf{S}_0), \tag{1.4}$$

where $d_H(\mathbf{S},\mathbf{S}_0)$ is the Hamming distance between genotype $\mathbf{S}$ and reference genotype $\mathbf{S}_0$, and $\eta(\mathbf{S})$ is a realization of the random variable η that is for each genotype $\mathbf{S}$ in the landscape independent from the realizations of all the other genotypes. If we set the drift parameter c to zero, we obtain the house–of–cards scenario, where each fitness is random and independent from and uncorrelated to the fitness of neighboring genotypes. We hence have maximum ruggedness. For values of c becoming larger, we pass through different degrees of ruggedness which finally leads to a Fuji landscape with a peak at the reference genotype $\mathbf{S}_0$.

A third alternative artificially designed landscape is the block model [86, 90]. It also considers strings $\mathbf{S}$ defined according to Equation (1.2) with length N and assumes that the string is divided into B blocks of equal length $\mu = \frac{N}{B}$ and $1 \leq \mu \leq N$. Each block epistatically interacts with other blocks and delivers a fitness distribution $f_i(\mu)$, $i = 1,2,\ldots,B$ which is independently taken from a random variable uniformly distributed on some finite interval. The fitness $f(\mathbf{S})$ of the string $\mathbf{S}$ is the sum of the block fitness distributions:

$$f(\mathbf{S}) = \sum_{i=1}^{B} f_i(\mu). \tag{1.5}$$

Similar to the NK landscape of Equation (1.3), the fitness correlation between neighboring strings (and hence the ruggedness) can be tuned; for the block model in Equation (1.5) by the length μ of the blocks. For $\mu = 1$, the landscape is maximally random and has maximum ruggedness. For $\mu = N$, the landscape is smooth and has a single global optimum. For intermediate values of μ, different degrees of ruggedness can be composed. Despite the similarity between the block model and the NK model, it has been pointed out that the block model has an interesting statistical property that the NK model has not [86, 90]. Because the NK model calculates fitness as an average of N contributions, its fitness distribution becomes normal due to the central limit theorem effect. The variance of this distribution decreases as N increases. This effect is also observable in the block model for the block number B growing large, but much weaker than in the NK model. The landscape models given with Equations (1.3), (1.4) and (1.5) are artificially designed to have the landscape features discussed in Section 1.3.1 and even allow tuning important aspects such as ruggedness and neutrality. This gives raise to tunable fitness landscapes. Hence, these artificially designed landscapes can be used as theoretical background and comparison for empirical studies of fitness landscapes.

An empirical approach involves to carry out step–wise mutational evolution of molecular functions. Hence, molecular evolution allows measuring the fitness effect that experimentally induced mutations have [67, 68, 96]. For instance, such experiments have been done for catalytic RNA [93], viruses such as *HIV-1* [62], bacteria such as *methylobacterium extorquens* [15], *β lactamase* [67, 96], *escherichia coli* [11, 31, 61] and *escherichia coli lac* [87], and fungi such as *aspergillus niger* [22, 37] and *saccharomyces cerevisiae* [44, 63].

According to a recent survey of empirical fitness landscapes by Szendro et al. [121], studies of microbiological evolution can be categorized into three groups. A first group comprises of studies that focus on establishing local ruggedness of landscapes by observing the repeatability of adaption in evolution experiments and a second group intends to detect sign epsistais between mutations to conclude local ruggedness. In other words, the works in these two groups of publications observe evolutionary processes and draw conclusions regarding topological features of the underlying fitness landscape. A third group tries to evaluate the fitness (or a fitness proxy) for all 2^{ℓ} combinations of a small set of ℓ mutations. For the fitness landscapes in the third group, qualitative measures of the landscape were calculated [121]. These qualitative measures are statistical landscape measures that show ruggedness and epistasis and are, in principle, suitable to match empirical landscape to landscape features. The results have shown that only a small number of the empirical landscapes available now allow for some (rather preliminary) conclusions. For instance, the landscapes of *methylobacterium extorquens* and *β lactamase* appear rather smooth, *escherichia coli lac* even Fuji–like single peaked, while the landscapes for *HIV-1*, *aspergillus niger* and *saccharomyces cerevisiae* are much more rugged. As pointed out by Szendro et al. [121], this might be caused by differing aims in the underlying studies. Works that studied deleterious mutations tended to obtain more rugged landscapes, while beneficial mutations yielded smoother landscapes.

In conclusion, studies of empirical fitness landscapes still only cover a small fraction of the space of possibilities that the genetic coding of even very simple forms of life such as viruses, bacteria and fungi offers. One of the reasons for difficulties in mapping more extended empirical landscapes is that the fitness effect of a mutation may be subject to epistasis [91]. Epistasis refers to the fact that the fitness associated with a mutation depends on the genetic context, that is the presence or absence of other mutations. Epistasis renders generating empirical fitness landscape combinatorially complex as the fitness effect of a mutation on one locus is influenced by the interaction of (possibly a larger number of) other loci [37]. Measuring these effects goes to the limits of current capacities in sequencing technology. However, extending our knowledge about empirical landscapes is not only promising, but also mandatory for clarifying the value of fitness landscape for providing a framework to predict evolutionary paths and explain evolutionary development – general scientific understanding is that the value of a theoretical framework can only be decided by real–world experiments.

1.4 Concluding Remarks

In this chapter, fundamental concepts and notions of fitness landscapes have been studied. Based on a brief discussion of the origin of landscape paradigms, major motivations to use fitness landscapes have been presented and mathematical descriptions of fitness landscapes have been introduced. Fitness landscapes bring together genotype, phenotype and fitness and establish relationships between these three quantities. Differences in fitness over genotypic or phenotypic space result in non–trivial fitness landscapes featuring the peaks, valleys, ridges and plateaus known from two–dimensional visualizations of the landscape metaphor. These topological features, in turn, define evolutionary pathways that a succession of evolutionary transitions (mutational or recombinatorial or otherwise) is likely to follow. Hence, the structure and topology of a fitness landscape defines potential for evolutionary dynamics and is a predictor of evolutionary outcomes. In this chapter, it has been discussed that establishing such predictions is interesting in evolutionary biology as well as in evolutionary computation. Notably by using computational and empirical landscapes, questions such as the likelihood of a particular evolutionary process (as in evolutionary biology) or the expected performance and behavior of an evolutionary search algorithm (as in evolutionary computation) become addressable.

Static fitness landscapes as discussed in this chapter and formally introduced by Equation (1.1) are a rather straightforward model of the forces underlying evolutionary dynamics, but a necessary question is whether the model accurately describes the relationships between genotypes, phenotypes and fitness in the real world. Answering this question is outside the scope of this introductory chapter, which has been focused on the implications of a given landscape. A first issue is that at least a considerable number of realistic fitness landscapes are not static in the sense that the relationship between genotype, phenotype and fitness is constant over evolutionary time. Part III of this book in entirely concerned with landscapes that are dynamic, particularly refer to Chapter 10 for an introduction to and overview on dynamic fitness landscapes. A second issue is if the picture of 'one genotype, one phenotype, one fitness value' painted by the static landscape metaphor is really sufficient. Recent experimental and theoretical findings about information transfer in biological systems indicate otherwise. For a discussion about the second issue and an outlook to future questions and issues of fitness landscapes refer to Chapter 19 of this book.

References

[1] Adami, C.: Information theory in molecular biology. Phys. Life Reviews 1, 3–22 (2004)

[2] Adami, C.: The use of information theory in evolutionary biology. Annals NY Acad. Sciences 1256, 49–65 (2012)

[3] Aita, T., Husimi, Y.: Adaptive walks by the fittest among finite random mutants on a Mt. Fuji-type fitness landscape. II. Effect of small non–additivity. J. Math. Biol. 41, 207–231 (2000)

[4] Aita, T.: Hierarchical distribution of ascending slopes, nearly neutral networks, highlands, and local optima at the dth order in an NK fitness landscape. J. Theor. Biology 254, 252–263 (2008)

[5] Angel, E., Zissimopoulos, V.: On the landscape ruggedness of the quadratic assignment problem. Theor. Comp. Sci. 263, 159–172 (2001)

[6] Ao, P.: Laws in Darwinian evolution. Phys. Life Reviews 2, 117–156 (2005)

[7] Bäck, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: comments on the history and current state. IEEE Trans. Evolut. Comput. 1, 3–17 (1997)

[8] Banavali, N.K., Roux, B.: Free energy landscape of A–DNA to B–DNA conversion in aqueous solution. J. Amer. Chem. Soc. 127, 6866–6876 (2005)

[9] Barnett, L.: Ruggedness and neutrality - the NKp family of fitness landscapes. In: Adami, C., Belew, R.K., Kitano, H., Taylor, C. (eds.) Proc. Artificial Life VI, pp. 18–27. MIT Press, Cambridge (1998)

[10] Baskaran, S., Stadler, P.F., Schuster, P.: Approximate scaling properties of RNA free energy landscapes. J. Theor. Biology 181, 299–310 (1996)

[11] Beerenwinkel, N., Pachter, L., Sturmfels, B., Elena, S.F., Lenski, R.E.: Analysis of epistatic interactions and fitness landscapes using a new geometric approach. BMC Evolutionary Biology 7, 60 (2013),
`http://www.biomedcentral.com/1471-2148/7/60`
(retrieved April 15, 2013)

[12] Binder, K., Young, A.P.: Spin glasses: Experimental facts, theoretical concepts, and open questions. Rev. Mod. Phys. 58, 801–976 (1986)

[13] Bouziri, H., Mellouli, K., Talbi, E.G.: The k-coloring fitness landscape. J. Comb. Optim. 21, 306–329 (2011)

[14] Caamaño, P., Bellas, F., Becerra, J.A., Duro, R.J.: Evolutionary algorithm characterization in real parameter optimization problems. Applied Soft Computing 13, 1902–1921 (2013)

[15] Chou, H.H., Chiu, H.C., Delaney, N.F., Segré, D., Marx, C.J.: Diminishing returns epistasis among beneficial mutations decelerates adaptation. Science 332(6034), 1190–1192 (2011)

[16] Chyba, C.F., Philipps, C.B.: Europa as an abode of life. Orig. Life Evol. Biosph. 32, 47–67 (2002)

[17] Cleland, C.E.: Life without definitions. Synthese 185, 125–144 (2012)

[18] Cleland, C.E., Chyba, C.F.: Defining life. Orig. Life Evol. Biosph. 32, 387–393 (2002)

[19] Cohen, J.E.: Mathematics is biology's next microscope, only better; Biology is mathematics' next physics, only better. PLoS Biol. 2(12), e439 (2004), doi:10.1371/journal.pbio.0020439

[20] Crow, J.F., Kimura, M.: Introduction to Population Genetics Theory. Blackburn Press, Caldwell (2009)

[21] Culberson, J.C.: On the futility of blind search: An algorithmic view of "no free lunch". Evolut. Comput. 6, 109–127 (1998)

[22] de Visser, J.A.G.M., Park, S.C., Krug, J.: Exploring the effect of sex on empirical fitness landscapes. The American Naturalist 174, S15–S30 (2009)

[23] Daolio, F., Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of the quadratic assignment problem. In: Fogel, G., Ishibuchi, H. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2010, pp. 1–8. IEEE Press, Piscataway (2010)

[24] Dobzhansky, T.: Biology, molecular and organismic. American Zoologist 4, 443–452 (1965)

[25] Droste, S., Jansen, T., Wegener, I.: Optimization with randomized search heuristics–the (A)NFL theorem, realistic scenarios, and difficult functions. Theor. Comp. Sci. 287, 131–144 (2002)

[26] Durrett, R., Limic, V.: Rigorous results for the NK model. Ann. Probability 31, 1713–1753 (2003)

[27] Eigen, M.: Selforganization of matter and the evolution of biological macromolecules. Die Naturwissenschaften 58, 465–523 (1971)

[28] Eigen, M., Schuster, P.: The Hypercycle: A Principle of Natural Self-Organization. Springer, Berlin (1979)

[29] Eigen, M., McCaskill, J., Schuster, P.: The molecular quasi-species. Adv. Chem. Phys. 75, 149–263 (1989)

[30] Egbert, M.D., Barandiaran, X.E., Di Paolo, E.A.: Behavioral metabolution: the adaptive and evolutionary potential of metabolism–based chemotaxis. Artificial Life 18, 1–25 (2012)

[31] Elena, S.F., Lenski, R.E.: Test of synergistic interactions among deleterious mutations in bacteria. Nature 390, 395–398 (1997)

[32] Ewens, W.J.: Mathematical Population Genetics. Springer, Berlin (2004)

[33] Fernando, C., Rowe, J.: The origin of autonomous agents by natural selection. Biosystems 91, 355–373 (2008)

[34] Fisher, R.A.: The Genetical Theory of Natural Selection. Clarendon Press, Oxford (1930)

[35] Flamm, C., Hofacker, I.L., Stadler, P.F., Wolfinger, M.T.: Barrier trees of degenerate landscapes. Z. Phys. Chem. 216, 1–19 (2002)

[36] Fogel, D.B.: Evolutionary Computation: The Fossil Record. IEEE Press, Piscataway (1998)

[37] Franke, J., Klözer, A., de Visser, J.A.G.M., Krug, J.: Evolutionary accessibility of mutational pathways. PLoS Comput Biol 7(8), e1002134 (2011), doi:10.1371/journal.ci.1002134

[38] Frauenfelder, H., Leeson, D.T.: The energy landscape in non–biological and biological molecules. Nature Structural Biology 5, 757–759 (1998)

[39] Garnier, J., Kallel, L.: Statistical distribution of the convergence time of evolutionary algorithms for long-path problems. IEEE Trans. Evolut. Comput. 4, 16–30 (2000)

[40] Gavrilets, S., Gravner, J.: Percolation on the fitness hypercube and the evolution of reproductive isolation. J. Theor. Biology 184, 51–64 (1997)

[41] Gavrilets, S.: Fitness Landscapes and the Origin of Species. Princeton University Press, Princeton (2004)

[42] Gillespie, J.H.: Molecular evolution over the mutational landscape. Evolution 38, 1116–1129 (1984)

[43] Gitchoff, P., Wagner, G.P.: Recombination induced hypergraphs: a new approach to mutation-recombination isomorphism. Complexity 2, 37–43 (1996)

[44] Hall, D.W., Agan, M., Pope, S.C.: Fitness epistasis among 6 biosynthetic loci in the budding yeast Saccharomyces cerevisiae. Journal of Heredity 101, S75–S84 (2010)

[45] Horn, J., Goldberg, D.E., Deb, K.: Long path problems. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN III 1994. LNCS, vol. 866, pp. 149–158. Springer, Heidelberg (1994)

[46] Jones, T.: Evolutionary algorithms, fitness landscape and search. PhD thesis, The University of New Mexico, Albuquerque (1995)

[47] Joseph, J., Schuster, G.B.: Emergent functionality of nucleobase radical cations in duplex DNA: Prediction of reactivity using qualitative potential energy landscapes. J. Amer. Chem. Soc. 128, 6070–6074 (2006)

[48] Joyce, G.F.: Foreword. In: Deamer, D.W., Fleischaker, G.R. (eds.) Origins of Life: The Central Concepts, pp. xi–xii. Jones and Bartlett, Boston (1994)

[49] Joyce, G.F.: The antiquity of RNA–based evolution. Nature 418, 214–221 (2002)

[50] Kant, E.: Metaphysische Anfangsgründe der Naturwissenschaft. Johann Friedrich Hartknoch, Riga (1786)

[51] Kallel, L., Naudts, B., Reeves, C.R.: Properties of fitness functions and search landscapes. In: Kallel, L., Naudts, B., Rogers, A. (eds.) Theoretical Aspects of Evolutionary Computing, pp. 177–208. Springer, Heidelberg (2001)

[52] Katada, Y., Ohkura, K.: Estimating the degree of neutrality in fitness landscapes by the Nei's standard genetic distance – An application to evolutionary robotics. In: Yen, G.G., Lucas, S.M., Fogel, G., Kendall, G., Salomon, R., Zhang, B.T., Coello Coello, C.A., Runarsson, T.P. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2006, pp. 483–490. IEEE Press, Piscataway (2006)

[53] Kauffman, S.A., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. J. Theor. Biology 128, 11–45 (1987)

[54] Kauffman, S.A., Weinberger, E.D.: The NK Model of rugged fitness landscapes and its application to the maturation of the immune response. J. Theor. Biology 141, 211–2445 (1989)

[55] Kauffman, S.A.: The Origin of Order: Self–Organization and Selection in Evolution. Oxford University Press, New York (1993)

[56] Kaul, H., Jacobson, S.: Global optima results for the Kauffman NK model. Mathematical Programming 106, 319–338 (2006)

[57] Keller, E.F.: A clash of two cultures. Nature 445, 603 (2007)

[58] Kimura, M.: The Neutral Theory of Molecular Evolution. Cambridge University Press, Cambridge (1983)

[59] Kimura, M.: The role of compensatory neutral mutations in molecular evolution. Journal of Genetics 64, 7–19 (1985)

[60] Kingman, J.: A simple model for the balance between selection and mutation. J. Appl. Probab. 15, 1–12 (1978)

[61] Khan, A.I., Dinh, D.M., Schneider, D., Lenski, R.E., Cooper, T.F.: Negative epistasis between beneficial mutations in an evolving bacterial population. Science 332(6034), 1193–1196 (2011)

[62] Kouyos, R.D., Leventhal, G.E., Hinkley, T., Haddad, M., Whitcomb, J.M., Petropoulos, C.J., Bonhoeffer, S.: Exploring the complexity of the HIV-1 fitness landscape. PLoS Genet. 8(3), e1002551, doi:10.1371/journal.pgen.1002551

[63] Kvitek, D.J., Sherlock, G.: Reciprocal sign epistasis between frequently experimentally evolved adaptive mutations causes a rugged fitness landscape. PLoS Genet. 7(4), e1002056 (2011), doi:10.1371/journal.pgen.1002056

[64] Hartl, D.L., Clark, A.G.: Principles of Population Genetics. Sinauer Associates, Sunderland (2007)

[65] van Hemmen, J.L.: Biology and mathematics: A fruitful merger of two cultures. Biol. Cybern. 97, 1–3 (2007)

[66] Lincoln, T.A., Joyce, G.F.: Self–sustained replication of an RNA enzyme. Science 323, 1229–1232 (2009)

[67] Lobkovsky, A.E., Wolf, Y.I., Koonin, E.V.: Predictability of evolutionary trajectories in fitness landscapes. PLoS Comput. Biol. 7(12), e1002302 (2011), doi:10.1371/journal.pcbi.1002302

[68] Lunzer, M., Miller, S.P., Felsheim, R., Dean, A.M.: The biochemical architecture of an ancient adaptive landscape. Science 310(5747), 499–501 (2005)

[69] Machery, E.: Why I stopped worrying about the definition of life..and why should you as well. Synthese 185, 145–164 (2012)

[70] Malan, K.M., Engelbrecht, A.P.: Quantifying ruggedness of continuous landscapes using entropy. In: Tyrrell, A. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2009, pp. 1440–1447. IEEE Press, Piscataway (2009)

[71] Malan, K.M., Engelbrecht, A.P.: A survey of techniques for characterising fitness landscapes and some possible ways forward. Information Science 241, 148–163 (2013)

[72] Manderick, B., de Weger, M., Spiessens, P.: The genetic algorithm and the structure of the fitness landscape. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 143–150. Morgan Kauffman, San Mateo (1991)

[73] Mayr, E.: The Growth of Biological Thought: Diversity, Evolution, and Inheritance. Belknap Press of Harvard University Press, Cambridge (1982)

[74] McCandlish, D.M.: Visualizing fitness landscapes. Evolution 65, 1544–1558 (2011)

[75] McGhee, G.R.: The Geometry of Evolution Adaptive Landscapes and Theoretical Morphospaces. Cambridge University Press, Cambridge (2006)

[76] Merz, P., Freisleben, B.: Memetic algorithms and the fitness landscape of the graph bi-partitioning problem. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN V 1998. LNCS, vol. 1498, pp. 765–774. Springer, Heidelberg (1998)

[77] Merz, P., Freisleben, B.: Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. IEEE Trans. Evolut. Comp. 4, 337–352 (2000)

[78] Merz, P.: Advanced fitness landscape analysis and the performance of memetic algorithms. Evolut. Comput. 12, 303–325 (2004)

[79] Mézard, M., Paresi, G., Virasoro, M.: Spin Glass Theory and Beyond. World Scientific, Singapore (1987)

[80] Mills, R., Watson, R.A.: On crossing fitness valleys with the Baldwin effect. In: Rocha, L., Yaeger, L., Bedau, M., Floreano, D., Goldstone, R., Vespignani, A. (eds.) Proc. Artificial Life X, pp. 493–499. MIT Press, Cambridge (2006)

[81] Ogata, S., Li, J., Yip, S.: Energy landscape of deformation twinning in bcc and fcc metals. Phys. Rev. B 71, 224102–224112 (2005)

[82] Onuchic, J.N., Luthey Schulten, Z., Wolynes, P.G.: Theory of protein folding: The energy landscape perspective. Ann. Rev. Phys. Chem. 48, 545–600 (1997)

[83] Onuchic, J.N., Wolynes, P.G.: Theory of protein folding. Curr. Opin. Struc. Biolog. 14, 70–75 (2004)

[84] Østman, B., Hintze, A., Adami, C.: Critical properties of complex fitness landscapes. In: Fellerman, H., Dörr, M., Hanczyc, M.M., Ladegaard Laursen, L., Maurer, S., Merkle, D., Monnard, P.A., Stoy, K., Rasmussen, S. (eds.) Proc. Artificial Life XII, pp. 126–132. MIT Press, Cambridge (2010)

[85] Orr, H.A.: The genetic theory of adaptation: a brief history. Nature Reviews Genetic 6, 119–127 (2005)

[86] Orr, H.A.: The population genetics of adaptation on correlated fitness landscapes: the block model. Evolution 60, 1113–1124 (2006)

[87] Otwinowski, J., Nemenman, I.: Genotype to phenotype mapping and the fitness landscape of the E. coli lac promoter. arXiv preprint (2013), `http://arxiv.org/abs/1206.4209`

[88] Paperin, G., Sadedin, S., Green, D., Dorin, A.: Holey fitness landscapes and the maintenance of evolutionary diversity. In: Bullock, S., Noble, J., Watson, R.A., Bedau, M.A. (eds.) Proc. Artificial Life XI, pp. 450–457. MIT Press, Cambridge (2008)

[89] Peng, B., Kimmel, M., Amos, C.I.: Forward–Time Population Genetics Simulations: Methods, Implementation, and Applications. Wiley–Blackwell, Hoboken (2012)

[90] Perelson, S.A., Macken, C.A.: Protein evolution on partially correlated landscapes. Proc. Natl. Acad. Sci. USA 92, 9657–9661 (1995)

[91] Phillips, P.C.: Epistasis–the essential role of gene interactions in the structure and evolution of genetic systems. Nature Reviews Genetics 9, 855–867 (2008)

[92] Pigliucci, M., Kaplan, J.: Making Sense of Evolution: The Conceptual Foundations of Evolutionary Biology. University of Chicago Press, Chicago (2006)

[93] Pitt, J.N., Ferré-D'Amaré, A.R.: Rapid construction of empirical RNA fitness landscapes. Science 330(6002), 376–379 (2010)

[94] Pitzer, E., Affenzeller, M.: A comprehensive survey of fitness landscape analysis. In: Fodor, J., Klempous, R., Suárez Araujo, C.P. (eds.) Recent Advances in Intelligent Engineering Systems. SCI, vol. 378, pp. 161–191. Springer, Heidelberg (2012)

[95] Poelwijk, F.J., Kiviet, D.J., Tans, S.J.: Evolutionary potential of a duplicated repressor-operator pair: Simulating pathways using mutation data. PLoS Comput. Biol. 2(5), e58 (2006), doi:10.1371/journal.pcbi.0020058

[96] Poelwijk, F.J., Kiviet, D.J., Weinreich, D.M., Tans, S.J.: Empirical fitness landscapes reveal accessible evolutionary paths. Nature 445, 383–386 (2007)

[97] Poelwijk, F.J., Tănase-Nicola, S., Kiviet, D.J., Tans, S.J.: Reciprocal sign epistasis is a necessary condition for multi-peaked fitness landscapes. J. Theor. Biology 272, 141–144 (2011)

[98] Polani, D.: Information: currency of life? HFSP Journal 3, 307–316 (2009)

[99] Provine, W.B.: Sewall Wright and Evolutionary Biology. University of Chicago Press, Chicago (1986)

[100] Prügel–Bennett, A., Tayarani–Najaran, M.H.: Maximum satisfiability: Anatomy of the fitness landscape for a hard combinatorial optimization problem. IEEE Trans. Evolut. Comp. 16, 319–338 (2012)

[101] Reidys, C., Stadler, P.F., Schuster, P.: Generic properties of combinatory maps: neutral networks of RNA secondary structures. Bull. Math. Biol. 59, 339–397 (1997)

[102] Reidys, C., Stadler, P.F.: Neutrality in fitness landscapes. Applied Mathematics and Computation 117, 321–350 (2001)

[103] Reidys, C.: Combinatorial Computational Biology of RNA: Pseudoknots and Neutral Networks. Springer, Heidelberg (2011)

[104] Richter, H.: Coupled map lattices as spatio–temporal fitness functions: Landscape measures and evolutionary optimization. Physica D237, 167–186 (2008)

[105] Rodrigues, E.S., de Oliveira, P.M.C.: Spin–glass energy landscape. Jour. Stat. Phys. 74, 1265–1272 (1994)

[106] Rowbottom, D.P.: Models in biology and physics: What's the difference? Found. Sci. 14, 281–294 (2009)

[107] Rudolph, G.: How mutation and selection solve long-path problems in polynomial expected time. Evolut. Comput. 4, 195–205 (1996)

[108] Ruse, M.: Are pictures really necessary? The case of Sewell Wright's 'adaptive landscapes'. In: Baigrie, B.S. (ed.) Picturing Knowledge: Historical and Philosophical Problems Concerning the Use of Art in Science, pp. 303–337. University of Toronto Press, Toronto (1996)

[109] Shakhnovich, E.I.: Theoretical studies of protein-folding thermodynamics and kinetics. Curr. Opin. Struc. Biolog. 7, 29–40 (1997)

[110] Sherrington, D.: Landscape paradigms in physics and biology: Introduction and overview. Physica D107, 117–121 (1997)

[111] Shpak, M., Wagner, G.P.: Asymmetry of configuration space induced by unequal crossover: implications for a mathematical theory of evolutionary innovation. Artificial Life 6, 25–43 (2000)

[112] Smith, T., Husbands, P., O'Shea, M.: Neutral networks in an evolutionary robotics search space. In: Kim, J.H., Zhang, B.T., Fogel, G., Kuscu, I. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2001, pp. 136–143. IEEE Press, Piscataway (2001)

[113] Smith, T., Husbands, P., Layzell, P., O'Shea, M.: Fitness landscapes and evolvability. Evolut. Comput. 10, 1–34 (2002)

[114] Stadler, B.M.R., Stadler, P.F., Wagner, G.P., Fontana, W.: The topology of the possible: Formal spaces underlying patterns of evolutionary change. J. Theor. Biology 213, 241–274 (2001)

[115] Stadler, P.F.: Landscapes and their correlation functions. J. Math. Chem. 20, 1–45 (1996)

[116] Stadler, P.F., Schnabl, W.: The landscape of the traveling salesman problem. Phys. Lett. A 161, 337–344 (1992)

[117] Stadler, P.F., Wagner, G.P.: The algebraic theory of recombination spaces. Evol. Computation 5, 241–275 (1998)

[118] Stadler, P.F., Seitz, R., Wagner, G.P.: Population dependent Fourier decomposition of fitness landscapes over recombination spaces: evolvability of complex characters. Bull. Math. Biol. 62, 399–428 (2000)

[119] Stadler, P.F., Stephens, C.R.: Landscapes and effective fitness. Comm. Theor. Biol. 8, 389–431 (2003)

[120] Suzuki, K., Ikegami, T.: Shapes and self–movement in protocell systems. Artificial Life 15, 59–70 (2009)

[121] Szendro, I.G., Schenk, M.F., Franke, J., Krug, J., de Visser, J.A.G.M.: Quantitative analyses of empirical fitness landscapes. Jour. Stat. Mech., P01005 (2013), doi:dx.doi.org/10.1088/1742-5468/2013/01/P01005

[122] Tavares, J., Pereira, F.B., Costa, E.: Multidimensional knapsack problem: a fitness landscape analysis. IEEE Trans. Sys. Man Cyber. B 38, 604–616 (2008)

[123] Tomassini, M., Verel, S., Ochoa, G.: Complex-network analysis of combinatorial spaces: The NK landscape case. Phys. Rev. E 78, 066114 (2008)

[124] Trifonov, E.N.: Vocabulary of definitions of life suggests a definition. J. Biomol. Struct. Dyn. 29, 259–266 (2011)

[125] van Nimwegen, E., Crutchfield, J.P.: Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths? Bull. Math. Biol. 62, 799–848 (2000)

[126] van Tendeloo, G., De Meulenaere, P., Schryvers, D.: Landscape roughness at an atomic scale. Physica D107, 401–410 (1997)

[127] Vassilev, V.K., Fogarty, T.C., Miller, J.F.: Information characteristics and the structure of landscapes. Evolut. Comput. 8, 31–60 (2000)

[128] Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of NK landscapes with neutrality. IEEE Trans. Evolut. Comput. 15, 783–797 (2011)

[129] Wade, M.J., Goodnight, C.J.: The theories of Fisher and Wright in the context of metapopulations: When nature does many small experiments. Evolution 52, 1537–1553 (1998)

[130] Watson, R.A., Weinreich, D.M., Wakeley, J.: Genome structure and the benefit of sex. Evolution 65, 523–536 (2011)

[131] Weinberger, E.D.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biol. Cybern. 63, 325–336 (1990)

[132] Weinberger, E.D.: Local Properties of Kaufman's N–k model: A Tunably Rugged Energy Landscape. Phys. Rev. A 44, 6399–6413 (1991)

[133] Weinreich, D.M., Watson, R.A., Chao, L.: Sign epistasis and constraint on evolutionary trajectories. Evolution 59, 1165–1174 (2005)

[134] Weissman, D.B., Desai, M.M., Fisher, D.S., Feldman, M.W.: The rate at which asexual populations cross fitness valleys. Theor. Pop. Biol. 75, 286–300 (2009)

[135] Wigner, E.: The unreasonable effectiveness of mathematics in the natural sciences. Comm. Pure Appl. Mathem. 13, 1–14 (1960)

[136] Wiles, J., Tonkes, B.: Hyperspace geography: Visualizing fitness landscapes beyond 4D. Artificial Life 12, 211–216 (2006)

[137] Wolf, J.B., Wade, M.J.: On the assignment of fitness to parents and offspring: whose fitness is it and when does it matter? Jour. Evolut. Biol. 14, 347–356 (2001)

[138] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evolut. Comp. 1, 67–82 (1997)

[139] Wright, S.: Evolution in Mendelian populations. Genetics 16, 97–159 (1931)

[140] Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones, D.F. (ed.) Proc. of the Sixth International Congress on Genetics, pp. 356–366 (1932)

[141] Wright, S.: The shifting balance theory and macroevolution. Annual Review of Genetics 16, 1–19 (1982)

[142] Wright, S.: Surfaces of selective value revisited. The American Naturalist 131, 115–123 (1988)

[143] Zhou, Q., Wong, W.H.: Energy landscape of a spin-glass model: Exploration and characterization. Phys. Rev. E 13, 051117–1–13 (2009)

Chapter 2
Fundamental Principles of Control Landscapes with Applications to Quantum Mechanics, Chemistry and Evolution

Herschel Rabitz, Re–Bing Wu, Tak–San Ho, Katharine Moore Tibbetts,
and Xiaojiang Feng

Abstract. The concept of a landscape or response surface naturally arises in applications widely ranging over the sciences, engineering and other disciplines. A landscape is the desired output as a function of a set of input variables, often of very high dimension. The relationship between the features of a landscape and the input variables is usually unknown *a priori* and often thought to be highly complex due to the anticipated intricate interactions involved. This chapter reviews recent developments in the analysis of landscape topology with the input variables considered as controls. Taking a control perspective allows for the specification of particular assumptions whose satisfaction permits a general analysis of the landscape topology. Satisfaction of these conditions leads to the conclusion that control landscapes should be devoid of suboptimal critical point traps, thereby permitting ready excursions without hindrance to the highest values of the landscape. These principles are set out in a general framework and then specifically illustrated for applications

Herschel Rabitz · Tak–San Ho
Department of Chemistry, Princeton University,
Washington Road, Princeton, NJ 08544, U.S.A.
e-mail: {hrabitz,tsho}@princeton.edu

Re–Bing Wu
Department of Automation, Tsinghua University and Center for Quantum Information Science and Technology, TNLIST, Beijing, 100084, China
e-mail: rbwu@tsinghua.edu.cn

Katharine Moore Tibbetts
Department of Chemistry, Temple University, 1901 N. 13th St.,
Philadelphia, PA 19122, U.S.A.
e-mail: katharine.moore@temple.edu

Xiaojiang Feng
Kadmon Corporation, 450 East 29th Street, New York, NY 10016, U.S.A.
e-mail: fengxiaojiang@yahoo.com

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_2, © Springer-Verlag Berlin Heidelberg 2014

involving control in quantum mechanics, chemical and material science, and in natural and directed evolution. Perspectives are given on the significance of these findings and potential future directions for additional analysis of landscape principles.

2.1 Introduction

Virtually any quantitative application in the sciences and engineering may be expressed in terms of a specified set of input variables and an associated observable system response in the laboratory or in a natural setting. A thorough understanding of the landscape defining such input→output relationships is of fundamental as well as practical significance, especially when the goal is to find an optimal output. Whether optimization or just a basic understanding of the landscape features is the ultimate goal, it is natural to consider the landscape in a control framework. In this fashion, a landscape can be viewed as a function of the control variables with a prime consideration being identification of the landscape topology. Various application domains employ distinct terminology when treating input→output relationships. This chapter will consistently refer to the output as a landscape, which is a map from the input control variables to the output. Following standard practice in the literature this chapter will often refer to control landscapes, which should be understood to mean control-dependent landscapes.

There is a vast literature on landscapes in science and engineering largely developed independently, likely with consideration that *per se*, the nature of the landscape topology should be application specific. This Chapter considers recent research taking a generic perspective on landscape topology with the aim of establishing whether a broad foundation may exist transcending any particular application. Importantly, this research demonstrates that a general assessment may be carried out for control landscape topology upon satisfaction of three basic physical and mathematical assumptions set out in Section 2.2. We refer to this overall analysis as **Opti**mal control in the **Sci**ences (OptiSci). The subsequent sections cover the specific aspects of these general principles realized in the control of **Q**uantum phenomena (OptiQ in Section 2.3), **Chem**ical and material science (OptiChem in Section 2.4) and in natural and directed **Evo**lution (OptiEvo in Section 2.5). Many mathematical challenges remain for the analysis of control landscapes, but the basic principles have been set out. Extensive experimental and simulation evidence for OptiSci supports the broad validity of the underlying assumptions and their conclusions regarding landscape topology, and a summary of the supporting evidence for OptiSci is given in the chapter. Sections 2.2 to 2.5 use terminology and notation particular to control theory and each type of application. The reader is referred to the cited literature for further mathematical and scientific details. The specific application domains treated in this paper (except for natural evolution) are considered in the context of laboratory optimization. Natural evolution is an optimization process as well, but carried out stochastically by Nature. The common landscape topological character found in all domains of OptiSci in Sections 2.3 to 2.5 has fundamental and practical significance. Finally, concluding remarks are given in Section 2.6.

2.2 Basic Foundations of OptiSci

Many aspects of science and engineering can be expressed in terms of cause and effect relationships. The field of control is motivated by the desire to exploit such relationships and alter a system's outcome. This work will address three classes of control processes whose general characteristics are listed in Table 2.1:

Table 2.1 Scope of OptiSci

Domain	Control	Objective Outcome
quantum mechanics (OptiQ)	electromagnetic field	state transformations
chemistry & materials (OptiChem)	reagents, processing conditions	synthesis yield, properties
natural & directed evolution (OptiEvo)	genes, genotypes	fitness

A control landscape is the mapping from the cause (control variables) to the effect (objective outcome). When the control and the outcome can be quantitatively described, the landscape is specified as a function of the control variables. The control landscape can correspond to a natural process (e.g., the fitness landscape in natural evolution) or a scientific venture willfully executed (e.g., a chemical reaction controlled by shaped laser pulses). Nature utilizes a stochastic search on evolutionary landscapes seeking the highest fitness for a species population, while in the laboratory any of a variety of algorithms may be utilized to explore landscapes for the best control outcome. In all cases the features of the control landscape are of fundamental importance to understand the complexity of seeking optimization. In the following, we will introduce the optimization over landscapes in classical and quantum mechanical frameworks. The control landscape will be defined in the so-called kinematic and dynamic pictures based on three assumptions, which enable a complete identification of the topological features of the landscapes.

2.2.1 Classical and Quantum Control Landscapes

A large class of physical, chemical and biological phenomena may be expressed as dynamical control systems. For example, a classical particle is described by its momentum p and position q, which may be controlled by an external field or force. More generally, from a statistical perspective, the state of a large ensemble of identical classical systems or a single classical system driven by stochastic noises can be described by a (nonnegative) probability distribution function ρ^{cl} over the phase space of $\omega = (p,q)$. The positive distribution function ρ^{cl}, which is governed by a dynamical equation (e.g., a Fokker-Planck equation [26]), can be taken as a function of the control $\mathbf{c}$ (in some properly defined space $\mathscr{C}$). The average value of an observable $O(\omega)$ at time T is then given by

$$J^{\mathrm{cl}}(\mathbf{c}) = \int_{\Omega} O(\omega)\rho_{\mathrm{T}}^{\mathrm{cl}}(\omega,\mathbf{c})\,\mathrm{d}\omega, \tag{2.1}$$

forming the classical control landscape. The objective is to find a suitable optimal control $\mathbf{c}^*$ that maximizes $J^{\mathrm{cl}}(\mathbf{c})$.

The quantum mechanical landscape description parallels that of a classical ensemble. Suppose that the quantum system possesses N discrete levels, corresponding to an N-dimensional Hilbert space $\mathscr{H} = \mathbb{C}^N$. The state of an N-level quantum system is represented by a positive, unit trace $N \times N$ density matrix [14], which belongs to the following set

$$\mathscr{D}_N := \{\rho^{\mathrm{qm}} \in \mathbb{C}^{N\times N},\ \ \rho^{\mathrm{qm}} \geq 0, \mathrm{Tr}(\rho^{\mathrm{qm}}) = 1\}. \tag{2.2}$$

Driven by some control $\mathbf{c}$ (e.g., a tailored laser field [25]) under the laws of quantum mechanics (see Equations (2.7) and (2.8) in Section 2.3), the density matrix evolves from some initial state ρ_0^{qm} to the final state $\rho_T^{\mathrm{qm}}(\mathbf{c})$.

Physical observables in a quantum system are represented by Hermitian operators O on the system's Hilbert space $\mathscr{H}$. A wide range of control problems for quantum systems can be formulated as maximization of the expectation value of a suitable target operator O. The goal is to find an optimal control $\mathbf{c}^*$ which maximizes the expectation value $\langle O\rangle_T = \mathrm{Tr}[\rho_T^{\mathrm{qm}} O]$ at the final time T. The corresponding quantum mechanical objective landscape function J^{qm} is

$$J^{\mathrm{qm}}(\mathbf{c}) = \mathrm{Tr}[\rho_T^{\mathrm{qm}}(\mathbf{c})O]. \tag{2.3}$$

A common feature shared by the classical and quantum optimal control formulations is that the function J depends on the controls through the final state $\rho_T(\mathbf{c})$, either as $\rho_T^{\mathrm{cl}}(\mathbf{c})$ or $\rho_T^{\mathrm{qm}}(\mathbf{c})$. This shared character forms the basis for the following landscape analysis.

2.2.2 General Features of Control Landscapes

The set of admissible controls form a function space $\mathscr{C}$, which, under more general circumstances, can be any topological space (e.g., a Boolean space or a differential manifold). The control landscape is formally defined as the mapping from the control variable $\mathbf{c} \in \mathscr{C}$ to a (scalar) outcome $J(\mathbf{c})$, which is a function(al) of $\mathbf{c}$ (see Figure 2.1). There are circumstances where competing multiple objectives arise, but they will not be treated here. Thus, we consider optimal controls as maximizing the scalar quantity J.

To analyze the landscape properties, we may Taylor expand J in the neighborhood of any $\mathbf{c} \in \mathscr{C}$:

$$J(\mathbf{c}+\delta\mathbf{c}) = J(\mathbf{c}) + \langle \nabla J(\mathbf{c}), \delta\mathbf{c}\rangle_{\mathbf{c}} + \frac{1}{2}\langle \delta\mathbf{c}, H_J(\mathbf{c})\delta\mathbf{c}\rangle_{\mathbf{c}} + O(\|\delta\mathbf{c}\|^3), \tag{2.4}$$

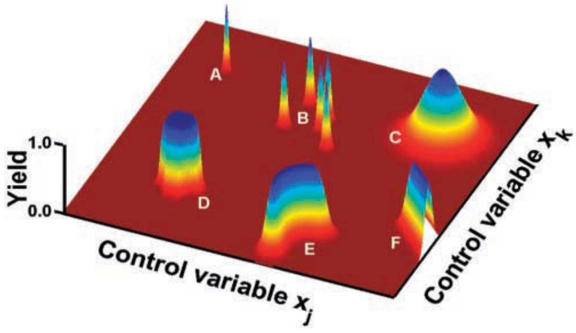


Fig. 2.1 A schematic illustration of a control landscape with two control variables denoted as x_j and x_k. The landscape can be viewed as the (two-dimensional in this case) curved surface in the (three-dimensional) product space of the yield and the space of control parameters. The surface forms the graph defined in Equations (2.1) and (2.3). The landscape's yield J runs from 0.0 to the maximum value of 1.0. The various classes of extrema A-F reflect differing ease of discovery and robustness properties. Adapted from [46].

where $\delta\mathbf{c}$ is a small perturbation and $\langle\cdot,\cdot\rangle_{\mathbf{c}}$ refers to a suitable inner product in the tangent space of $\mathscr{C}$ at $\mathbf{c}$. A critical point at $\mathbf{c} \in \mathscr{C}$ is where the gradient $\nabla J(\mathbf{c}) = \frac{\mathrm{d}J}{\mathrm{d}\mathbf{c}}$ vanishes.

The term quadratic in $\delta\mathbf{c}$ in Equation (2.4) involves the Hessian operator $H_J(\mathbf{c})$. Most critical points can be identified as either locally maximal (when all of the Hessian eigenvalues are negative), locally minimal (when all of the Hessian eigenvalues are positive) or locally a saddle (when positive and negative Hessian eigenvalues co-exist). For the critical points where there are either only positive, or negative, or zero eigenvalues (i.e., a semi-definite Hessian), higher-order Taylor expansion terms may be needed for identifying the type of the critical point) [18].

The optimization procedure can be envisaged as hiking over the landscape (see Figure 2.1) towards the highest peak, which corresponds to a global maximum point. An important goal of control landscape analysis is to assess whether local suboptimal maximum points exist, which could trap a search seeking the best outcome. Saddle points will not trap the search, but they may hinder the effort to find an optimal solution. The existence of traps and saddle points can greatly affect the algorithmic efficiency of finding an optimal solution from a neighboring initial point on the landscape. Thus, a major goal of landscape analysis is to address the following questions:

- How many locally maximal points exist?
- How many globally maximal points exist?
- How can knowledge of the landscape topology aid in the design of efficient optimization algorithms?

A full answer to all of these questions is not at hand, but much can be said now about these issues under certain assumptions discussed below.

2.2.3 Landscape Mapping

In physical, chemical and biological processes, the landscapes are in general difficult to analyze due to their dimensions and high nonlinearities. One can facilitate the analysis by mapping the landscape to another space $\mathscr{X}$ via $\phi : \mathscr{C} \to \mathscr{X}$. Denote $\tilde{J} = J \circ \phi$ as a composition such that $\tilde{J}(\mathbf{c}) = J[\phi(\mathbf{c})]$ for any $\mathbf{c} \in \mathscr{C}$. Going from $\tilde{J}(\mathbf{c})$ to $J(\phi)$ is a landscape mapping induced by $\phi : \mathscr{C} \to \mathscr{X}$.

For example, ϕ is referred to as the end-point map for dynamical systems discussed in OptiQ. Here $\tilde{J}(\mathbf{c})$ is called the *dynamic control landscape* which depends on the system evolution, while $J(\phi)$ is the *kinematic control landscape* and only depends on the final state. Determining the landscape topology can be difficult in the dynamic picture, but it is frequently much simpler in the kinematic picture because J is often linear in $\mathscr{X}$ and lower-dimensional. The corresponding critical points are referred to as dynamic or kinematic.

Here we seek to extract the landscape topology of a complex dynamic control landscape from the corresponding kinematic landscape. This analysis can be performed through the linkage of the critical points in the two landscapes by the chain rule:

$$\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\mathbf{c}} = \left\langle \frac{\mathrm{d}J}{\mathrm{d}\mathbf{x}}|_{\mathbf{x}=\phi(\mathbf{c})}, \frac{\mathrm{d}\phi}{\mathrm{d}\mathbf{c}} \right\rangle = 0, \tag{2.5}$$

where $\mathrm{d}\phi/\mathrm{d}\mathbf{c}$ is the Jacobian of the mapping ϕ at $\mathbf{c}$. If $\mathbf{x}_0 = \phi(\mathbf{c}_0)$ is a critical point of J, i.e., $\frac{\mathrm{d}J}{\mathrm{d}\mathbf{x}} = 0$ at $\mathbf{x}_0$, then the corresponding $\mathbf{c}_0 \in \mathscr{C}$ must also be a critical point of $\tilde{J}$ because Equation (2.5) implies that $\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\mathbf{c}} = 0$ at $\mathbf{c}_0$. But, the reverse is not necessarily true unless the Jacobian $\frac{\mathrm{d}\phi}{\mathrm{d}\mathbf{c}}$ at $\mathbf{c}_0$ is non-singular, i.e., the rank is exactly the dimension of $\mathscr{X}$. Elements in $\mathscr{C}$ that make the Jacobian rank-deficient are called singular, otherwise they are referred to as regular [69].

Circumstances where regularity applies guarantees consistency between the critical conditions in the two mapped landscapes. This conclusion can be understood from the following relationship

$$H_{\tilde{J}}(\mathbf{c}) = \left\langle \frac{\mathrm{d}\phi}{\mathrm{d}\mathbf{c}}, H_J(\phi)\frac{\mathrm{d}\phi}{\mathrm{d}\mathbf{c}} \right\rangle \tag{2.6}$$

between the Hessian forms in the two pictures, where the right hand side can be taken as a congruent transformation of $H_J(\phi)$ when both $\mathscr{X}$ and $\mathscr{C}$ are finite dimensional. It can be shown from linear algebra that the eigenvalues of $H_{\tilde{J}}(\mathbf{c})$ and $H_J(\phi)$ have identical signs (not necessarily their magnitudes) when the Jacobian $\frac{\mathrm{d}\phi}{\mathrm{d}\mathbf{c}}$ is full rank. This implies that the numbers of positive and negative eigenvalues are preserved when the landscape mapping is regular at a critical point, which determines the nature of optimality to second-order. Thus, $\mathbf{x}_0 = \phi(\mathbf{c}_0)$ is a second-order maximum (minimum or saddle) of J if and only if $\mathbf{c}_0$ is a second-order maximum (minimum or saddle) of $\tilde{J}$. The number of zero Hessian eigenvalues is usually different in the two spaces, reflecting the different "size" of the critical set in each space.

2.2.4 Requirements for the Existence of Trap-Free Landscapes

In the following application oriented sections of the paper, the mapping described above will be used to simplify the landscape analyses. For example, in OptiQ, the mapping from the space of control fields to the state space will be employed.

A landscape mapping from $\tilde{J}(\mathbf{c})$ to $J(\phi)$ preserves the landscape topology if and only if the following three assumptions are satisfied:

Assumption (1): The system is controllable such that for every point $\mathbf{x} \in \mathscr{X}$, there must exist some $\mathbf{c} \in \mathscr{C}$ such that $\phi(\mathbf{c}) = \mathbf{x}$. In a dynamical system, this condition guarantees that every final state is reachable at some sufficiently long time.

Assumption (2): Every critical point in $\phi(\mathscr{C})$ that is mapped from $\mathscr{C}$ to $\mathscr{X}$ is regular, i.e., the Jacobian is full rank. This surjectivity condition implies that locally the search for optimal controls can be conducted along any direction in $\mathscr{X}$ after being mapped from $\mathscr{C}$. This condition guarantees that every dynamic critical point must correspond to a kinematic critical point with the same optimality type.

Assumption (3): No restrictions are placed on the control. This (strong) demand assures full access to reachable points on the landscape.

The detailed assessment of the Assumptions (1), (2) and (3) above and landscape analyses differs across OptiQ, OptiChem and OptiEvo. The assumptions may be either difficult or easy to satisfy in practice, depending on the application. In this regard, the most stringent is Assumption (3), and in practice all of the assumptions may be relaxed while still finding a favorable landscape topology consistent with the formal analysis. The practical applications clearly demonstrate this point. Under satisfaction of the strict assumptions above, we will prove in various ways within the following sections that, despite the many differences in the applications, all OptiSci landscapes are trap free, i.e., there are no local suboptima trapping extrema. This conclusion provides strong support for interpreting the large number of observed landscapes and successful optimization experiments in quantum control, chemistry and biological evolution.

2.3 Optimal Control of Quantum Dynamics (OptiQ)

2.3.1 Background

Employing electromagnetic fields to control quantum dynamics is an active area of research with many potential applications in physics, chemistry, information science and biology [7]. Advances in ultrafast laser technology along with flexible pulse shaping capabilities [62] has enabled successful control of broad varieties of quantum phenomena. Regardless of the particular quantum control application, a main goal of **Opti**mal control of **Q**uantum dynamics (OptiQ) is to achieve the best possible results, while also working with inevitable laboratory constraints. This perspective has led to performing **O**ptimal **C**ontrol **E**xperiments (OCE) as the means to attain the best outcomes [8].

The underlying principle for realizing OCE grew out of extensive groundwork from quantum **O**ptimal **C**ontrol **T**heory (OCT) calculations on model systems [30, 43, 56], which continues to serve as an indispensable guide to controlling physical phenomena. Any successful control in the laboratory must be able to handle many uncertainties, including insufficient knowledge about typical complex system Hamiltonians and the practical difficulty of generally performing high quality numerical designs for the controls. The implementation of closed-loop learning control [25] is a practical and effective scheme to carry out OCE under these circumstances. Such experiments started in 1997 [3, 4] and their number has steadily grown with various applications [7]. Recent OCT and OCE developments indicate that there may be a systematic "chemistry" associated with shaped laser pulses acting as "photonic reagents". Despite having a fleeting existence, photonic reagents can permanently alter quantum states and transform molecules and condensed phase materials.

Quantum optimal control seeks to find a field $\varepsilon(t)$, $t \in [0,T]$ to steer a particular quantum system from an initial state $|\psi(0)\rangle = |\psi_i\rangle$ (or $\rho(0)$ in the density matrix formulation) to some target state $|\psi(T)\rangle$ (or its density matrix analog $\rho(T)$), which maximizes the expectation value $\langle\psi(T)|O|\psi(T)\rangle$ (or $\mathrm{Tr}\{\rho(T)O\}$) of the chosen target observable O. The common assumption is that evolution with the free Hamiltonian H_0 alone can not satisfactorily reach the objective. Thus, the photonic reagent $\varepsilon(t)$ is needed to redirect the dynamics described by the time-dependent Schrödinger equation, which often has the form

$$\imath\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = [H_0 - \mu\varepsilon(t)]\,|\psi(t)\rangle,\ |\psi(0)\rangle = |\psi_i\rangle, \tag{2.7}$$

where $\imath \equiv \sqrt{-1}$ and μ is the dipole moment operator.

Many OCEs involve highly complex systems with rich Hamiltonian structure that is not known in quantitative detail. For example, the multidimensional potential energy surface $V(\mathbf{R},\mathbf{r})$ residing in the Hamiltonian term H_0 can depend on both the nuclear coordinates $\mathbf{R}$ and electronic coordinates $\mathbf{r}$ when the applied field simultaneously controls multiple atoms or electrons. The adaptive feedback OCEs automatically take into consideration all of the relevant potential energy surface details (and limits on the available lasers) for determining an optimal field. However, these considerations alone do not explain the observed ease in finding optimal controls over quantum phenomena.

2.3.2 *Basic Assumptions Regarding Quantum Control Landscapes*

The success in readily finding optimal controls of various quantum dynamical processes can be understood by consideration of the quantum control landscape [21, 22, 44, 46], which is the expectation value $J[\varepsilon(\cdot)] \equiv \langle\psi(T)|O|\psi(T)\rangle$ of the observable O as a function of the control field $\varepsilon(t)$. In the more general case of the density matrix, $J[\varepsilon(\cdot)] = \mathrm{Tr}\{\rho(T)O\}$. Extensive research has been carried out to understand the properties of quantum control landscapes for the manipulation of arbitrary N-level quantum systems. Although the energy spectrum of a typical molecule is composed of both discrete bound rotational-vibrational-electronic energy levels and continuous ionization/dissociative channels, in practice, it may be modeled in a finite-dimensional Hilbert space. Thus, we assume that the molecule under analysis can be adequately represented by a finite number of N energy levels, for example, via proper discretization of continuum states. Despite the fact that each molecule is uniquely characterized by its Hamiltonian H_0 and dipole moment μ, theoretical analysis shows that quantum control landscapes should have a universal quality that is only determined by the initial state $|\psi(0)\rangle$ of the molecule and the nature of the observable operator O. This result suggests that a broad foundation must be behind the ease of finding effective quantum controls, regardless of the complexity of the quantum system. The above conclusion rests on satisfaction of the three basic Assumptions of OptiSci in Section 2.2, expressed here in the context of OptiQ:

(1) **The quantum system is controllable [23, 47, 52, 68]:** The quantum system described by Equation (2.7) evolves on the $N \times N$ unitary matrix group $\mathrm{U}(N)$ and is controllable if it can be driven from any initial state $|\psi_i\rangle$ to any desired final state $|\psi_f\rangle$ in some sufficiently long time. This is possible if the Lie algebra generated by the field-free Hamiltonian H_0 and the dipole moment μ via their nested commutators coincide with the Lie algebra associated with the unitary matrix group $\mathrm{U}(N)$.

(2) **The control end-point map is surjective [69]:** The functional derivative of the underlying control end-point map $\varepsilon(\cdot) \mapsto |\psi(T)\rangle$ (or $\rho(T)$), given the initial state $|\psi(0)\rangle$ (or $\rho(0)$), is of full rank. The resultant end-point map is surjective and the corresponding control $\varepsilon(t)$ is regular. Surjectivity assures free movement around any local end-point on the landscape.

(3) **The control field is not constrained [39, 45]:** The control field $\varepsilon(t)$ is allowed to have an arbitrary form to eliminate artificially introduced constraints on moving over the landscape.

The satisfaction of Assumption (1) for pure state-to-state N-level optimal control problems implies that some control exists to reach the target state $|\psi_f\rangle$ at a suitable time T from $|\psi(0)\rangle = |\psi_i\rangle$ at $t = 0$. Although exceptions to fulfillment of this assumption can be found, mathematical and physical analysis suggests that most finite dimensional quantum systems are likely to be controllable.

Assumption (2) requires that the rank of the functional derivative $\delta\rho(T)/\delta\varepsilon(t)$ be full. For pure state transition probability control problems, this assumption requires that $\delta\rho(T)/\delta\varepsilon(t)$, with $\rho(T) \equiv |\psi(T)\rangle\langle\psi(T)|$, possess a rank of $2N-2$. Assumption (2) is equivalent to stating that a differential change in the state $\delta\rho(T)$ at the final time has a corresponding differential control $\delta\varepsilon(t)$ producing it. Although this assumption could be violated by so-called singular controls, their occurrence seems to be rare.

Assumption (3) assures access to all electromagnetic resources. In practice, however, the controls only need to be sufficiently unconstrained in order to address all the necessary transitions to create the desired final state.

Collectively, satisfaction of the Assumptions (1), (2) and (3) provides the conditions for generically assessing the topological nature of quantum control landscapes. The conclusion of this analysis is that suboptimal local maxima forming traps should not exist on quantum control landscapes. A trap is a local submaximal value of the objective $J[\varepsilon(\cdot)]$, from which it is not possible to climb further on the landscape to a higher value through small variations of $\varepsilon(t)$ guided by a myopic algorithm (e.g., a gradient based procedure). Rigorously establishing that the three assumptions above are satisfied in any particular case is difficult. But, it is plausible to expect that these assumptions should be fulfilled, likely including Assumption (3) to a practical degree, in many applications, especially as control resources become richer. Accepting satisfaction of the assumptions, the resultant conclusion about the trap free nature of the quantum control landscape topology can be tested in simulations and in the laboratory.

2.3.3 Quantum Control Landscape Theory

Consider a generic observable (Hermitian operator) O, with the corresponding N-level quantum dynamical control problem posed as follows: Find a control field $\varepsilon(t)$ over period T such that the initial density matrix $\rho(0)$, $\mathrm{Tr}\{\rho(0)\} = 1$, can be transformed by the corresponding propagator $U(T,0)$ to form $\rho(T) = U(T,0)\rho(0)U^\dagger(T,0)$ which maximizes the expectation value $\langle O(T)\rangle = \mathrm{Tr}\{\rho(T)O\}$. Here the propagator $U(t,0)$ satisfies the equation

$$\imath\hbar\frac{\partial}{\partial t}U(t,0) = [H_0 - \mu\varepsilon(t)]\,U(t,0),\; U(0,0) = \mathbf{I}\text{ (identity operator).} \tag{2.8}$$

The quantum control landscape [21, 22, 44, 46] is defined as a functional of the control field $\varepsilon(t)$ over $t \in [0,T]$:

$$J[\varepsilon(\cdot)] \equiv \langle O(T)\rangle. \tag{2.9}$$

The essential features of $J[\varepsilon(\cdot)]$ are its critical points where the first-order functional derivative (gradient) satisfies

$$\frac{\delta\langle O(T)\rangle}{\delta\varepsilon(t)} = 0,\; \forall t \in [0,T]. \tag{2.10}$$

The characterization of a critical point with the field $\varepsilon(t)$ is revealed by analyzing the spectrum of the landscape curvature (Hessian), $\delta^2\langle O(T)\rangle/\delta\varepsilon(t')\delta\varepsilon(t)$. The critical points satisfying Equation (2.10) may correspond to global extrema (maxima or minima) or local (suboptimal) values of $J[\varepsilon(\cdot)]$, which may be either traps or saddles. Assessment of the latter prospects is of prime interest in OptiQ. The gradient $\delta\langle O(T)\rangle/\delta\varepsilon(t)$ can be written as

$$\frac{\delta\langle O(T)\rangle}{\delta\varepsilon(t)} = \frac{\imath}{\hbar}\mathrm{Tr}\{[\rho(0), O(T)]\,\mu(t)\}, \tag{2.11}$$

where $O(T) \equiv U^\dagger(T,0)OU(T,0)$ and $\mu(t) \equiv U^\dagger(t,0)\mu U(t,0)$, and the Hessian of the cost functional $\langle O(T)\rangle$ is

$$\frac{\delta^2\langle O(T)\rangle}{\delta\varepsilon(t')\delta\varepsilon(t)} = -\frac{1}{\hbar^2}\mathrm{Tr}\left\{\left[\rho(0), \left[O(T), \mu(t')\right]\right]\mu(t)\right\},\ t' \geq t. \tag{2.12}$$

Here we have introduced the commutator definition $[A,B] \equiv AB - BA$ for any two operators A and B.

Considering that all three assumptions above are satisfied, then $\mu(t)$ in Equation (2.11) should be a matrix whose elements are linearly independent function of time. Thus, the dynamic critical point criterion $\delta\langle O(T)\rangle/\delta\varepsilon(t) = 0\ \forall t \in [0,T]$ reduces to the kinematic one $[\rho(0), O(T)] = 0$ [22, 69] for a specific unitary transformation $U(T,0)$. Given the initial density matrix $\rho(0)$ for an N-level quantum system, a control is called regular when the rank $R^N_{\rho(0)}$ of the functional derivative

$$\frac{\delta\rho(T)}{\delta\varepsilon(t)} = \frac{\imath}{\hbar}U(T,0)\{[\mu(t), \rho(0)]\}U^\dagger(T,0) \tag{2.13}$$

of the end-point map (defined in the neighborhood of the control field $\varepsilon(\cdot)$)

$$E^{\rho(0),T} :\ \varepsilon(\cdot) \mapsto \rho(T),\ \varepsilon(\cdot) \in \mathscr{L}^2[0,T] \tag{2.14}$$

is of full rank, otherwise, it is called singular [69]. For pure state-to-state regular controls, given $\rho(0) = |i\rangle\langle i|$ as the initial density matrix, $\delta\rho(T)/\delta\varepsilon(t)$ is of rank $2N-2$, corresponding to the case that the real and imaginary parts of the $(N-1)$ dipole matrix elements $\langle i|\mu(t)|k\rangle$, $1 \leq k \neq i \leq N$, form $2N-2$ independent real functions of time $t \in [0,T]$.

Consider the common case that the initial density matrix describes a mixed state of the form

$$\rho(0) = \sum_{i=1}^{N} \rho_i|i\rangle\langle i|, \quad \sum_{i=1}^{N}\rho_i = 1, \quad \rho_1 \geq \rho_2 \geq \cdots \geq \rho_N \geq 0, \tag{2.15}$$

where $|i\rangle$'s are eigenstates of H_0. The rank $R^N_{\rho(0)}$ of $\delta\rho(T)/\delta\varepsilon(t)$ is at most equal to the dimension $D^N_{\rho(0)}$ of the corresponding reachable set $\mathscr{O}(\rho(0))$, given the initial density matrix $\rho(0)$, i.e. $R^N_{\rho(0)} \leq D^N_{\rho(0)}$ [69, 1]. For example, $R^N_{\rho(0)}$ is at most $2N-2$ when $\rho(0)$ is a pure state (e.g., $\rho_1 \neq 0$ and $\rho_2 = \cdots = \rho_N = 0$) and at most $N^2 - N$ when $\rho(0)$ is fully non-degenerate (i.e., $\rho_i \neq \rho_j \neq 0 \forall i \neq j$). Note that $D^N_{\rho(0)} = 2N-2$ in the former case and $D^N_{\rho(0)} = N^2 - N$ in the latter case, respectively.

At kinematic critical points, since $\rho(0)$ and $O(T)$ commute, we arrive at $O(T) = \sum_i \lambda_i(T)|i\rangle\langle i|$, where $\lambda_1(0) \geq \cdots \geq \lambda_N(0) \geq 0$ are eigenvalues of the initial observable O, arranged in a descending order (assuming that O is positive semi-definite without loss of generality). The eigenvalues $\lambda_1(T), \cdots, \lambda_N(T)$ of $O(T)$ at the kinematic critical point are particular permutations of the eigenvalues $\lambda_1(0), \cdots, \lambda_N(0)$ prescribed by the evolution of the system under the control field $\varepsilon(t)$. In this fashion the propagator $U(T,0)$ acts as a permutation matrix that rearranges the ordering of the eigenvalues of O at time T. Specifically, we find that (1) $U(T,0)$ is an $N \times N$ identity matrix at the global maximum, (2) $U(T,0)$ is an anti-diagonal unit matrix that renders the permutations $i \to N-i+1$ for all $i = 1,2,\cdots,N$ at the global minimum, and (3) $U(T,0)$ is a particular permutation matrix at each local critical point. There are at most $N!$ allowed critical-point values of $\langle O(T)\rangle$ specified by [22]

$$\langle O(T)\rangle = \sum_{i=1}^{N} \rho_i \lambda_i(T), \tag{2.16}$$

where $\rho_1 \geq \cdots \geq \rho_N \geq 0$ for the eigenvalues of $\rho(0)$, and $\{\lambda_i(T)\}$ is appropriately permuted at each critical point. From Equation (2.16), the expectation value $\langle O(T)\rangle$ is bounded by the relation [22, 59]

$$\langle O(T)\rangle_{\min} = \sum_{i=1}^{N} \rho_i \lambda_{N-i+1}(0) \leq \langle O(T)\rangle \leq \langle O(T)\rangle_{\max} = \sum_{i=1}^{N} \rho_i \lambda_i(0). \tag{2.17}$$

The topology of the control landscape at the kinematic critical points may be analyzed via the corresponding Hessian, which is inherently separable (degenerate) and possesses a finite number of non-zero eigenvalues, i.e., at most either $2N-2$ or N^2-N, depending on the dimension of the reachable set $\mathscr{O}(\rho(0))$ associated with the initial density matrix $\rho(0)$. The Hessian with respect to the dynamic control at the kinematic critical points is [22]

$$\begin{aligned}\frac{\delta^2\langle O(T)\rangle}{\delta\varepsilon(t')\delta\varepsilon(t)} &= -\frac{2}{\hbar^2}\sum_{i=1}^{N}\sum_{j>i}^{N}(\rho_i-\rho_j)\left(\lambda_i(T)-\lambda_j(T)\right) \\ &\quad \times \left\{\Re\langle i|\mu(t)|j\rangle\Re\langle i|\mu(t')|j\rangle + \Im\langle i|\mu(t)|j\rangle\Im\langle i|\mu(t')|j\rangle\right\},\end{aligned} \tag{2.18}$$

where $\Re$ and $\Im$, respectively, denote the real and imaginary parts of a complex number. Consequently, we have $\lambda_i(T) - \lambda_j(T) = \lambda_i(0) - \lambda_j(0) \geq 0\ \forall i < j$ at the global maximum and $\lambda_i(T) - \lambda_j(T) = \lambda_{N-i+1}(0) - \lambda_{N-j+1}(0) \leq 0\ \forall i < j$ at the global minimum. Moreover, since the set $\left\{\lambda_i(T) - \lambda_j(T) \mid 1 \leq i < j \leq N\right\}$ consists of both positive and negative numbers, all local suboptimal critical points correspond to saddles (and are trap-free).

2.3.4 Simulated Excursions over Quantum Control Landscapes

Carefully performed numerical studies of controlled quantum dynamics can be valuable for assessing the validity of the landscape analyses based on the Assumptions (1), (2) and (3) discussed above. Recently extensive numerical simulations have been performed to study this matter, especially, for the control of the state-to-state transition probability $P_{i\to f}(T) \equiv |\langle f|U(T,0)|i\rangle|^2$ [39]. The simulation models considered from 2 to approximately 100 states. Remarkably, controlling complex quantum systems appears to be no more difficult than controlling simple ones, where the level of difficulty is expressed in terms of the number of iterations required to converge on the target objective. Using a gradient-based algorithm [51] these detailed simulations involved approximately 5000 cases and all tests reached perfection for the objective to at least three decimal places upon paying due attention to numerical details. Similar results were observed for optimization of the control objective of generating a transformation $U(T,0)$ that matches a target unitary matrix W [35]. This objective may be measured by considering the fidelity function $J = ||W - U(T,0)||^2$. In the latter study, approximately 20000 tests were performed on quantum systems with 2 to 16 states, and each optimization converged to a fidelity value of $J \leq 10^{-6}$.

Although more studies are still necessary, these results are consistent with the three basic assumptions leading to the conclusion that quantum control landscapes are generally expected to have a trap-free landscape topology. A key limiting factor is the requirement of Assumption (3) for free access to all essential control resources. While not an issue in computer simulations, there will always be a need for further control resources in the laboratory, especially when treating complex systems. In some cases more bandwidth and energy may suffice to meet the control needs of Assumption (3). But, even subtle resource limitations can become important in some circumstances [40].

2.3.5 Experimental Excursions over Quantum Control Landscapes

While thousands of carefully performed numerical simulations have successfully reached the highest landscape values with ease, direct exploration of control landscapes in the laboratory has only just begun. To this end, it is important to consider that in the laboratory many factors (e.g., noise, inhomogeneities, and, especially, constraints on the controls) enter beyond those that arise in simulations. Such constraints can have important consequences by possibly imposing artificial boundaries that limit movement over the control landscape and lead to a reduction of the attainable control yield. For example, Figure 2.2 shows the laboratory data for the quantum control landscape of a filtered **S**econd **H**armonic **G**eneration (SHG) signal [50]. The results show that the landscape is free of traps provided that constraints are not placed on the control variables. But, the figure also shows that constrained paths (i.e., a violation of Assumption (3)) over the landscape can produce artificial traps.

A particularly important finding from the landscape analysis concerns the rank of the Hessian $\delta^2 P_{i\to f}(T)/\delta\varepsilon(t')\delta\varepsilon(t)$ at the top and bottom of the transition probability landscape $J[\varepsilon(\cdot)] = P_{i\to f}(T)$. The analysis [22, 44, 46] shows that there exist at most $2N-2$ routes off the top of the landscape (i.e., $P_{i\to f}(T) = 1.0$) and at most two routes up from the bottom (i.e., $P_{i\to f}(T) = 0.0$) for a quantum system with N states.

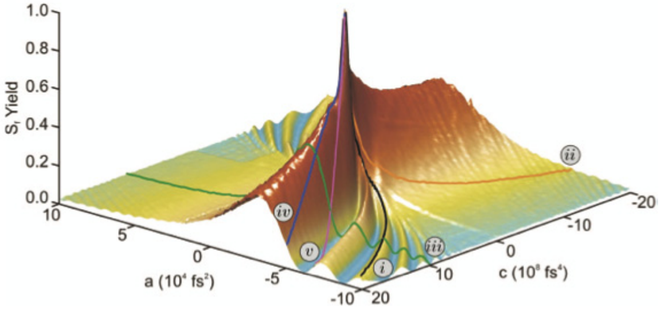


Fig. 2.2 Experimental trap-free quantum control landscape for filtered second harmonic generation signal $S(a,c)$ as a function of quadratic and quartic chirp control field coefficients, a and c. The key landscape topological features are preserved by this form of the control field. Color is used here to accentuate the landscape features, and it maps onto yield in a one-to-one fashion. Shown are five trajectories $i, \cdots, v$ placed onto the experimental landscape after acquiring the data. The control landscape possesses a single global optimum that may be reached monotonically by following the curvilinear channels that slice through the landscape. Two such curvilinear channels are indicated by the black and orange trajectories (Adopted from Figure 4 of [50]).

At both landscape extremes the routes are specified by the associated eigenvectors of the Hessian with non-zero eigenvalues. Importantly, the numbers of non-zero eigenvalues of the Hessian, at the top or bottom of the landscape, does not depend on the quantitative details of the Hamiltonian, besides the need to fix Assumptions (1) and (2) of OptiQ. Although the Hessian is infinite dimensional (i.e., time or frequency is continuous), in practice it would be of finite size through digitization of the control field in the laboratory. An experimental test of the Hessian spectral predictions at the top and bottom of the $P_{i \to f}(T)$ landscape was performed in atomic Rb vapor, and the results were found to be fully consistent with the theoretical predictions [49]. Finally, an additional feature of practical importance is the null Hessian spectrum at the top of the landscape, which implies an inherent degree of robustness to noise when controlling quantum phenomena.

2.3.6 Perspective

Understanding the properties of quantum control landscapes is essential for designing successful experiments and to identify promising directions for the future of OptiQ research as well as practical applications. For example, optimal control simulations [39, 40] and experiments [49, 50] on a wide range of model quantum systems have been performed to study the quantum control landscape Hessian spectra for the identification of the energy levels that are dynamically involved in the presence of optimal control fields, thus revealing the corresponding control mechanisms.

The universal trap-free quality of the quantum control landscapes, upon satisfaction of the three basic OptiQ assumptions, provides very favorable circumstances for maximizing physical observables. However, many additional issues remain to be explored. For example, it is important to know how constraints on the controls enter to limit the attainable yields and introduce landscape traps. Such traps will be "artificial" due to their constrained control resource origin, although they will appear to be real in the associated experiment or simulation. Moreover, many experiments achieve optimal control upon comparison of one observable value to another. Thus, further study on the structure of control landscapes is needed for multiple observables, including in the presence of realistic laboratory constraints.

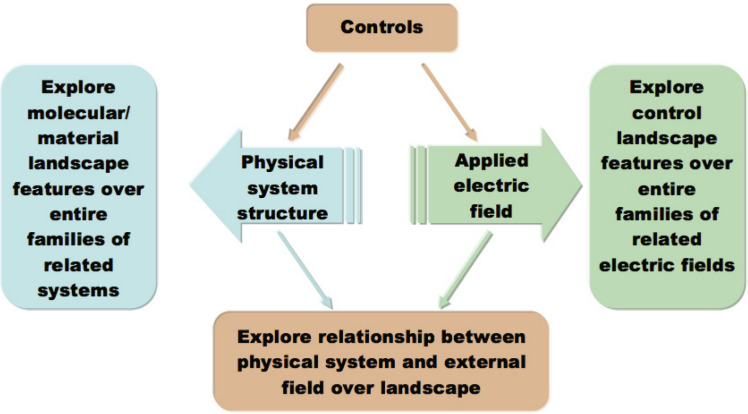

Controls
Explore molecular/ material landscape features over entire families of related systems
Physical system structure
Applied electric field
Explore control landscape features over entire families of related electric fields
Explore relationship between physical system and external field over landscape

Fig. 2.3 The full prospects for OptiQ. The traditional perspective for controlling quantum dynamics phenomena in OptiQ is to start with a molecular or material sample and search through the photonic reagent 'stockroom' to find an optimal control field. An enhanced scenario involves simultaneously searching over the chemical/material and photonic reagent stockrooms to meet demanding objectives, as indicated above. A fully automated machine of this type may be envisioned for this purpose.

The concept of quantum control landscape in OptiQ can be extended to explore the full nature of candidate molecules and other systems through their total Hamiltonian. In this fashion the stationary part of the Hamiltonian as well as the temporal control field can simultaneously be considered as overall control variables with the goal of achieving the highest possible yields. In this most general setting, we may simultaneously search over the chemical or material stockroom along with the photonic reagent 'stockroom' for an effective field $\varepsilon(t)$ (see Figure 2.3). A final possibility is to seek control in the absence of the photonic reagents and only utilize the chemical and material stockroom as well as processing conditions to optimize synthesis yields and properties, which naturally forms the basis of OptiChem discussed in Section 2.4.

A long-standing interest in the field of quantum control is the possible existence of systematic rules for identifying successful controls. In the background, motivating this goal is the fact that chemical transformations carried out with standard reagents generally follow well-defined classes of rules and mechanisms. The prospect that analogous rules may exist for the action of photonic reagents is tantalizing and would surely be related to the favorable landscape topology in OptiQ.

2.4 Optimal Control of Chemical Synthesis and Properties (OptiChem)

2.4.1 Introduction

Common goals in chemistry include optimizing both synthesis yields and molecular/material properties, which can be expressed as optimal control problems by specifying a suitable set of variables. Although chemical synthesis and property discovery are typically not considered in an optimal control framework, recent research shows that employing optimal control concepts can help to explain why it is surprisingly easy to optimize chemical reaction conditions and properties. In this case, “easy” refers to the laboratory effort reflected in the *number* of independent experiments required for the search/optimization procedure to find optimal values of the control variables, ignoring the often time-consuming and expensive overhead involved in setting up experiments. This remark also applies to OptiQ and OptiEvo laboratory efforts. In particular, widespread evidence shows that solving chemical optimization is consistently easier than the pessimistic “curse of dimensionality” [5] would imply. For instance, the application of genetic algorithms (GA) [20] and other optimal search procedures shows that optimization of both synthesis yield and properties typically may be achieved by carefully screening a small number of potential combinations of variables, even for nominally highly complex systems. An extensive review of practical synthesis and property optimizations beating the curse of dimensionality is given in [37, 38]. The origin of these mounting successes is the key topic addressed in this section.

In analogy to OptiQ in Section 2.3, optimization in chemistry is considered as a search over the appropriate control landscape, defined as the measurable objective value J (e.g., the synthesis yield or property value) as a function of the variables. This definition is more general than the commonly encountered concept of an *energy landscape*, where the objective is defined as minimizing the free energy or potential energy of the system (e.g., an atomic cluster or protein). Energy landscapes constitute a specific sub-class drawn from the general family of control landscapes in chemistry because the variables are often highly constrained, for instance, the only significant variable freedom for protein folding is torsional motion. While energy landscapes typically have a rugged topology [58], recent studies [37, 38] demonstrate that control landscapes in chemistry possess a “regular” trap-free topology, upon satisfaction of Assumptions (1), (2) and (3).

The mathematical formulation of optimization in chemistry is that of controlling an *open quantum system*, consisting of the target system and the environment with which it interacts [67]. The topological analysis of the resulting control landscape and its practical implications have recently been termed "OptiChem" theory [37, 38]. The presence of a trap-free control landscape provides an organizing principle for explaining why synthesis and property discovery is far easier than the apparent complexity of such problems would suggest. Importantly, the landscape analysis underlying OptiChem may be carried out with minimal assumptions about the specific nature of the control variables employed. OptiChem theory makes general predictions about the behavior of synthesis and property optimizations that are widely confirmed in laboratory studies [38], and also leads to specific algorithmic suggestions on how to exploit the landscape topology to significantly accelerate laboratory optimization efforts and provide new tools for chemical property prediction.

2.4.2 OptiChem Theory

The following landscape topology analysis is based on a quantum mechanical description of an open system; a proof of landscape topology for a classical description of an open system is presented in Section 2.5. The state of an open quantum system may be represented by a density matrix ρ, whose properties are given in Equation 2.2. The set of all states can be identified with the set of all density matrices $\mathscr{D}$. Analysis of the control landscape topology via OptiChem theory requires the satisfaction of three physical Assumptions, as a specific case of those set out generally in Section 2.2.

(1) **The system is controllable:** The objective must be well-posed, such that the optimal density matrix $\rho^\star \in \mathscr{D}$ that maximizes the objective J is a reachable state of the quantum system.
(2) **The control end-point map is surjective:** A small change in a state ρ must be associated with a corresponding small change in the control, denoted as a vector of variables $\mathbf{c}$.
(3) **No significant constraints are imposed on the control variables:** The set of variables $\mathbf{c}$ must be sufficiently flexible to permit free movement on the landscape.

Assumption (1) states that an ill-posed objective may not be reached by *any* values of the variables, where an obvious example would be a target synthetic molecule that contains an atom with an unattainable valence state. Realistic applications will be more subtle, and the issue of designing a well-posed problem always deserves attention. Satisfaction of Assumption (2) is difficult to *a priori* assess, but nominally more "complex" systems may afford the added freedom needed to satisfy this assumption. Assumption (3) may easily be violated for certain highly constrained objectives, such as optimizing the free energy of a molecule using torsional degrees of freedom as variables [58]. In general, full access to *all* possible types and values of the relevant variables $\mathbf{c}$ will not be available, but the key issue for Assumption (3) is the presence of no *significant* constraints. The general question is whether practical optimization conditions satisfy the assumptions of OptiChem theory and can take advantage of its features. These matters are assessed below in Section 4.4.

OptiChem theory rests on the ability to describe chemical transformations of open systems upon satisfaction of the three assumptions above. The general transformation of an initial state ρ_0 (i.e., before the chemical transformation has occurred) to the state ρ (i.e., the state of the products after the chemical transformation) can be represented by a function Φ such that $\rho = \Phi(\rho_0)$. The transformation described by Φ has the properties of being trace-preserving (i.e., $\mathrm{Tr}\,\Phi(\rho_0) = \mathrm{Tr}\,(\rho) = \mathrm{Tr}\,(\rho_0) = 1$) and completely positive (CP). The trace-preservation property of Φ ensures that the sum over the probabilities of all transformation outcomes is unity. The CP requirement ensures satisfaction of the property $\rho \geq 0$ under transformation of ρ_0 to some other density matrix ρ [17, 31]. Such CP, trace preserving functions Φ are called Kraus maps, described by the matrices $\boldsymbol{K}$. The CP and trace preservation conditions imply certain constraints on the structure of the Kraus matrices $\boldsymbol{K}$, which results in the set of all possible Kraus matrices $\{\boldsymbol{K}\}$ being a convex set [17].

The objective J is characterized by a suitable quantum mechanical observable O [54], such that

$$J = \mathrm{Tr}\,[\rho O], \tag{2.19}$$

where ρ is the state of the system after the chemical transformation described by the Kraus map $\boldsymbol{K}$. In practice, the initial state ρ_0 of the system may be transformed through alteration of the accessible control variables $\mathbf{c} = [c_1, c_2, \ldots]$ to specify a landscape $J(\rho)$

$$J = J(\mathbf{c}) = J(\boldsymbol{K}^{\mathbf{c}}) = \mathrm{Tr}\,[\Phi_{\boldsymbol{K}^{\mathbf{c}}}(\rho_0)O]\,. \tag{2.20}$$

In Equation (2.20), the variables $\mathbf{c}$ determine the fitness J by acting on the system through the matrix $\boldsymbol{K}^{\mathbf{c}}$, where the superscript indicates that the Kraus matrix depends on the variables $\mathbf{c}$. The explicit relation of $\boldsymbol{K}^{\mathbf{c}}$ to the variables $\mathbf{c}$ will depend on the particular system and may be complicated, but knowledge of this relationship is not necessary to establish the landscape topology of $J(\mathbf{c})$ for OptiChem theory. The formal description of chemical transformations using the analysis below in terms of the set of Kraus matrices $\{\boldsymbol{K}\}$ is general for all chemical systems and synthesis or property optimization objectives.

Satisfaction of the assumptions implies that maximization of J is possible and that any desired set $\{\boldsymbol{K}\}$ may be generated by some choice of $\mathbf{c}$. Combining the convex property of the set $\{\boldsymbol{K}\}$ with the inherent linear dependence of $J(\boldsymbol{K})$ upon $\boldsymbol{K}$ (i.e., from Equation (2.20)) shows that the resulting landscape J contains no local minima or maxima (i.e., traps) and that connected optimal solutions of J (i.e., level sets) may exist on the landscape [37]. This result arises naturally from considering unconstrained convex optimization theory [6]. Importantly, the landscape topological properties hold regardless of the particular nature of the initial density matrix ρ_0 or the chemical objective described by the observable operator O. The general trap-free nature of the landscape can only be shown to hold when $\{\boldsymbol{K}\}$ is a convex set; constraints placed on the variables $\mathbf{c}$ that limit the attainable Kraus matrices could make the set $\{\boldsymbol{K}\}$ non-convex and thereby destroy the trap-free property of the control landscape and/or reduce the set of reachable optimal transformations to isolated set(s) of Kraus matrices.

2.4.3 Predictions of OptiChem Theory

The main conclusions of OptiChem theory are that (i) the control landscape for synthesis and property optimization contains no traps and (ii) the optimal regions of the control landscape may form connected level sets. Importantly, these conclusions hold regardless of the particular nature of the objective or variables employed, and thus generally apply to any chemical optimization application upon satisfaction of Assumptions (1), (2), and (3) above, in addition to the general consideration that the objective value is always finite (i.e., the maximum synthesis yield is 100% and physical or chemical property values are generally finite). When these criteria are met, OptiChem theory predicts the following characteristics of laboratory optimization of chemical objectives:

(a) **Trap-free landscapes readily enable "easy" optimization:** The favorable control landscape topology predicts that the search effort to find optimal values of the variables could be less than that implied by the seeming complexity of the tasks (e.g., based on the curse of dimensionality). In principle, an optimal solution may be found by starting anywhere on the control landscape and employing an efficient search algorithm. This prediction is consistent with the findings that chemical objectives can often be optimized in significantly fewer experiments than implied by the "size" of typical search spaces [38].

(b) **Observed trapping indicates constrained variables:** The fundamental existence of trap-free landscapes provides a framework to understand trapping behavior when it occurs. Importantly, the landscape J is inherently trap-free upon satisfaction of the three assumptions, with failure to satisfy Assumption (3) the most likely breakdown. Traps encountered for this reason are referred to as "false", as is typically seen in free energy landscapes [58]. Although OptiChem theory cannot *a priori* identify the specific limitations on the variables that cause false traps to arise on the landscape, the theory may be used to identify when the current choice of variables is insufficient to meet the desired objective.

(c) **Homologous molecules or materials constitute optimal level sets or multiple solutions:** The possible presence of multiple optimal solutions or level sets on the control landscape is consistent with the common existence of "homologous" chemical systems. Homologous systems constitute a set of distinct members that possess the same chemical properties reflected in the value of J. For example, a large set of chemically related solvents can produce similar high rates of a particular alkylation reaction [70]; this set of solvents may constitute an optimal level set or possibly a set of isolated multiple solutions. In other cases a set of optimal material compositions, solvents, or processing conditions might have much less obvious structural connections, yet still produce the same observable J value.

2.4.4 Illustrations

The degree to which the predictions (a) through (c) of OptiChem theory hold in laboratory investigations has recently been assessed through examination of the extensive literature reporting chemical control landscapes, with the overwhelming finding that reported control landscapes are almost all trap-free [37, 38]. Although these studies cannot claim to have performed an exhaustive search of the literature, overall 142 separate control landscapes were identified, with 123 appearing trap-free and with 19 containing traps (i.e., a clear sub-optimal maximum). In some of the latter cases, the traps can be attributed to variable constraints explicitly discussed by the authors; the remaining works make no mention of the presence or absence of traps. Trap-free landscapes have been reported for the chemical synthesis and property goals listed in Table 2.2, with the literature references given in [38]. This collective literature demonstrates wide applicability of the predictions of OptiChem theory to chemical synthesis and property optimization; a few illustrative studies are summarized here.

Table 2.2 Optimization goals producing trap-free control landscapes in the chemical literature [37, 38]

type	goal
synthesis	organic compounds
	polymers
	material catalytic activity
material properties	luminescence intensity/frequency
	color
	X-ray spectral structure
	mechanical constants
	dielectric constants
	electrical resistivity
	binding between molecules/materials
molecular properties	oxidation potential
	glass transition temperature
	NMR and IR spectra
	protein binding

Control landscapes for chemical synthesis and property objectives are readily generated when the control variables are (quasi)-continuous, such as concentrations of reagents or mole fractions of elements composing a material. The first illustration here considers the mole fractions of three metal components as control variables c_i, $i = 1, 2, 3$ of a catalytic material, with each component varying from $0 \leq c_i \leq 1$ and subject to the constraint $\sum_i c_i = 1$. The objective was to optimize the catalytic activity of the material for oxidizing isobutane to methacrolein, isobutene, and CO_2 [42]. When the elements Mo, V, and Sb were used as variables, all catalytic oxidation landscapes were trap-free, as shown in Figure 2.4(a) through

2.4(c). The authors comment that the active regions for formation of methacrolein in Figure 2.4(a) and CO_2 in Figure 2.4(c) consist of a “large plateau” [42], which according to OptiChem theory would be interpreted as a level set of homologous catalysts. Replacing Sb with Bi as a control variable causes traps to arise on the isobutene landscape, shown in Figure 2.4(e). Furthermore, the maximal catalytic activity on this landscape is only approximately 160% of the reference catalyst, while the maximal activity from the Mo-V-Sb landscape is over 1200% of the reference catalyst. This behavior indicates that the choice of Bi instead of Sb produces a significant constraint on the variables when the objective is to catalyze the formation of isobutene, with Bi limiting the maximum objective yield and introducing traps on the landscape.

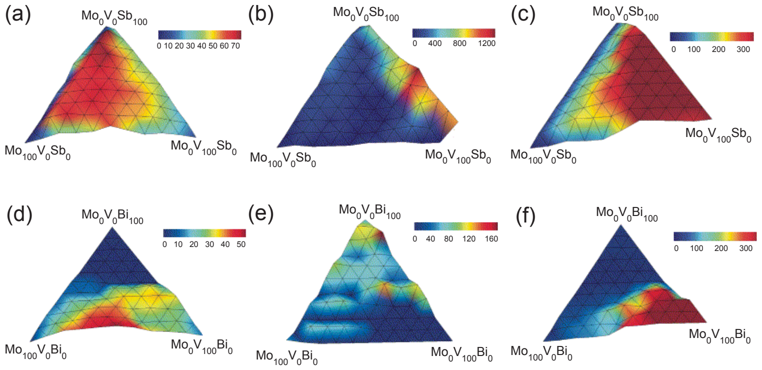
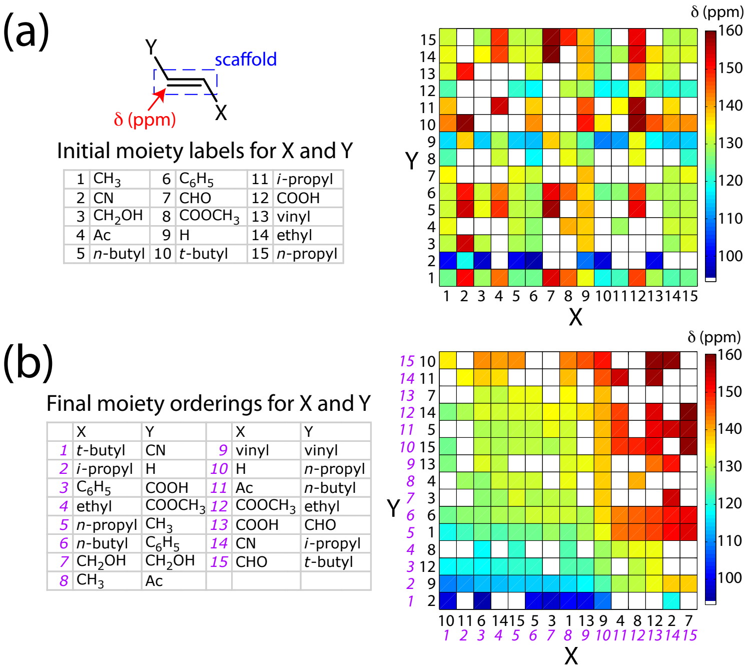


Fig. 2.4 Relative catalytic activity (compared to a fixed reference catalyst) for oxidation of isobutane to form (a) methacrolein, (b) isobutene, and (c) CO_2 with Mo, V, and Sb as the variables, and relative catalytic activity for oxidation of isobutane to form (d) methacrolein, (e) isobutene, and (f) CO_2 with Mo, V, and Bi as the variables. When Mo, V, and Sb are the variables, all landscapes are trap-free, but when Bi replaces Sb, a trap appears in (e) for the isobutene product. Reprinted from [42], © 2004, with permission from Elsevier

Constructing molecular property landscapes is often more difficult than optimizing synthesis conditions or material properties because the variables may not be continuous parameters such as the concentration of reagents, fractional content of the material components, etc. In order to assess for the existence of a trap-free control landscape for such cases with discrete variables, a proper *ordering* $\mathscr{O}$, (i.e., the proper sequential labeling) of the variables is needed. For a family of molecules in a “library” built around a fixed molecular scaffold, the variables can be defined in terms of prescribed integer labels of the chemical moieties on each site. This procedure is illustrated in Figure 2.5 with an example from [36]. The molecular scaffold is a vinyl group, the variable moieties X and Y may take on the integer values 1 through 15 as labeled in Figure 2.5(a), and the objective “property” is spectral, consisting of the ^{13}C nuclear magnetic resonance (NMR) chemical shift of the indicated carbon atom. Plotting the objective property versus the original arbitrarily chosen integer labels for X and Y will generally produce a rough or even random looking landscape, as shown in Figure 2.5(a); such randomness is expected for any randomly ordered landscape, irrespective of any underlying structure. As a result, this initial random assignment of the integer variable labels can hide the true underlying regularity of the landscape if such regularity exists. Accepting that the Assumptions of OptiChem in Section 2.4.2 are satisfied, a regular trap-free landscape under correct ordering of the labels for X and Y is expected, keeping in mind that there will always be a discrete character to the landscape at some level of resolution.



Initial moiety labels for X and Y

1	CH_3	6	C_6H_5	11	*i*-propyl
2	CN	7	CHO	12	COOH
3	CH_2OH	8	$COOCH_3$	13	vinyl
4	Ac	9	H	14	ethyl
5	*n*-butyl	10	*t*-butyl	15	*n*-propyl

Final moiety orderings for X and Y

	X	Y		X	Y
1	*t*-butyl	CN	*9*	vinyl	vinyl
2	*i*-propyl	H	*10*	H	*n*-propyl
3	C_6H_5	COOH	*11*	Ac	*n*-butyl
4	ethyl	$COOCH_3$	*12*	$COOCH_3$	ethyl
5	*n*-propyl	CH_3	*13*	COOH	CHO
6	*n*-butyl	C_6H_5	*14*	CN	*i*-propyl
7	CH_2OH	CH_2OH	*15*	CHO	*t*-butyl
8	CH_3	Ac			

Fig. 2.5 An illustration of the process of reordering integer labels for discrete variables to produce smooth landscapes, in this case for the ^{13}C NMR chemical shift of a vinyl carbon atom. For a molecular scaffold (here a vinyl group), the moiety variables X and Y are given arbitrary integer labels as shown in (a). Permuting the order of the labels on the rows and columns in the landscape (a) results in a the most regular landscape in (b). The white squares correspond to untested compounds. Adapted from [36].

Various algorithms can reorder the variable labels with the aim of producing a control landscape that is as smooth as possible [33, 34, 55]. For the two-dimensional case with moieties X (rows) and Y (columns), reordering involves permuting the rows and columns in order to identify an ordering $\mathcal{O}$ that reveals the most regular property landscape, as exemplified by going from Figure 2.5(a) to 2.5(b)

[33, 34, 36, 55]. In these and other studies, the recovered landscapes were trap-free. It has been shown that surfaces that inherently contain traps are generally not transformed into trap-free surfaces under reordering [11], indicating that reordering cannot artificially transform inherently rugged landscapes into smooth ones. The reordering technique has been applied to a number of molecular property objectives [12, 28, 33, 34, 36, 55].

An investigation on a family of drug candidate molecules demonstrated how a search algorithm based on the reordering technique can potentially accelerate the discovery of molecules with desired properties [34]. The molecular library consisted of a common scaffold with two sites having respectively $X = 93$ and $Y = 151$ distinct moieties, where the objective was to optimize the inhibition of a protein by suitable choices of X and Y. Inhibition values were available for 4110 of the 14,043 molecules (29%) [34]. A reordering algorithm revealed a trap-free landscape to within experimental noise, as shown in Figure 2.6(a). This study also employed an iterative algorithm to illustrate how to accelerate the discovery of effective compounds by taking advantage of automated reordering combined with the potential for finding trap-free control landscapes. In the first step of the algorithm, a small sample of approximately $100-250$ randomly chosen compounds was reordered to identify the most regular landscape. Next, an additional sample of approximately 50 compounds in the vicinity of the best inhibitors on the landscape was added to the existing sampled compounds, and reordering was performed again with the entire collection of available compounds to identify a new landscape. This cyclic addition of a few compounds followed by landscape reordering was shown to significantly reduce the number of syntheses needed to identify the most effective compounds: the method discovered 70 effective compounds (50% of the total number of effective compounds in the entire library) with just 650 syntheses, far fewer than the approximately 2000 syntheses needed to identify the same number of effective members with random screening, as shown in Figure 2.6(b) [34].

2.4.5 Perspective

A generic favorable topology of control landscapes for chemical optimization has been found to arise upon satisfaction of the three Assumptions set out in Section 2.4.2. The widespread satisfaction of the predictions of OptiChem theory evident from the numerous landscapes reported in the literature [37, 38] indicates that the assumptions may be expected to hold under many reasonable experimental conditions and for a wide variety of optimization objectives. The existence of a trap-free landscape topology provides an organizing principle to help understand why even complex chemical objectives are often achieved with significantly fewer experiments than would be expected based on the apparent mathematical complexity of these problems.

OptiChem theory potentially has important practical implications for the optimization of chemical synthesis and property objectives. Recent work showing that an iterative method can accelerate the identification of optimal molecules or materials [34] and facilitate accurate NMR spectral peak prediction [36] represents just

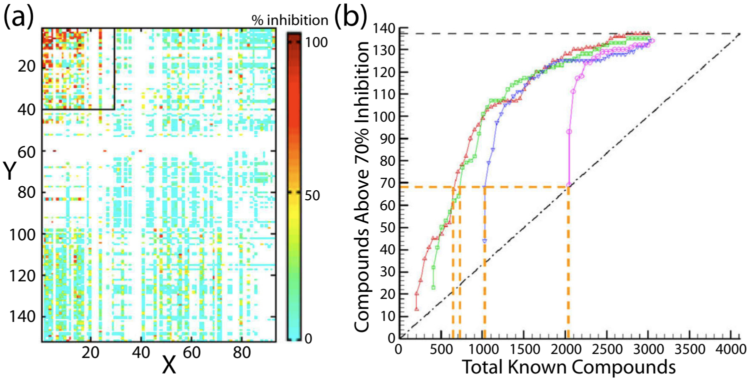


Fig. 2.6 (a) Optimally reordered landscape for the objective of protein binding efficiency from [34]. 90% of the effective compounds (i.e., with inhibition $> 70\%$) are in the box in the upper left hand corner of the landscape. (b) The number of effective compounds versus the total number of "known" compounds. The dash-dot line indicates the expected number of effective compounds upon random sampling. The red, green, blue, and purple curves correspond to initial samplings of 5%, 10%, 25%, and 50% of the available data, respectively. The algorithm is most efficient when starting with the smallest initial sampling (red), and identifies half of the known effective compounds with only 650 syntheses instead of the ~ 2000 expected from random sampling.

the beginning of many potential applications of OptiChem theory. Additionally, the understanding that a well-posed chemical optimization objective should produce a trap-free control landscape could be important for designing effective optimization search algorithms, as well as taking advantage of well-known empirical "rules" of chemistry [38]. The existence of a trap-free control landscape is of particular importance for interpreting cases where traps *are* observed on the landscape. First, landscape traps may imply that inadequate control resources are being employed (i.e., Assumption (3) is violated), as was observed in Figure 2.4(e) above and discussed in [42, 57]. In such cases, OptiChem theory may be used to prevent the waste of experimental resources in a suboptimal portion of the available search space. Second, individual outliers on the landscape may indicate errors in the experimental data, as was shown in [36], where the outlier nature of the associated data points was only revealed upon applying the landscape principles. Outlier points on the landscape may also reveal previously hidden distinct physical mechanisms underlying the control objective, as was found in a study optimizing photoluminescent quantum yield in transition metal complexes [33]. In these cases, OptiChem theory provides a unique method to identify when further examination of the outliers may be necessary to determine the origins of the observed behavior.

OptiChem theory is an organizing principle that provides (a) a basis to explain why optimization in chemistry is often easier to achieve than expected and (b) a foundation to predict that many other well-posed control objectives, possibly including those previously eschewed as too complex to address, may be attainable goals.

The promising assessment from the chemical literature indicates that OptiChem theory should find many applications in chemistry, including when the target molecules are biological, as discussed further in Section 2.5 below.

2.5 Biological Fitness Landscapes (OptiEvo)

2.5.1 Background

This section focuses on the application of OptiSci to the fitness landscape topology of biological systems. The topology of fitness landscapes has been an important topic in evolutionary biology since Sewall Wright first introduced the concept in 1932 [66]. In most studies, a genotypic fitness landscape represents the mapping from an organism's sequence space to its fitness value. Consequently, natural evolution can be viewed as an excursion over the fitness landscape, whose topology is critical for determining the outcome and dynamics of evolution.

In recent years, the practical field of directed evolution has become important for engineering proteins and other bio-molecules and bio-networks [2]. Similar to the case of natural evolution, a fitness landscape in directed evolution captures the relationship between the target fitness value and the sequence space for the associated genes. If the fitness landscape is globally smooth and monotonic, then simple strategies, possibly including local search algorithms, should be effective in guiding mutations to reach the desired fitness optimum. If the landscape topology is "rugged" instead, then optimizing the target property can be much more difficult.

Due to the large size of the genome in natural evolution (or the target genes in directed evolution) and the exponential relationship between the total number of possible genotypes and the genome/gene size, it has been extremely difficult to establish the topology of a full genotypic fitness landscape. Since the relationship between fitness and genotype is nonlinear and highly complex, fitness landscapes should intuitively contain many local fitness peaks and valleys and fitness convergence should rarely been seen, if at all, in the course of evolution. However, some laboratory evolution studies have observed fitness convergence in duplicate populations living in the same environment [13].

To explain this contradiction, most models reasonably *assume* a globally rugged landscape and discuss how organismic populations can escape the fitness valleys [27, 63]. The only previous model that seeks to directly deduce the topology of fitness landscapes in evolution is based on statistics [19]. This model shows that fitness peaks in high-dimensional landscapes are statistically likely to be connected by ridges of similar height, thereby alleviating the problem of population escape from local peaks in some cases.

Below we summarize recent work utilizing the concepts and mathematical framework of OptiSci to study the topology of fitness landscapes in both natural and directed evolution, and we refer to the resultant theory as **Opti**mal Landscapes in **Evo**lution (OptiEvo). In OptiEvo, an organism (or the target gene(s) in directed evolution) is viewed as a large open system that interacts with its environment, where fitness is determined by the set of all relevant nucleotides and the genotype-by-environment interactions. Here, natural evolution corresponds to a stochastic process that seeks the best genotypes to maximize the fitness value. Counter-intuitively, OptiEvo shows mathematically that the topology of fitness landscapes inherently should not contain local traps when a proper genetic/genomic flexibility condition is satisfied that captures the OptiSci assumptions in Section 2.2. We also compared several OptiEvo predictions with results collected from laboratory evolution experiments, indicating that the sufficient flexibility condition can be satisfied in many evolutionary processes.

2.5.2 Physical Foundation of OptiEvo Theory

The topological analysis of evolution landscapes can be performed classically or quantum-mechanically (see Sections 2.2 and 2.4). Here it is more natural to adopt a classical treatment. In OptiEvo theory, an organism's population with N individuals is represented by a set of variables $G = \{g_1, \dots, g_N\}$, where $g_i = (n_1, \dots, n_{r_i}, \dots, n_{R_i})$ is an R_i-dimensional vector that defines all the variables (nucleotides, with n_{r_i} being 1, 2, 3 or 4) of the i-th individual organism in the population and R_i is the genome length of this individual. The state of a population in a constant environment is represented by a probability distribution $\rho_G(\omega)$, where ω is a point (a microscopic state) in Ω, which denotes the "state space" containing all biologically relevant degrees of freedom for the population. As stated in Section 2.2, the fitness value J can be viewed as a characteristic physical observable of the population, which can always be described by the expectation value of the fitness function $f(\omega)$ defined over Ω [41]:

$$J(\rho_G) = \int_{\Omega} f(\omega)\rho_G(\omega)d\omega. \tag{2.21}$$

Note that this equation is *not* a model, but a truism from statistical physics, applicable universally to any classical physical observable.

In an individual step of evolution, a change of the population $G \to G'$ occurs (via gene mutations, deletions, insertions, etc.), leading to a change of the probability distribution $\rho_G \to \rho_{G'}$ and the fitness value $J(\rho_G) \to J(\rho_{G'})$. From an optimization perspective, the process of evolution aims to find the best variable set(s) G^* to maximize the objective function $J(\rho_G)$. In Section 2.5.3, we will show that two basic conclusions can be drawn on the topology of such landscapes [15]: (1) The fitness function $J(\rho_G)$ only has global maxima on F, the union of all possible genotypes G; local maxima do not exist. (2) The set of optimal genotypes G^* is a connected level set with the same fitness value. Very importantly, to reach these conclusions, no species-specific knowledge of $J(\rho_G)$, Ω, $f(\omega)$, and $\rho_G(\omega)$ is needed beyond the generic form in Equation (2.21).

As described in Section 2.2, three basic Assumptions need to be satisfied in order to reach the two OptiEvo conclusions in realistic cases. These assumptions are presented in a control framework in keeping with the other domains of OptiSci.

Importantly, in the context of biological evolution the notion of *control* needs to be understood as a mathematical formulation to assess the associated fitness landscapes.

(1) **The system is controllable:** Biologically, the global fitness optima can be reached by some genotype.
(2) **The control end-point map is surjective:** Biologically, a small change in the population state $\rho_G(\omega)$ has a small corresponding change in the genotype G.
(3) **The controls are fully adequate:** Biologically, the physically realizable genotype changes $G \to G'$ can provide sufficient coverage of all important local directions on the fitness landscape $J(\rho_G)$ around ρ_G.

In the case of biological evolution, the Assumptions (1), (2) and (3) above can be subsumed into a "sufficient flexibility condition", meaning that the genome structure, the accessible gene changes, and the genotype-to-environment interactions allow for sufficient flexibility, so that the fitness landscape around J can be freely explored. We need to emphasize that, only when this condition is combined with the inherent topological properties of Equation (2.21), can conclusions (1) and (2) of OptiEvo theory be reached. In particular, violation of the flexibility condition, due to inherent constraints in gene changes, can create local traps that prevent the global fitness optima from being reached. It needs to be emphasized that, upon satisfaction of the flexibility condition, the two OptiEvo conclusions above are inherent properties of the corresponding genotypic fitness landscapes and are independent of evolution dynamics (i.e., independent of how an organism moves on a fitness landscape). However, the rate that evolution climbs the landscape will depend on the dynamics. Section 2.5.3 summarizes the mathematical proof leading to the above OptiEvo conclusions (the full proof is presented in [15]), and Section 2.5.4 provides empirical assessments of OptiEvo theory.

2.5.3 Evolutionary Fitness Landscape Analysis

Classical probability space is defined by a triple $(\Omega, \mathscr{F}, P)$, where Ω is a non-empty set (the phase space), $\mathscr{F}$ is a σ-algebra of subsets of Ω, and $P : \mathscr{F} \to [0,1]$ is a probability measure [53]. Each classical stochastic system is characterized by some set of elementary events Ω, where any point $\omega \in \Omega$ denotes a specific microscopic state of the system. In biological systems, such a phase space would be high dimensional and complex. However, what counts here is the existence of the phase space rather than its details.

Biologically measurable properties for an evolving organism's population are given by average values of an associated random function. If $f(\omega)$ $(\omega \in \Omega)$ is a random function representing the fitness and if the organism's population is in a state with probability distribution $\rho_G(\omega)$, then the fitness value is determined by Equation (2.21) and will be maximized during the course of evolution. Genetic changes during evolution will produce successive transformations of the genome $G \to G'$ corresponding to a trajectory over the landscape leading to maximization of $J(G)$.

A probability space $(\Omega, \mathscr{F}, P)$ should satisfy the following three axioms [29]:

A$_1$. $P(E) \geq 0$ for any $E \in \mathscr{F}$
(i.e., probability of any event is a non-negative number).

A$_2$. $P(\emptyset) = 0$
(here $\emptyset$ denotes the empty set) and $P(\Omega) \equiv \int\limits_{\Omega} \rho(\omega) d\omega = 1$ (the assumption of unit measure states that the probability that some event will occur is 1).

A$_3$. For any countable sequence of pairwise disjoint events $E_1, E_2, \dots \in \mathscr{F}$:

$$P(E_1 \cup E_2 \cup \dots) = P(E_1) + P(E_2) + \dots$$

A subset $X \subset \mathbb{V}$ of a linear space $\mathbb{V}$ is a convex set if for any $x_0, x_1 \in X$ and any $\lambda \in [0,1]$ the point $x_\lambda := (1-\lambda)x_0 + \lambda x_1$ is in X [61]. According to this definition, the set $\mathscr{P}_\Omega$ is a convex subset in the linear space $\mathscr{P}_\Omega^{\mathrm{R}}$ of all distributions associated to real-valued signed measures over $(\Omega, \mathscr{F})$. Let P_0 and P_1 be any two probability measures over Ω with associated probability distributions $\rho_0(\omega)$ and $\rho_1(\omega)$, and define $P_\lambda = \lambda P_1 + (1-\lambda)P_0$ for any $0 \leq \lambda \leq 1$. With this definition, we can prove that the set $\mathscr{P}_\Omega$ satisfies the three Axioms above and has a convex structure [15].

The fitness function $J(\rho)$ in Equation (2.21) is linear in ρ and is defined on the convex domain $\mathscr{P}_\Omega$. Since (a) J is a concave functional and (b) a concave functional does not have local maxima over a convex domain [48, 53], the two OptiEvo conclusions in Section 2.5.2 can be proved [15].

The absence of local maxima for the objective function $J(\rho)$ means that in a neighborhood of any ρ there exists a direction in which the objective $J(\rho)$ increases through a change in the genotype $G \to G'$. The sufficient flexibility condition states that the neighborhood U_G of G covers, on some coarse grained scale, all directions in the space $\mathscr{P}_\Omega$ around ρ_G, particularly allowing for movement in the direction in which $J(\rho)$ increases. Significant constraints on the available genotypes G (violating the sufficient flexibility condition), which reduce the domain of probability distribution functions P_Ω to a non-convex set, may introduce traps in the genotypic landscapes.

2.5.4 Predictions and Empirical Assessments of OptiEvo Theory in Natural Evolution

Since it is impossible to express the fitness of an organism's population explicitly in terms of the genotype, we cannot *a priori* determine whether the sufficient flexibility condition is satisfied in evolution. However, we can perform assessments by comparing biological predictions of OptiEvo theory (assuming satisfaction of the sufficient flexibility condition) with results from evolution experiments.

Four general predictions arise from OptiEvo theory upon consideration of the sufficient flexibility condition:

(a) **Coexistence of global fitness convergence and genotypic diversity:** OptiEvo theory implies that, given adequate time, evolution of an organism's population in a constant homogeneous environment should always converge to the same globally optimal fitness value.
(b) **Connected global optima:** The different genotypes at the top of a fitness landscape are connected. Thus, evolving between them does not require going through valley genotypes.
(c) **Non-existence of reciprocal sign epistasis:** The phenomenon of epistasis (Figure 2.7) arises when the fitness effect of a mutation depends on its genetic background [63]. Certain types of epistasis can slow down the dynamics of evolution (Figures 2.7.i and ii). According to OptiEvo theory, however, the only type of epistasis that cannot exist is reciprocal sign epistasis (Figure 2.7.iii), which will create local fitness peaks/traps that prevent the global fitness optima from begin reached.
(d) **Condition for the existence of local sub-optimal peaks:** Local sub-optimal fitness peaks or separated global maxima can exist only when the sufficient flexibility condition is violated, i.e., constraints in the evolutionary changes of the genome prevent accessing important domains of the fitness landscape. Statistically, such violations are more likely to occur for smaller genomes.

Below we briefly compare predictions (a)–(d) of OptiEvo theory with results obtained from laboratory evolution studies [13]. A summary of these experiments along with a more detailed discussion of their relationship to OptiEvo theory is presented in [15].

A classic well known laboratory evolution experiment has been running over 30,000 generations since 1988 on twelve initially identical *Escherichia coli* populations in identical glucose-limited minimal medium [65]. It was observed that the fitness converged to similar values, while cell morphology, physiology, mutation rates, and gene sequence had large variations across the populations [32]. While these results are not easily explainable by most models of fitness landscapes, they agree well with prediction (a) of OptiEvo theory. Reverse evolution experiments also show that ancestral fitness values can be recovered given sufficient evolution time [60]. In addition, compensatory mutations occur more often than back mutations [10], indicating the existence of multiple evolutionary trajectories. These results are consistent with OptiEvo theory predictions (a) and (c).

Prediction (b) of OptiEvo theory states that different optimal genotypes should be fully connected. This prediction is supported by recent experiments on *Antirrhinum* [64] species, showing that phenotypically distinct species well adapted in similar environments may be adjacent in genotypic space.

In some evolution experiments, convergence of fitness was not observed. A well-cited experiment involved the RNA bacteriophage $\phi 6$, where two genetically different virus populations converged to different fitness maxima in the same environment [9]. OptiEvo theory provides two possible explanations. First, the evolution experiment (about 100 generations) may be too short to allow for convergence of the fitness value. Second, local fitness optima may exist in this system due to a violation of the flexibility condition. According to prediction (d) above, the small genome of the virus makes it more susceptible to various genetic constraints and can cause

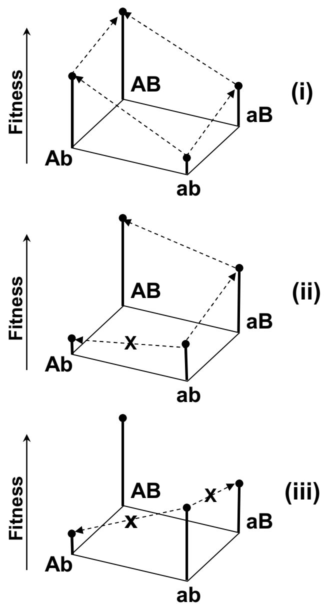


Fig. 2.7 Magnitude and sign epistasis for a two-nucleotide, two-allele system. Evolution starts from the genotype ab and ends with AB. (i): Magnitude epistasis leads to different selection coefficients for the two pathways (ab→Ab→AB and ab→aB→AB), but both pathways are evolutionarily accessible because the fitness value always increases in each evolution step. (ii): Sign epistasis makes the pathway ab→Ab→AB inaccessible, while ab→aB→AB is still accessible. (iii): Reciprocal sign epistasis makes both pathways inaccessible and isolates the two genotypes ab and AB, which according to OptiEvo theory is the only type of epistasis that cannot exist if the sufficient flexibility condition is satisfied. Adopted from Figure 1 of [15].

some of the transformations $G \to G'$ to be inaccessible, creating local traps in the corresponding fitness landscape.

2.5.5 Application of OptiEvo Theory to Directed Evolution

In addition to providing topological analysis of fitness landscapes in natural evolution, OptiEvo theory also has important implications for strategies of directed evolution of proteins, nucleic acids, and complex biological networks. This section will summarize these implications and describe a general substituent reordering strategy (whose applicability depends critically on OptiEvo's conclusions) to facilitate property optimization (including directed evolution) of biological systems.

OptiEvo has two main implications for directed evolution:

(1) OptiEvo states that constraints in the mutations $G \to G'$ may introduce local traps in the evolution steps. In directed evolution, OptiEvo suggests that performing mutations on a large number of nucleotides (corresponding to fewer constraints) can be more effective in reaching the global fitness optimum, although possibly at the cost of slower initial fitness increase.
(2) OptiEvo suggests that, if the sufficient flexibility condition is satisfied, the property landscape will be globally regular (i.e., trap free). From an optimization point of view, a regular property landscape will allow for local search /pattern recognition / interpolation methods to efficiently optimize the target property. In contrast, if the property landscape is rugged, then these tools will have little predictive power without detailed knowledge of the structure-property relationships.

Utilizing the second implication above, we have developed a general and easy-to-use substituent reordering strategy for property optimization through directed evolution of proteins and other biosystem components. The same strategy has been employed for property optimization of chemical compounds and materials in Section 2.4. Depending on the requirements of the applications, different algorithms have been designed to enable optimal performance in guiding directed evolution experiments. The text below will describe one such algorithm most relevant to directed evolution of biological systems, especially proteins.

A substituent reordering algorithm was designed especially for optimization of protein properties in focused libraries [16]; also see Section 2.4.4 for like applications to molecular discovery. The algorithm was evaluated in a proof-of-principle study to optimize the enatioselectivity (E-value) of the expoxide hydrolase from *Aspergillus niger*. To be experimentally manageable, we selected two amino acid positions from the wild-type enzyme that are known to contribute to its E-value. Each amino acid on each protein backbone position is assigned a random but unique integer between 1 and 20, and the total number of possible mutants from the two positions is 400. We then synthesized a random subset of the 400 mutants and measured the E-value for each of them. The property landscape for the mutants is irregular (Figure 2.8.A) due to the random integer assignments to the amino acids and offers no predictive power. However, using the reordering algorithm, we rearranged (permuted) the integer assignments and obtained a smooth property landscape, which immediately points to the lower right corner of the reordered landscape as the region with better mutants (Figure 2.8.B). Based on the prediction, we then synthesized 45 additional mutants, most of which were then found to be within the predicted region with high E-values. Figure 2.8.C shows that the algorithm even identified a laboratory error (a misplaced mutant) based on finding local landscape irregularity.

Importantly, the substituent reordering algorithm does not utilize any structure information of the target protein; the algorithm operates to identify an optimal integer assignment to each amino acid, so that a specified landscape regularity measure is optimized (i.e., the smoothest landscape is identified over the mutant samples). As

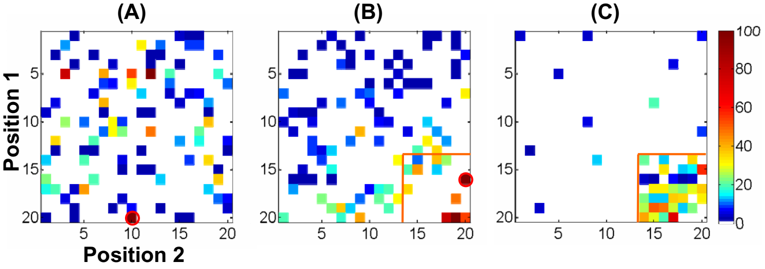


Fig. 2.8 Optimal reordering of the E-value landscapes for an epoxide hydrolase [16]. (A) Color heat map for the E-value landscape of 95 randomly sampled mutants plotted with a random amino acid ordering. Each color square represents one mutant with red indicating high E-value and blue corresponding to low E-value (see color bar on the far right). White squares are unsampled proteins. (B) E-value landscape of the 95 mutants using the amino acid ordering identified by the substituent reordering algorithm. The result predicts that proteins with high E-values are most likely located in the lower right corner. The mutant at position [16,20] (circled in red in both (A) and (B)) of the reordered landscape turned out to be the same as the mutant at position [20,19]; the wrong protein was accidentally placed in this position in the experiment. (C) E-value landscape for 45 newly sampled mutants, guided by the ordering in (B). Adopted from Figure 2 of [16].

stated in Section 2.4 and demonstrated in a computational study [11], if the underlying property landscape contains traps due to violation of the assumptions, it is highly likely that even the smoothest reordered landscape will remain rugged, giving little predictive power. Hence, the fact that the substituent reordering algorithm yielded a smooth property landscape (within data noise) and correctly predicted the region with the best E-values implies that the sufficient flexibility condition appeared to be satisfied even in this highly constrained two-dimensional property landscape.

So far, the substituent reordering strategy has been applied mostly to two-dimensional cases. To improve the performance, especially minimization of sampling in higher-dimensional systems, we have recently integrated the substituent reordering algorithm with a high-dimensional model representation (HDMR) method [24]. HDMR decomposes nonlinear high-dimensional functions into a hierarchy of lower-dimensional (but still nonlinear) functions, and substituent reordering can be performed on each lower-dimensional function separately. Very importantly, HDMR samples all the variables simultaneously (e.g., mutations are performed on all substitution positions simultaneously) and it enables highly favorable scalability of the mutant sampling as the dimensionality of the system increases [24]. The capabilities of the HDMR-reordering algorithm was illustrated in a simulation study with high-dimensional functions [24]. The study showed that, for a nonlinear, 20-dimensional simulated protein library with 20^{20} members, the HDMR-reordering algorithm can provide reliable predictions from a very small number of samples ($\sim 10^4$).

2.5.6 Perspective

The physically-grounded and mathematically based OptiEvo theory was introduced to analyze the general topology of fitness landscapes in biological evolution [15]. OptiEvo theory views an organism's population as an open system and natural evolution as a stochastic process of fitness optimization. The theory states that, when gene changes in evolution can provide sufficient flexibility to adequately access local landscape structure, then (1) genotypic fitness landscapes should be free of local traps and (2) the globally optimal genotypes are connected. The extreme complexity of biological systems makes it impractical to perform mathematical or model-based assessment of optiEvo theory and the sufficient flexibility condition. However, a comparison of several predictions of OptiEvo with empirical evidence from laboratory evolution experiments implies that the sufficient flexibility condition can be satisfied in many evolutionary processes. Consequently, OptiEvo theory provides a physical explanation for (a) the apparent contradiction between fitness convergence and genome complexity in many evolution processes, as well as (b) the potential cause of local fitness traps (i.e., violation of the sufficient flexibility condition).

OptiEvo theory also has important implications for directed evolution: (1) performing mutations on a larger number of amino acid or nucleic acid positions is more likely to alleviate the introduction of local traps in evolution, and (2) satisfaction of the sufficient flexibility condition may allow for local search and interpolation methods to effectively guide the property optimization of the target biomolecules or bionetworks. Building on this foundation, OptiEvo lends itself to a general substituent reordering strategy for optimizing target properties of complex biosystems with minimal sampling effort and without requiring explicit information about the structure-property relationships [16, 24]. In cases where gene mutations occur at many sites in the directed evolution experiments, the order of $\sim 10^3 - 10^4$ samples may be required for reliable prediction. However, with the staggering pace of developments in high-throughput technologies, we expect that such experiments can be performed with reasonable overhead in the near future.

In the development of OptiEvo theory and its assessment by empirical data, we have considered the environment as constant for an organismic population. When the environment changes, the geometry of the fitness landscape may also change, thereby altering evolutionary dynamics. However, when the flexibility condition remains satisfied, the trap-free topology of the landscape will in turn remain according to OptiEvo theory.

2.6 Conclusion

This chapter summarized recent advances in research exploring the topology of control landscapes. The key finding in these studies is that upon satisfaction of the assumptions (1), (2) and (3) given in Section 2.2, a mathematical analysis can be executed drawing general conclusions about the topology of the landscapes. In particular, under the specified criteria, then the control landscapes should be generically free of suboptimal trapping critical points. Considerable quantitative experimental and simulation evidence over wide-ranging applications summarized in the chapter demonstrates that the underlying assumptions may often be satisfied providing confirmation of the very attractive landscape topological conclusions. The implications of these findings have fundamental as well as practical importance.

The OptiSci landscape formulation is best referred to as a "principle" rather than a theory, despite the use of the latter word in the preceding sections. This distinction is made, as the landscape topological conclusions naturally follow mathematically upon satisfaction of the three key assumptions. Thus, an observed violation of the predicted topological character in any particular application would imply that one or more of the assumptions is not satisfied. This circumstance has important implications. For example, the observed violation of the predicted topology would provide an early indication of an ill-designed experiment, which could be especially relevant in cases of expensive or time-consuming applications. In turn, the identification of a particular minimal set of control variables that permit trap-free excursions over a landscape could be viewed as specifying a "rule" for the application. In this regard, it may be possible to consider the construction of automated apparatuses whose goal is to identify systematic control rules for particular applications.

Although the landscape topology may be assessed generically upon satisfaction of the key assumptions, an additional challenge is to understand how partial satisfaction of the assumptions (e.g. a system could be partially controllable regarding Assumption (1)) impacts the landscape topology. Such limitations can lead to anomalous landscape features, including what may be referred to as "false traps" when they arise from constraints being present. Some applications may not permit overcoming the presence of practical constraints, while in other cases knowledge of the impact of constraints may motivate the extra effort or the gathering of added resources to overcome the constraints in order to reach the best performance through an unfettered search over the landscapes. A related matter concerns the nature of the non-topological features on the landscapes (i.e., the "twists and turns" that do not have critical point character). Although the latter features would not inherently stop even a local algorithm from reaching optimal locations on the landscape, they could significantly impact the efficiency of such procedures. A full analysis of non-topological landscape features remains to be explored.

We hope that the developments reported in this chapter provide the basis for considering the unification of landscape studies across multiple disciplines. These cross connections may open up new applications which were heretofore viewed as unreachable due to the diversity of variables involved or the anticipated complexity of the landscapes.

Acknowledgements. The authors acknowledge support from the NSF, DOE, ARO, and DARPA. RBW acknowledges support from NSFC (No.60904034, 61134008). We are also indebted to Ruixing Long for his invaluable comments and suggestions.

References

[1] Altafini, C.: Controllability and simultaneous controllability of isospectral bilinear control systems on complex flag manifolds. Systems and Control Letters 58(3), 213–216 (2009)

[2] Arnold, F., Georgiou, G.: Directed Enzyme Evolution: Screening and Selection Methods. Humana Press, Totowa (2003)

[3] Assion, A., Baumert, T., Bergt, M., Brixner, T., Kiefer, B., Seyfried, V., Strehle, M., Gerber, G.: Control of chemical reactions by feedback-optimized phase-shaped femtosecond laser pulses. Science 282(5390), 919–922 (1998)

[4] Bardeen, C.J., Yakovlev, V.V., Wilson, K.R., Carpenter, S.D., Weber, P.M., Warren, W.S.: Feedback quantum control of molecular electronic population transfer. Chemical Physics Letters 280(1-2), 151–158 (1997)

[5] Bellman, R.E.: Dynamic Programming. Princeton University Press, Princeton (1957)

[6] Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)

[7] Brif, C., Chakrabarti, R., Rabitz, H.: Control of Quantum Phenomena, pp. 1–76. John Wiley & Sons, Chichester (2011)

[8] Brixner, T., Gerber, G.: Quantum control of gas-phase and liquid-phase femtochemistry. Chem. Phys. Chem. 4(5), 418–438 (2003)

[9] Burch, C., Chao, L.: Evolvability of an RNA virus is determined by its mutational neighbourhood. Nature 406, 625–628 (2000)

[10] Davis, B.H., Poon, A.F.Y., Whitlock, M.C.: Compensatory mutations are repeatable and clustered within proteins. Proc. Royal Soc. B: Biological Sciences 276, 1823–1827 (2009)

[11] DiMaggio, P.A., McAllister, S.R., Floudas, C.A., Feng, X.J., Rabinowitz, J.D., Rabitz, H.A.: Biclustering via optimal re-ordering of data matrices in systems biology: rigorous methods and comparative studies. BMC Bioinformatics 9, 458–1–16 (2008)

[12] Dooley, C.T., Chung, N.N., Wilkes, B.C., Schiller, P.W., Bidlack, J.M., Pasternak, G.W., Houghten, R.A.: An all D-amino acid opioid peptide with central analgesic activity from a combinatorial library. Science 266, 2019–2022 (1994)

[13] Elena, S.F., Lenski, R.E.: Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation. Nat. Rev. Genetics 4, 457–469 (2003)

[14] Fano, U.: Description of states in quantum mechanics by density matrix and operator techniques. Rev. Mod. Phys. 29, 74–93 (1957)

[15] Feng, X., Pechen, A., Jha, A., Wu, R., Rabitz, H.: Global optimality of fitness landscapes in evolution. Chem. Sci. 3, 900–906 (2012)

[16] Feng, X., Sanchis, J., Reetz, M., Rabitz, H.: Enhancing the efficiency of directed evolution in focused enzyme libraries by the adaptive substituent reordering algorithm. Chemistry 18, 5646–5654 (2012)

[17] Fujiwara, A., Algoet, P.: One-to-one parametrization of quantum channels. Phys. Rev. A 59, 3290–3294 (1999)

[18] Gabasov, R., Kirillova, F.: Higher order necessary conditions for optimality. SIAM J. Control 10, 127–168 (1972)

[19] Gavrilets, S.: Fitness Landscapes and the Origin of Species. Princeton University Press, Princeton (2004)

[20] Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Kluwer Academic Publishers, Boston (1989)

[21] Ho, T.S., Dominy, J., Rabitz, H.: Landscape of unitary transformations in controlled quantum dynamics. Phys. Rev. A 79, 013422–1–16 (2009)

[22] Ho, T.S., Rabitz, H.: Why do effective quantum controls appear easy to find? Journal of Photochemistry and Photobiology A: Chemistry 180(3), 226–240 (2006)

[23] Huang, G.M., Tarn, T.J., Clark, J.W.: On the controllability of quantum-mechanical systems. Journal of Mathematical Physics 24(11), 2608–2618 (1983)

[24] Izmailov, S., Feng, X., Li, G., Rabitz, H.: A scalable algorithm for molecular property estimation in high dimensional scaffold-based libraries. J. Math. Chem. 50, 1765–1790 (2012)

[25] Judson, R.S., Rabitz, H.: Teaching lasers to control molecules. Phys. Rev. Lett. 68, 1500–1503 (1992)

[26] Kadanoff, L.P.: Statistical Physics: Statics, Dynamics and Renormalization. World Scientific, Singapore (2000)

[27] Kauffman, S., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. J. Theor. Biol. 128, 11–45 (1987)

[28] Kim, W., Kim, Y., Min, J., Chang, Y.T., Hecht, M.H.: A high-throughput screen for compounds that inhibit aggregation of the Alzheimer's peptide. ACS Chem. Biol. 1, 461–469 (2006)

[29] Kolmogorov, A.N.: Foundations of the Theory of Probability. Chelsea Pub. Co., New York (1950)

[30] Kosloff, R., Rice, S., Gaspard, P., Tersigni, S., Tannor, D.: Wavepacket dancing: Achieving chemical selectivity by shaping light pulses. Chemical Physics 139(1), 201–220 (1989)

[31] Kraus, K.: States, Effects and Operations: Fundamental Notions of Quantum Theory. Springer, Berlin (1983)

[32] Lenski, R.: Phenotypic and genomic evolution during a 20,000-generation experiment with the bacterium *Escherichia coli*. Plant Breeding Reviews 24, 225–265 (2004)

[33] Liang, F., Feng, X., Lowry, M., Rabitz, H.: Maximal use of minimal libraries through the adaptive substituent reordering algorithm. J. Phys. Chem. B 109, 5842–5854 (2005)

[34] McAllister, S.R., Feng, X., DiMaggio, P.A., Floudas, C.A., Rabinowitz, J.D., Rabitz, H.A.: Descriptor-free molecular discovery in large libraries by adaptive substituent reordering. Bioorg. Med. Chem. Lett. 18, 5967–5970 (2008)

[35] Moore, K.W., Chakrabarti, R., Riviello, G., Rabitz, H.: Search complexity and resource scaling for the quantum optimal control of unitary transformations. Phys. Rev. A 83, 012326–1–15 (2011)

[36] Moore, K.W., Li, R., Pelczer, I., Rabitz, H.: NMR landscapes for chemical shift prediction. J. Phys. Chem. A 136(16), 9142–9157 (2012)

[37] Moore, K.W., Pechen, A., Feng, X.J., Dominy, J., Beltrani, V., Rabitz, H.: Universal characteristics of chemical synthesis and property optimization. Chemical Science 2, 417–424 (2011)

[38] Moore, K.W., Pechen, A., Feng, X.J., Dominy, J., Beltrani, V., Rabitz, H.: Why is chemical synthesis and property optimization easier than expected? Phys. Chem. Chem. Phys. 13(21), 10048–10079 (2011)

[39] Moore, K.W., Rabitz, H.: Exploring quantum control landscapes: Topology, features, and optimization scaling. Phys. Rev. A 84, 012109–1–14 (2011)

[40] Moore, K.W., Rabitz, H.: Exploring constrained quantum control landscapes. J. Chem. Phys. 137(13), 134113–1–16 (2012)

[41] Papoulis, A.: Expected Value; Dispersion; Moments. McGraw-Hill, New York (1984)

[42] Paul, J.S., Jacobs, P.A., Weiss, P.W., Maier, W.F.: Combinatorial discovery of new catalysts for the selective oxidation of isobutane. Appl. Catal. A: Gen. 265(2), 185–193 (2004)

[43] Peirce, A.P., Dahleh, M.A., Rabitz, H.: Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications. Phys. Rev. A 37, 4950–4964 (1988)

[44] Rabitz, H., Ho, T.S., Hsieh, M., Kosut, R., Demiralp, M.: Topology of optimally controlled quantum mechanical transition probability landscapes. Phys. Rev. A 74, 12721–12721 (2006)

[45] Rabitz, H., Ho, T.S., Long, R., Wu, R., Brif, C.: Comment on are there traps in quantum control landscapes?. Phys. Rev. Lett. 108, 198901–1–2 (2012)

[46] Rabitz, H., Hsieh, M., Rosenthal, C.: Quantum optimally controlled transition landscapes. Science 303, 1998–2001 (2004)

[47] Ramakrishna, V., Salapaka, M.V., Dahleh, M., Rabitz, H., Peirce, A.: Controllability of molecular systems. Phys. Rev. A 51, 960–966 (1995)

[48] Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)

[49] Roslund, J., Rabitz, H.: Phys. Rev. A. (to be submitted)

[50] Roslund, J., Rabitz, H.: Experimental quantum control landscapes: Inherent monotonicity and artificial structure. Phys. Rev. A 80, 013408–1–9 (2009)

[51] Rothman, A., Ho, T.S., Rabitz, H.: Observable-preserving control of quantum dynamics over a family of related systems. Phys. Rev. A 72, 023416–1–15 (2005)

[52] Schirmer, S.G., Fu, H., Solomon, A.I.: Complete controllability of quantum systems. Phys. Rev. A 63, 063410–1–8 (2001)

[53] Schulz, M.: Control Theory in Physics and other Fields of Science: Concepts, Tools, and Applications. Springer, Berlin (2006)

[54] Schwabl, F.: Quantum Mechanics. Springer, Berlin (2002)

[55] Shenvi, N., Geremia, J.M., Rabitz, H.: Substituent ordering and interpolation in molecular library optimization. J. Phys. Chem. A 107, 2066–2074 (2003)

[56] Shi, S., Rabitz, H.: Quantum mechanical optimal control of physical observables in microsystems. The Journal of Chemical Physics 92(1), 364–376 (1990)

[57] Sohn, K.S., Zeon, I.W., Chang, H., Seung, K.L., Park, H.D.: Combinatorial search for new red phosphors of high efficiency at VUV excitation based on the YRO4 (R= As, Nb, P, V) system. Chem. Mater. 14, 2140–2148 (2002)

[58] Stillinger, F., Weber, T.: Packing structures and transitions in liquids and solids. Science 225, 983–989 (1984)

[59] Stoustrup, J., Schedletzky, O., Glaser, S.J., Griesinger, C., Nielsen, N.C., Sørensen, O.W.: Generalized bound on quantum dynamics: Efficiency of unitary transformations between non-hermitian states. Phys. Rev. Lett. 74, 2921–2924 (1995)

[60] Teotonio, H., Michael, R.R.: Reverse evolution. Evolution 55, 653–660 (2001)

[61] Valentine, F.A.: Convex Sets. McGraw-Hill, New York (1964)

[62] Weiner, A.M.: Femtosecond pulse shaping using spatial light modulators. Review of Scientific Instruments 71(5), 1929–1960 (2000)

[63] Weinreich, D., Watson, R., Chao, L.: Sign epistasis and genetic constraint on evolutionary trajectories. Evol. Int. J. Org. Evol. 59, 1165–1174 (2005)

[64] Whibley, A.C., Langlade, N.B., Andalo, C., Hanna, A.I., Bangham, A., Thebaud, C., Coen, E.: Evolutionary paths underlying flower color variation in Antirrhinum. Science 313, 963–966 (2006)

[65] Woods, R., Schneider, D., Winkworth, C., Riley, M., Lenski, R.: Tests of parallel molecular evolution in a long-term experiment with Escherichia coli. Proc. Natl. Acad. Sci. USA 103, 9107–9122 (2006)

[66] Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution, pp. 356–366 (1932)

[67] Wu, R., Pechen, A., Rabitz, H., Hsieh, M., Tsou, B.: Control landscapes of observable preparation with open quantum systems. J. Math. Phys. 49, 022108–1–12 (2008)

[68] Wu, R.B., Hsieh, M.A., Rabitz, H.: Role of controllability in optimizing quantum dynamics. Phys. Rev. A 83, 062306–1–6 (2011)

[69] Wu, R.B., Long, R., Dominy, J., Ho, T.S., Rabitz, H.: Singularities of quantum control landscapes. Phys. Rev. A 86, 013405–1–8 (2012)

[70] Zaugg, H.E.: Specific solvent effects in the alkylation of enolate anions. IV. kinetic order of solvent participation. J. Am. Chem. Soc. 83, 837–840 (1961)

Chapter 3
Biomolecular Information Gained through *In Vitro* Evolution on a Fitness Landscape in Sequence Space

Takuyo Aita and Yuzuru Husimi

Abstract. Biological evolution at the molecular level is conceptually regarded as the genetic information gaining process. Analyzing the *in vitro* evolution process, which is a simplified Darwinian evolution under a well-controlled environment, we can clarify the concept of the information gaining process. This evolution process can be modeled as a hill-climbing or adaptive walk on a fitness landscape in sequence space. Through the hill-climbing process, the evolving biopolymer (as the adaptive walker) stores the following two aspects of information: one stems from the sequences converged in sequence space and the other stems from the fitness increment on the fitness landscape. In Eigen's words, the former and latter are described as the "extent" and "content" of biological information, respectively [25]. In our approach, these two aspects can be interpreted based on the analogy between evolutionary dynamics and thermodynamics. Several studies introduced the concept of "free fitness" (which is analogous to free energy) as the Lyapunov function for evolution: *Free fitness* $\equiv$ *Fitness* $+$ *Temperature-like parameter* $\times$ *Entropy*. Furthermore, we focus on the novel quantity of *Fitness* divided by *Temperature-like parameter*, and regard this quantity as the content of information, while we regard *Entropy* as the extent of information. The quantity of *Free fitness* divided by *Temperature-like parameter* is a Lyapunov function of the evolution process, and then it should be called "biomolecular information", which includes both aspects of information.

Takuyo Aita
Graduate School of Science and Engineering, Saitama University, Saitama 338-8570, Japan
e-mail: taita@mail.saitama-u.ac.jp

Yuzuru Husimi
Innovative Research Organization, Saitama University, Saitama 338-8570, Japan
e-mail: husimi.@mail.saitama-u.ac.jp

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_3, © Springer-Verlag Berlin Heidelberg 2014

3.1 Introduction

3.1.1 Outline of In Vitro Molecular Evolution

In vitro molecular evolution is an artificial one, conducted in a laboratory and driven by the Darwinian mechanism. Study into this field began with Spiegelman's experiment in 1967 [39], and has been further developed [14, 27, 29, 40, 46, 52, 54]. It is the quest for the principle of emergence of functional biopolymers and is applied to industry and medicine, *e.g.* evolutionary (or adaptive) drug design, the first of which was approved by the Food and Drug Administration (USA). In 2004 [55] reviewed theoretical studies of *in vitro* evolution.

As an introductory section, we explain several important terms to comprehend *in vitro* evolution theoretically. A quantitative measure of a molecular phenotype, that is, a certain physicochemical property (such as enzymatic activity or affinity to a target receptor or replication rate constant) of an evolving molecule is designated as the "fitness". The conceptual space of all conceivable base- or amino acid-sequences (=genotypes) is designated as the "sequence space" (Maynard-Smith, 1970). Each of the conceivable sequences is mapped onto their corresponding points in the sequence space. A distance between two arbitrary points is measured with the "Hamming distance" between the two corresponding sequences. The scalar field constructed by mapping the fitness-value of each sequence into the corresponding point in the sequence space is designated as the "fitness landscape"[1], which is regarded as the evolutionary attribute of the biopolymer [22, 23, 33]. When the shape of the fitness landscape is fixed under a constant environment, an evolution process of biopolymers is considered as an "adaptive walk" or "hill-climbing" on the corresponding fitness landscape in sequence space.

In vitro evolution is classified into the following two types: "natural selection type" and "artificial selection type". In natural selection type, the fitness is the specific growth rate of evolving molecules, and the selection is driven by simultaneous evaluation and amplification by the molecules themselves. An example of this is self-replicating RNA molecules in a flow reactor. In artificial selection-type, the fitness is one of the physicochemical properties of evolving molecules (*e.g.* binding free energy to a target receptor), and the selection process is conducted through a cycle of evaluation and amplification by the experimenter. Therefore, *in vitro* evolution of the artificial selection-type is also called the "directed evolution". When we accentuate the aspect of molecular design, *in vitro* evolution of both types is also called "evolutionary molecular engineering" [24].

[1] As an alternative definition, the fitness landscape is defined in the space of allele frequency [20, 59]. The mean fitness-value of each allele frequency is mapped into the corresponding point in the space.

3.1.2 In Vitro Evolution as an Information Gaining Process

The *in vitro* evolution process have two parts: the "evolving system" that consists of phenotype/genotype molecules as evolving entities and the "surroundings" (or "environments") as an experimental setup. For example, we consider a typical case where one tries to create specific ligand molecules with high binding affinity to target-receptor molecules through *in vitro* evolution (Figure 3.1). The ligand molecules correspond to the evolving system, while the shape of the receptor molecular surface including distribution of the electric charge on it, and other experimental conditions of the solvent correspond to the surroundings. In this case, the fitness should be defined as the logarithm of the association constant between the ligand and the receptor, i.e. $\ln K_a$. Using the logarithm of the association constant means that the fitness is handled at the free-energy level. We must set and control the surroundings properly for correct evaluation of the fitness. The evolving biopolymers in Figure 3.1 gain a particular sequence of information and a high fitness value from the surroundings to bind with the target-receptor molecules under given conditions. Then, we can say that Darwinian evolution is considered as an information gaining process from the surroundings. In order to simplify the information gaining process and treat it physically, we focused on *in vitro* evolution in a well-controlled environment as an extremely simplified process of biological evolution [25].

In this simplified system, an evolution process of a biopolymer is considered as an "adaptive walk" or "hill-climbing" on the corresponding fitness landscape in the sequence space (Figure 3.2 right). Here, the evolving molecular system gains at least two kinds of information originating from the converged sequences and the fitness increment of the evolving biopolymer as the adaptive walker. These two represent two aspects of the biomolecular information, i.e. extent and content, respectively. The two are interconnected through physicochemical properties of the biopolymer. The interconnection may be analyzed based on the analogy between the evolution process and thermodynamics. Actually, the picture of the adaptive walk on a fitness landscape is analogous to the conceptual view that the molecular folding process is considered as a downhill walk on the energy landscape in conformation space (Figure 3.2) [58]. Therefore, there have been many studies on the analogy between evolution and thermodynamics.

Extending the interpretation of evolution by thermodynamics-like concepts, we can clarify a view of the evolution process as an information gaining process from the surroundings. Eigen raised a question about the "extent" and "content" of information in biological evolution [25]. According to him, the extent of information is related to the constrained volume of the sequence space and can be handled within the classical information theory [48]. This is comprehended within the framework of Shannon's information theory. On the other hand, the content of information means the meaning or semantic value of information, and is related to the concept of fitness. We must combine these two aspects of information in the same mathematical framework.

3.1.3 Basis of Thermodynamic Concepts

Here, we discuss the basis of thermodynamic concepts that are essentially related with the content in this chapter. First, the "universe" is divided into two parts: the "system" and the "surroundings". One focuses on the process including physical and chemical changes of the system. The second law of thermodynamics states that the system tends toward the minimal free-energy state at equilibrium. The Gibbs free energy of the system, G, is defined by $G \equiv H - TS$, where H, T and S represent the enthalpy, thermodynamic temperature and entropy of the system, respectively. The "chemical potential" G_{m} is equivalent to the molar Gibbs free energy. Under a constant temperature and constant external pressure, a spontaneous process of the system gives $\varDelta G = \varDelta H - T\varDelta S \leq 0$. The free energy is the Lyapunov function of the system. Furthermore, this inequality is rewritten as $-\varDelta G/T = -\varDelta H/T + \varDelta S \geq 0$, where $-\varDelta H/T$ represents the thermal entropy that the system emits to the surroundings and $-\varDelta G/T$ represents the change in total entropy of the universe. The above inequality represents that a spontaneous process is driven in the direction in which the total entropy of the universe increases. Thermodynamics is deeply related with classical information theory [17, 48]. Particularly, the concept of information is equivalent with negative entropy.

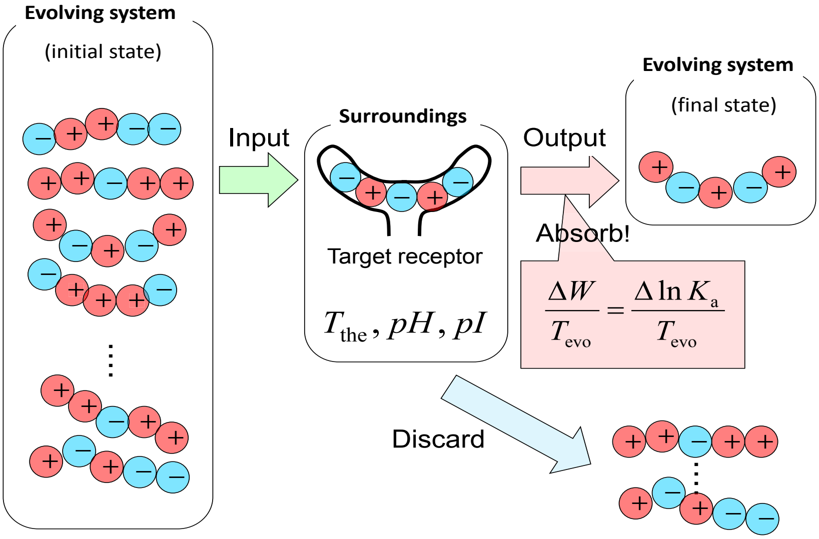
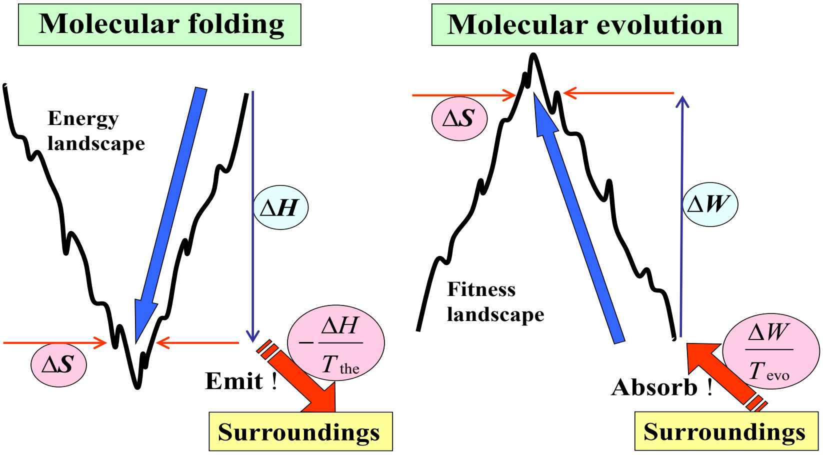

Evolving system
(initial state)
Input
Surroundings
Target receptor
T_{the}, pH, pI
Output
Evolving system
(final state)
Absorb!
$\frac{\Delta W}{T_{\mathrm{evo}}} = \frac{\Delta \ln K_{\mathrm{a}}}{T_{\mathrm{evo}}}$
Discard

Fig. 3.1 Emergence of a specific ligand sequence with high affinity to target-receptor molecules. Through mutation and selection, the specific ligand sequence "$+-+-+$" is found out from among all possible sequences to fit the surface of the target-receptor molecules "$-+-+-$". We interpret that the specific ligand sequence absorbs the fitness information $\varDelta W/T_{\mathrm{evo}}$ from the surroundings, where T_{evo} represents the evolutionary temperature, while T_{the} represents the thermodynamic temperature. Adapted from Aita and Husimi [7].

3.1.4 Outline of This Chapter

In this chapter, we summarize two typical studies on the evolutionary dynamics of *in vitro* evolution, which has been interpreted in terms of thermodynamics-like concepts leading to both aspects of the information concept, its extent and its content. One is a theory for a "natural selection-type model". That is the quasi-species theory developed by Eigen's group [22, 23, 25, 56]. The other is a theory of an "artificial selection-type model", which was developed by the authors [2, 3, 6, 7]. These studies gave a formulation for the information gaining process from the surroundings. Other studies are also reviewed within the framework of the two typical studies. In Section 3.2, we describe two different models of *in vitro* evolution: the natural selection-type model and the artificial selection-type model. In Section 3.3, the analogy between evolution and thermodynamics is described: first we review the previous studies related to this subject, next the results based on the two models. In Section 3.4, the analogy is developed to the concept of information gained through *in vitro* evolution.



$$-\frac{\Delta G}{T_{\text{the}}} = -\frac{\Delta H}{T_{\text{the}}} + \Delta S \geq 0 \qquad \frac{\Delta G}{T_{\text{evo}}} = \frac{\Delta W}{T_{\text{evo}}} + \Delta S \geq 0$$

Fig. 3.2 An analogy between molecular folding and molecular evolution. In molecular folding, a folding biopolymer (e.g. protein) descends the energy landscape by emitting the thermal entropy $-\Delta H/T_{\text{the}}$ $(= \Delta S_{\text{sur}})$ to the surroundings. In molecular evolution, we interpret that the evolving sequences climb the fitness landscape by absorbing "fitness information" $\Delta W/T_{\text{evo}}$ $(= \Delta I_{\text{fit}})$ as the negative entropy from the surroundings, which is defined in an experimental setup. Adapted from Aita and Husimi [4, 7].

3.2 Model of *In Vitro* Evolution

3.2.1 Natural Selection-Type Model

Eigen proposed the quasi-species model as the natural selection-type model. This model describes the evolutionary dynamics of the ensemble of simple self-replicators in a flow reactor, such as self-replicating RNA molecules [22, 23, 25]. The genome sequence of each species of the self-replicators consists of ν sites, with λ available symbols at each site. The number of all conceivable sequences is $n \equiv \lambda^{\nu}$. Each species (=sequence) is numbered by using a serial number s (or u) ($s, u = 1, 2, \cdots, n$). Then, each genome sequence is mapped onto the corresponding point in the λ-valued ν-dimensional sequence space. In the replication processes, it is possible to replace each symbol with one of the other $\lambda - 1$ symbols with a probability of μ, that is the mutation rate μ. The mole fraction of a certain species s at time t, denoted by $x_s(t)$, obeys the following differential equation:

$$\frac{\mathrm{d}x_s(t)}{\mathrm{d}t} = \sum_{u=1}^{n} m_{su} f_u x_u(t) - D(t) x_s(t) \qquad (s = 1, 2, \cdots, n) \tag{3.1}$$

$$m_{su} \equiv (1-\mu)^{\nu - d(s,u)} \left(\frac{\mu}{\lambda - 1}\right)^{d(s,u)}, \tag{3.2}$$

where f_s is the fitness of a species s, precisely $m_{ss} f_s$ is the replication rate constant of s (here we neglect decomposition) and m_{su} (for $s \neq u$) is a probability of mutation from a species u to a species s (note that $\sum_{s=1}^{n} m_{su} = 1$ for every u). The $d(s,u)$ is the Hamming distance between s and u. $D(t)$ is the dilution rate and works to satisfy $\sum_{s=1}^{n} x_s(t) = 1$ at every time t. Then, we can easily derive

$$D(t) = \sum_{s=1}^{n} f_s x_s(t) = \langle f \rangle, \tag{3.3}$$

where $\langle * \rangle$ represents the population average of a quantity $*$. It should be noted that the dilution rate $D(t)$ is equal to the population average of fitness, $\langle f \rangle$. Equations (3.1) and (3.2) mean that any species propagates cooperatively with neighbor species in the sequence space through mutation.

3.2.2 Artificial Selection-Type Model

3.2.2.1 Fitness Landscape

The evolutionary dynamics of a finite population is dependent on local structures of fitness landscapes. We supposed the following "NK landscape" [33] in the λ-valued ν-dimensional sequence space [3, 5][2]. In this model, an arbitrary site in a sequence interacts with other k sites. The fitness W for a given sequence "$\mathrm{A}_1\mathrm{A}_2 \cdots \mathrm{A}_\nu$" is defined by

[2] The validity of using the NK landscape was demonstrated in [5].

$$W = \sum_{j=1}^{\nu} w_j(\mathrm{A}_j|\mathrm{A}_{j_1}, \mathrm{A}_{j_2}, \cdots, \mathrm{A}_{j_k}), \tag{3.4}$$

where w_j is the "site-fitness," *i.e.*, a fitness contribution from the jth site, and A_j represents a particular symbol at the jth site. The value of w_j is given as a function of $1+k$ symbols at the jth site (A_j) and other k sites ($\mathrm{A}_{j_1}, \cdots, \mathrm{A}_{j_k}$). The interacting k sites $\{j_1, \cdots, j_k\}$ are randomly chosen from among all of the $\nu - 1$ sites except the jth site. Once a set of symbols $\{\mathrm{A}_{j_1}, \cdots, \mathrm{A}_{j_k}\}$ at these k sites is given, the value of w_j for an arbitrary symbol A_j, $w_j(A_j|\mathrm{A}_{j_1}, \cdots, \mathrm{A}_{j_k})$, is assigned randomly from a given probability distribution. Here, we adopt a discrete uniform distribution in the range $[-\varepsilon, \varepsilon]$, where ε is a positive constant ($\varepsilon > 0$). Let σ^2 be the variance of the uniform distribution, then $\sigma^2 = \varepsilon^2/3$ for this case. On the whole, the ruggedness of the landscape is controlled by the parameter k. In the case of $k = 0$, the resulting fitness landscape has a smooth surface and a single global peak. As the k-value increases, the surface of the fitness landscape becomes more rugged and many local optima appear. We denote the fitness coordinate by W. The frequency distribution of fitness over all conceivable sequences (of λ^ν) is given approximately by the following normal distribution:

$$\Omega(W) = \frac{\lambda^\nu}{\sqrt{2\pi\mathscr{V}}} \exp\left(-\frac{W^2}{2\mathscr{V}}\right) \quad \text{for } -\varepsilon\nu \leqq W \leqq \varepsilon\nu, \tag{3.5}$$

where $\mathscr{V}$ is the variance of fitness over all conceivable sequences in the sequence space and given by $\mathscr{V} = \sigma^2 \nu$. The average of fitness over the whole sequence space corresponds to the "foot" of the landscape, while regions where $W < 0$ correspond to below sea level and are negligible for the adaptive walks that start from random points, which are likely to be located at the foot of the landscape. Since the fitness at the global peak takes about $\varepsilon\nu$, the $\varepsilon\nu$ corresponds to the height of the landscape from the foot to the global peak. In this chapter, we focus on the regions from the foot of the landscape to the global peak: $0 \leqq W \leqq \varepsilon\nu$.

3.2.2.2 Protocol of Adaptive Walk

In this model, the *in vitro* evolution process is modeled as an "adaptive walk" or "hill-climbing" process. Suppose a single parent sequence as the walker on the fitness landscape, where the walker represents the evolving sequence or evolving phenotype/genotype molecule. The adaptive walk process is conducted by repetition of the following mutagenesis and selection processes. Furthermore, two different selection protocols are considered in parallel: one is the "sampling-screening" [2, 3, 7] and the other is the "selective enrichment" [8].

Case 1. Sampling-screening: First, a single parent sequence produces N mutant sequences as offspring. In the reproduction process, d-fold point mutations occur in each sequence randomly, that is, d (=the number of mutated sites) represents the Hamming distance between a parent and each of its offspring. Next, the fitness value of each mutant is evaluated, and then the best mutant with the highest fitness among the all become a new parent in the next generation. N is the library size of mutants to be screened in a single generation. The d is interpreted as the step length of the walker in sequence space.

Case 2. Selective enrichment: First, a single parent sequence produces all possible d-fold point mutants. The number of all possible d-fold point mutants derived from a single sequence is

$$N_d \equiv \binom{\nu}{d}(\lambda - 1)^d. \tag{3.6}$$

Each of all the N_d mutants has the same occurrence frequency. Here, we assume the population size is large enough to consider the population dynamics deterministically. The resulting library is called the primary library. Subsequently, each mutant in the primary library is amplified deterministically. The occurrence frequency of each mutant with fitness W is multiplied by the

$$\text{amplification factor} = \exp\left(\frac{rW}{k_{\text{B}}T_{\text{the}}}\right), \tag{3.7}$$

where k_{B} is the Boltzmann constant, T_{the} represents the thermodynamic temperature ($T_{\text{the}} = 270 \sim 370[K]$) and r is the number of rounds of the selective enrichment ($r = 1 \sim 6$). The *amplification factor* (Equation (3.7)) is based on an affinity selection system such as a ligand-receptor binding system[3]. In this system, W and $\exp(W/k_{\text{B}}T_{\text{the}})$ correspond to the negative binding-free energy and association constant, respectively. Equation (3.7) (for $r = 1$) is also interpreted as the Malthusian fitness in biology.

Subsequently, an arbitrary mutant is randomly chosen from the resulting population and it becomes a new parent in the next generation.

The controllable parameters N, d, T_{the} and r are constant throughout the adaptive walk. We assume $d(1+k) \ll \nu$, where $d(1+k)$ is the mean number of sites affected by random d-fold point mutations. Based on the NK landscape model, after the mutations, the $d(1+k)$ sites update their site-fitness values.

3.3 Analogy between Evolution and Thermodynamics

3.3.1 Overview

Thermodynamic concepts connect the evolution process to information. Here, we review studies related to the analogy between evolution and thermodynamics from various viewpoints.

[3] Here, we neglect the effects of the non-specific binding and the washing process for simplicity.

Some of the previous studies introduced the concept of "free fitness" that is analogous to free energy, and then proposed that evolution is driven in the direction in which free fitness increases. In their common scheme, free fitness is defined as

$$\textit{Free fitness} \equiv \textit{Fitness} + \textit{Temperature-like parameter} \times \textit{Entropy}, \qquad (3.8)$$

where *Fitness* is sometimes the population average of log fitness or selection coefficient. The *Temperature-like parameter* is dependent on the models. For example, concerning the natural evolution with a weak mutational effect, the theories by Iwasa [32], Berg and Lässig [13] and Sella and Hirsh [47] were based on the Markovian process and detailed balance for transition probability, which means a probability of transition from a homogeneous population occupied with a certain species to that occupied with another species in sufficient generation time. In their scheme, the *Temperature-like parameter* corresponds to the inverse of the double population size, and the *Entropy* is the ensemble average of the logarithm of the occurrence frequency of each species. The special case of and infinite population corresponds to Fisher's fundamental theorem of natural selection [28]. Recently, Barton's group [11, 12] extended the concept of free fitness to incorporate mutational effects explicitly, based on the stationary distribution of allele frequencies [59]. They reviewed the studies on the relationship between statistical physics and evolutionary biology [12]. Demetrius [18, 19] discussed the evolutionary dynamics of an organismic population in the "age space" in terms of the age-specific fecundity and mortality. In his scheme, the *Temperature-like parameter* corresponds to the inverse of the generation time, while the entropy is defined as a measure of the uncertainty in the age of the immediate ancestor of a randomly chosen newborn. Ao presented the relationship between Darwinian evolution and thermodynamics from the viewpoint of Langevin dynamics [9]. His theory describes the dynamics on potential surface in "genotype frequency space", where each coordinate axis represents the frequency of a genotype. The concept of the free fitness is a key role in this chapter and will be again referred to in the natural selection-type model and the artificial selection-type model.

Blackburne and Hirst conducted a simulation of population dynamics using simple lattice model proteins [15]. They also estimated the population using the analogy with Boltzmann distribution in thermodynamics, in which a temperature-like parameter was empirically derived as a function of the mutation rate and selection pressure.

Sato *et al.* referred to a mathematical relationship between fluctuation $V[X]_a$ and response $\langle X\rangle_{a+\Delta a} - \langle X\rangle_a$ in a biological system (where $\langle X\rangle_a$ and $V[X]_a$ are the average and variance of the variable X at the initial parameter value a), and demonstrated that the relationship they found, $\langle X\rangle_{a+\Delta a} - \langle X\rangle_a \propto V[X]_a \times \Delta a$, is similar to Einstein's relation in the fluctuation-dissipation theorem in Brownian motion [43]. They confirmed the relationship through an experimental observation, in which X represents the logarithm of fluorescence intensity per *E. coli.* cell including mutant GFP proteins and Δa represents the synonymous mutation rate of their genes.

From the viewpoint of molecular imprinting, Pande *et al.* developed statistical mechanics of protein folding and design [42], introducing the "design temperature" T_{des}, which controls the probability of the occurrence of amino acid sequences with low energy (designed sequences) in a given canonical ensemble. They obtained a phase diagram for model heteropolymers in a two dimensional T_{des}-T_{the} (T_{the} is thermodynamic temperature) space.

3.3.2 Quasi-species

We summarized the result of the mathematical analysis of the natural selection-type model (Equation (3.1)). In a special case of no mutations ($\mu = 0$), Equation (3.1) is simplified as follows:

$$\frac{\mathrm{d}x_s(t)}{\mathrm{d}t} = (f_s - D(t))x_s(t) \qquad (s = 1,2,\cdots,n), \tag{3.9}$$

with $\sum_{s=1}^{n} x_s(t) = 1$. The solution of Equation (3.9) is obtained as follows:

$$x_s(t) = \frac{x_s(0)\exp(f_s t)}{\sum_{u=1}^{n} x_u(0)\exp(f_u t)} \qquad (s = 1,2,\cdots,n). \tag{3.10}$$

Let s^* be the fittest species that has the maximal fitness value among all n species. In the stationary state, we observe

$$\lim_{t\to\infty} x_s(t) = \begin{cases} 1, \text{ if } & s = s^* \\ 0, \text{ if } & s \neq s^* \end{cases} \tag{3.11}$$

That is, only the fittest species s^* exists in the reactor. Furthermore, according to Equation (3.9), the first derivative of Equation (3.3) is given by

$$\frac{\mathrm{d}D(t)}{\mathrm{d}t} = \langle f^2 \rangle - \langle f \rangle^2 \geq 0, \tag{3.12}$$

where $\langle f^2 \rangle - \langle f \rangle^2$ represents the population variance of fitness. This means that the dilution rate $D(t)(= \langle f \rangle)$ is the Lyapunov function of the process. In the stationary state,

$$\lim_{t\to\infty} D(t) = \lim_{t\to\infty} \langle f \rangle = f_{s^*} \equiv \max\{f_s\}. \tag{3.13}$$

This situation is the so called "survival of the fittest". These results correspond to Fisher's fundamental theorem of natural selection [28].

In general cases of non-zero mutation rates ($\mu > 0$), Equation (3.1) must be transformed in the following manner. Considering the $n \times n$ square matrix $[m_{su} f_u]$ (s and u represent rows and columns, respectively), we denote the qth eigenvalue and eigenvector of this matrix by Λ_q and $\mathbf{r}_q \equiv [r_{q1}, r_{q2}, \cdots, r_{qn}]^\top$ (for $q = 1,2,\cdots,n$), respectively, where $^\top$ is the transpose operation. Here, the eigenvector $\mathbf{r}_q$ should be determined to satisfy $\sum_{s=1}^{n} r_{qs} = 1$. By diagonalizing the matrix $[m_{su} f_u]$ by the $n \times n$ square matrix $R \equiv [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_n]$ and its inverse matrix R^{-1}, and introducing $[y_1(t), y_2(t), \cdots, y_n(t)]^\top \equiv R^{-1}[x_1(t), x_2(t), \cdots, x_n(t)]^\top$, Equation (3.1) is transformed to

$$\frac{\mathrm{d}y_q(t)}{\mathrm{d}t} = (\Lambda_q - D(t))\, y_q(t) \qquad (q = 1, 2, \cdots, n), \tag{3.14}$$

with $\sum_{q=1}^n y_q(t) = 1$, where q is designated as the "quasi-species". It is remarkable that Equation (3.14) is the same mathematical form as Equation (3.9). Therefore, the solution of Equation (3.14) is the same mathematical form as Equation (3.10):

$$y_q(t) = \frac{y_q(0)\exp(\Lambda_q t)}{\sum_{v=1}^n y_v(0)\exp(\Lambda_v t)} \qquad (q = 1, 2, \cdots, n). \tag{3.15}$$

It is noted that $x_s(t) \geq 0$ in Equation (3.10), while $y_q(t) \lesseqqgtr 0$ in Equation (3.15). Let q^* be the quasi-species that has the maximal eigenvalue (according to Perron-Frobenius theorem, $\Lambda_{q^*} > 0$ and $r_{q^*s} > 0$ for every s). In the stationary state, we can observe

$$\lim_{t\to\infty} y_q(t) = \begin{cases} 1, \text{ if } & q = q^* \\ 0, \text{ if } & q \neq q^* \end{cases} \tag{3.16}$$

That is, only the quasi-species q^* is realized in the reactor. The $x_s(t)$ in the stationary state is given by

$$\lim_{t\to\infty} x_s(t) = \lim_{t\to\infty} \sum_{q=1}^n r_{qs}\, y_q(t) = r_{q^*s}. \tag{3.17}$$

The realized eigenvector $\mathbf{r}_{q^*}$ is designated as the "quasi-species distribution", which is caused by the mutation-selection balance. Furthermore, the dilution rate is rewritten by $D(t) = \sum_{q=1}^n \Lambda_q y_q(t) = \langle \Lambda \rangle_y$, where $\langle * \rangle_y \equiv \sum_{q=1}^n * \times y_q(t)$. According to Equation (3.14), the first derivative of $D(t)$ is given by $\mathrm{d}D(t)/\mathrm{d}t = \langle \Lambda^2 \rangle_y - \langle \Lambda \rangle_y^2 \lesseqqgtr 0$. This means that the dilution rate $D(t)(= \langle f \rangle)$ is not necessarily the Lyapunov function for the quasi-species model. In the stationary state,

$$\lim_{t\to\infty} D(t) = \lim_{t\to\infty} \langle f \rangle = \Lambda_{q^*} \equiv \max\{\Lambda_q\}. \tag{3.18}$$

That is, the population average of fitness, $\langle f \rangle$, becomes equal to the maximal eigenvalue Λ_{q^*}.

It is important to note that the quasi-species distribution is strongly dependent on the shape of the fitness landscape ($\{f_s\}$) and mutation rate (μ). For proper landscapes, the quasi-species distribution shows a phase transition at several critical points of the mutation rate. Let m be the species as the master sequence and let $\langle f_s \rangle_{s\neq\mathrm{m}}$ be the mean fitness over all the species except m. When the mutation rate μ exceeds a certain critical point called the "error threshold",

$$\mu_{\mathrm{cr}} = \frac{\ln f_{\mathrm{m}} - \ln\langle f_s \rangle_{s\neq\mathrm{m}}}{v} \quad (\text{for } \lambda = 2), \tag{3.19}$$

an "error catastrophe" or a localization-delocalization transition occurs. In the delocalization state, all species have the identical mole fraction.

An example of a fitness landscape demonstrating a sharp localization-localization transition is an asymmetric twin peaks consisting of a sharp high peak and a broad low peak. When μ is very small, the quasi-species members localize at the high peak. When μ becomes greater than some critical value (and less than the error threshold), the quasi-species members become to localize at the broad low peak, because the population at the broad peak is mutationally robust based on mutational interconnectedness. The transition is very narrow for μ and shows a critical slowing down phenomenon [30, 45]. Wilke *et al* called this situation the "survival of the flattest" [57].

These results can be interpreted by thermodynamics-like concepts. Mutation causes the species to diffuse in the sequence space, while selection causes them to converge on the local area. Therefore, the mutation rate μ corresponds to a temperature-like parameter T. When $\mu = 0$, the fittest s^* that has the maximal fitness f_{s^*} is realized. This is analogous to the case of $T = 0$ in thermodynamics, because the thermodynamic system realizes the minimal energy state. On the other hand, when $\mu > 0$, the quasi-species q^* that has the maximal eigenvalue Λ_{q^*} is realized. This is analogous to the case of the thermodynamic system realizing the minimal free energy state. In the above mentioned asymmetric twin peak case, the localization at the broad low peak is an analogous state to an intermediate conformation X of a protein in the unfolding process (Native $\leftrightarrow$ X $\leftrightarrow$ Denatured). Therefore, the eigenvalue Λ_q could be called the "free fitness" [30].

In thermodynamics, the phase transition temperature between state A and state B is given by $T_{\mathrm{cr}} = \Delta H/\Delta S$, where ΔH and ΔS are the enthalpy change and entropy change between A and B, respectively. In Equation (3.19), the numerator means energy- or enthalpy-like quantity, and the denominator means entropy-like quantity ($\nu = \log_2 2^\nu - \log_2 1$). Therefore, Equation (3.19) is analogous to $T_{\mathrm{cr}} = \Delta H/\Delta S$.

3.3.3 Attractor of Fitness in the Artificial Selection-Type Model

We summarized the result of the statistical analysis of the artificial selection-type model. Denoting the fitness of the walker (=parent sequence) by W, we focus on the statistical properties of a time course of W through the hill-climbing process. Consider that the hill-climbing starts from the foot of the landscape ($W = 0$). The walker's fitness W increases exponentially and tends toward a stationary value denoted by W^*, which is called the attractor. In the stationary state, the value of W fluctuates around the attractor W^*. As a result, under extreme conditions where λ, ν, d, N and $k_{\mathrm{B}}T_{\mathrm{the}}$ have large values[4], the attractor W^* is given by:

[4] The conditions for Equation (3.20) are $\ln N \ll d(1+k) \ll \nu$ and $1 \ll N \ll \binom{\nu}{d}(\lambda-1)^d$ (where $\binom{\nu}{d}(\lambda-1)^d$ is the size of the "d-boundary" of any sequence). The conditions for Equation (3.21) is $k_{\mathrm{B}}T_{\mathrm{the}}/r \gg \sigma\sqrt{8/3}$. The derivations of Equation (3.20) and Equation (3.21) are described in [5, 8].

$$W^* = 2\sqrt{\frac{\mathscr{V}\nu \ln N}{d(1+k)}} \quad \text{for sampling-screening} \tag{3.20}$$

$$W^* = \frac{\mathscr{V}}{k_{\mathrm{B}}(T_{\mathrm{the}}/2r)} \quad \text{for selective enrichment.} \tag{3.21}$$

$\mathscr{V}$ represents the variance of fitness over all conceivable sequences in the sequence space (Equation (3.5)). This stationary state is caused by the balance of the effects of mutation, sampling and selection.

Let J be the expectation of the change in walker's fitness W after a single step of walk. For both cases, J (= the climbing rate) is given by

$$J \approx -\frac{d(1+k)}{\nu}(W - W^*). \tag{3.22}$$

These results are interpreted in the analogy with thermodynamics.

3.3.4 Evolutionary Boltzmann Constant and Evolutionary Temperature

In order to interpret the evolutionary dynamics for the artificial selection-type model, we introduce the following thermodynamics-like functions. First, we introduce the "evolutionary Boltzmann constant," k_{E}, which is analogous to Boltzmann constant, k_{B}. Next, based on the global fitness distribution given in Equation (3.5), we define the "entropy" S as a function of W as follows:

$$S(W) \equiv k_{\mathrm{E}} \times \ln \Omega(W), \tag{3.23}$$

$$= -\frac{k_{\mathrm{E}}W^2}{2\mathscr{V}} + const. \tag{3.24}$$

Following the definition of thermodynamic temperature, we define "evolutionary temperature" T_{evo} as follows:

$$T_{\mathrm{evo}} \equiv -\left(\left.\frac{\mathrm{d}S}{\mathrm{d}W}\right|_{W=W^*}\right)^{-1} = \frac{\mathscr{V}}{k_{\mathrm{E}}W^*}, \tag{3.25}$$

where W^* is the attractor we defined in the previous section. Then, W^* is also given by

$$W^* = \frac{\mathscr{V}}{k_{\mathrm{E}}T_{\mathrm{evo}}}. \tag{3.26}$$

By comparing Equation (3.20) with Equation (3.26) and comparing Equation (3.21) with Equation (3.26), we obtain

$$k_E T_{evo} = k_L \times \sqrt{\frac{d}{4 \ln N}} \quad \text{for sampling-screening} \tag{3.27}$$

$$k_E T_{evo} = k_B \times \frac{T_{the}}{2r} \quad \text{for selective enrichment,} \tag{3.28}$$

where k_L is defined as

$$k_L \equiv \sigma \sqrt{1+k} \tag{3.29}$$

and designated as the "landscape constant". The meaning of k_L is the standard deviation of a change in fitness for a unit Hamming distance. Then, k_L indicates the degree of ruggedness of the landscape.

It should be noted that, for each of the above equations, dividing the right-hand side into k_E and T_{evo} is not so important in the essence of this scheme and should be done arbitrarily. However, it is clear that the former parts consist of the natural constants (k_L and k_B) inherent in the evolving system (including the fitness landscape), while the latter parts consist of the controllable parameters (d, N, T_{the} and r) in the surroundings. Therefore, we can divide as follows:

$$k_E \equiv k_L, \quad T_{evo} \equiv \sqrt{\frac{d}{4 \ln N}} \quad \text{for sampling-screening} \tag{3.30}$$

$$k_E \equiv k_B, \quad T_{evo} \equiv \frac{T_{the}}{2r} \quad \text{for selective enrichment.} \tag{3.31}$$

In Equation (3.30), the d indicates the degree of diffusion in the sequence space by random mutation, while $\ln N$ indicates the degree of convergence of sequence diversity by selection. T_{evo} is the ratio of these conflicting effects.

According to thermodynamics of protein folding, the most probable energy of a canonical ensemble is given by an equation similar to Equation (3.26) (e.g. Equation (11) in Wolynes and Luthey-Schulten [58]). Therefore, we can say that k_E and T_{evo} are analogous to the Boltzmann constant and thermodynamic temperature of the thermal bath, respectively. Beyond the analogy, Equation (3.31) connects the evolutionary dynamics and thermodynamics in the simple form [42].

3.3.5 Evolutionary Potential, Free Fitness and Evolutionary Force

We define the "evolutionary potential," ϕ, which is analogous to the chemical potential, as follows:

$$\phi(W) \equiv W + T_{evo} \times S(W). \tag{3.32}$$

When the number of walkers (=parent sequences) is M, we define the "free fitness G"[5] by $G \equiv \phi \times M$ [7]. By substituting Equation (3.24) into Equation (3.32),

[5] In this scheme, the evolutionary potential ϕ plays a more significant and essential role than the free fitness.

ϕ is rewritten as the following convex function which takes the maximal values at $W = W^*$:

$$\phi(W) = -\frac{k_{\mathrm{E}}T_{\mathrm{evo}}}{2\mathscr{V}}\left(W - \frac{\mathscr{V}}{k_{\mathrm{E}}T_{\mathrm{evo}}}\right)^2 + const \tag{3.33}$$

$$= -\frac{k_{\mathrm{E}}T_{\mathrm{evo}}}{2\mathscr{V}}(W - W^*)^2 + const, \tag{3.34}$$

(by using Equation (3.26)). Furthermore, the evolutionary potential ϕ and free fitness G is the Lyapunov functions of the evolution process (proof is shown in the next section). Namely, the *in vitro* evolution is driven in the direction in which ϕ and G increase. In addition to these quantities, we define the "evolutionary force," X, which is analogous to the generalized force, by

$$X(W) \equiv \frac{\mathrm{d}}{\mathrm{d}W}\frac{\phi(W)}{T_{\mathrm{evo}}} \tag{3.35}$$

$$= -\frac{k_{\mathrm{E}}}{\mathscr{V}}(W - W^*). \tag{3.36}$$

Substituting Equation (3.32) into Equation (3.35), the evolutionary force X is decomposed into the "fitness force" X_{fit} and the "entropy force" X_{ent}: $X = X_{\mathrm{fit}} + X_{\mathrm{ent}}$, where

$$X_{\mathrm{fit}} \equiv \frac{1}{T_{\mathrm{evo}}} \quad \text{and} \quad X_{\mathrm{ent}} \equiv \frac{\mathrm{d}S(W)}{\mathrm{d}W} = -\frac{k_{\mathrm{E}}}{\mathscr{V}}W. \tag{3.37}$$

X_{fit} is caused by a selection event and pushes the walkers upward, while X_{ent} is caused by a mutation event and pushes the walkers downward. The mutation-selection-random drift balance occurs when X_{fit} and X_{ent} cancel each other out.

Based on the definition we mentioned above, we can describe the dynamics of the adaptive walk as follows (see Figure 3.3). Driven by the evolutionary force X, the walkers tend to achieve the stationary state with the maximal evolutionary potential ϕ, in which the fitness force X_{fit} and the entropy force X_{ent} cancel each other out. The evolutionary force X depends strongly on the evolutionary temperature T_{evo}. Here, consider the walkers are located at the middle point on the landscape. If $T_{\mathrm{evo}} = \infty$, then W^* lies at the foot of the landscape and a negative force ($X < 0$) acts on the walkers so the walkers are pushed downward. In this case, the maximal entropy state is realized. As T_{evo} becomes lower, the W^* becomes higher up near the top of the landscape and a positive force ($X > 0$) acts on the walkers so the walkers are pushed upward. In this case, the (nearly) maximal fitness state is realized. If $T_{\mathrm{evo}} = 0$, the walkers cannot move in the sequence space, and evolution does not occur.

3.3.6 Fitness Flux and Einstein's Relation-Like Formula

We consider the hill-climbing rate for a single step of walk. Let ΔW be the change in walker's fitness W after a single step of walk (top of Figure 3.4). The expectation of ΔW is denoted by J ($\equiv \mathbf{E}[\Delta W]$). For both of the two cases (sampling-screening

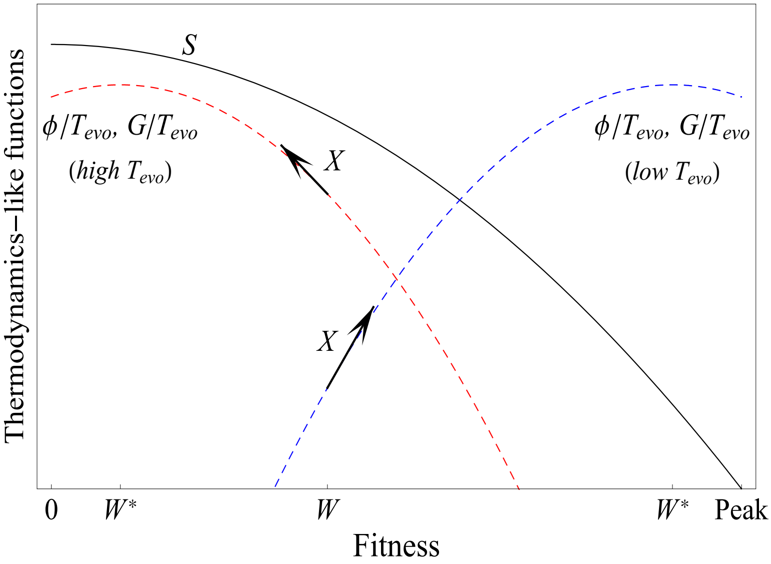


Fig. 3.3 Interpretation of evolutionary dynamics by thermodynamics-like concepts. The solid line represents the entropy S, while the dashed lines represent ϕ/T_{evo} or G/T_{evo}, where ϕ and G are the evolutionary potential and free fitness, respectively. The right blue (left red) dashed line is for the case of a low (high) evolutionary temperature T_{evo}. W^* represents the stationary point, in which ϕ and G take the maximum values under the given T_{evo}. The black arrow represents the evolutionary force X that acts on the walker with the current fitness W. Details are described in the text. Adapted from Aita and Husimi [4, 7].

and selective enrichment), J is given by Equation (3.22). By using Equation (3.36), Equation (3.22) is rewritten as

$$J \approx LX, \qquad L = \frac{D}{k_{\mathrm{E}}}, \tag{3.38}$$

where L is the "linear transport coefficient" and $D \equiv k_{\mathrm{L}}^2 d$ is the "diffusion coefficient" of the walker's fitness W along the fitness coordinate when the walker performs a random walk in the sequence space[6]. The random walk occurs when $N = 1$ because of no selection pressure ($X_{\text{fit}} = 0$). Equation (3.38) is analogous to Einstein's relation in Brownian motion [26].

Sato *et al.* found a similar scheme to Equation (3.38) for the relationship between fluctuation and response in a biological system [43]. Iguchi has tried to extend Equation (3.38) to a case of coevolving biopolymers in a similar framework of

[6] Consider that all possible d-fold point mutants are generated from a parent sequence. The variance of the probability density of the fitnesses over all the mutants is approximately given by $k_{\mathrm{L}}^2 d$ ($\equiv D$).

the Onsager's reciprocal relations [31]. The linear relationship shown in Equation (3.38) is similar to Fisher's fundamental theorem of natural selection [28], although the theorem holds in cases of no mutational events. Yang *et al.* [60] experimentally tested the theorem through *in vitro* evolution of Mnt repressor-binding DNA, and obtained the correlation between evolution speed and sequence diversity.

3.4 Information Gained through *In Vitro* Evolution

3.4.1 Fitness Fluctuation and Energy Fluctuation

First, we discuss sampling-screening. Since exploration on a fitness landscape is done by random sampling of finite mutants from among the underlying mutant population, the fitness change ΔW after a single step of walk is a stochastic quantity and its probability density is described by the theory of extreme value statistics. Let Σ_{fit} be the standard deviation of ΔW ($\Sigma_{\mathrm{fit}} \equiv \mathbf{SD}[\Delta W]$). $k_{\mathrm{E}}T_{\mathrm{evo}}$ is related with Σ_{fit} by

$$k_{\mathrm{E}}T_{\mathrm{evo}} \approx \frac{\Sigma_{\mathrm{fit}}}{2} \quad \text{for sampling-screening.} \tag{3.39}$$

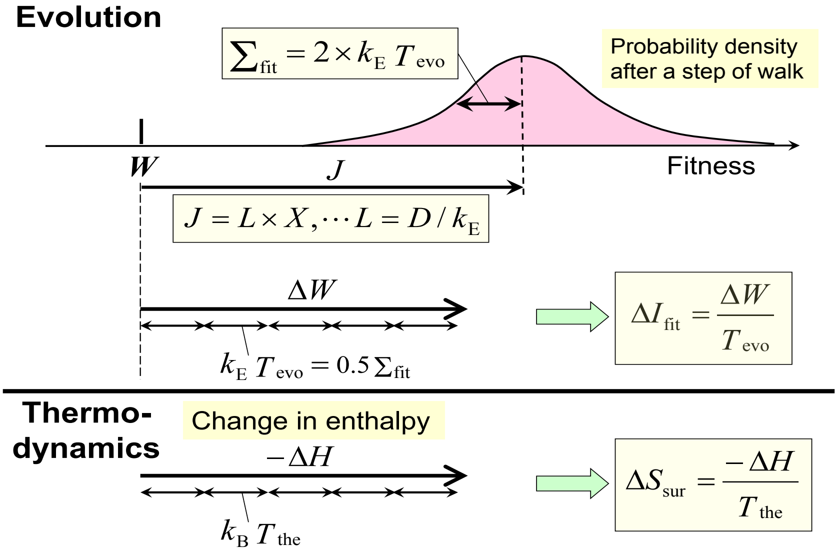

Evolution
$\sum_{\mathrm{fit}} = 2 \times k_{\mathrm{E}}\, T_{\mathrm{evo}}$
Probability density after a step of walk
W
J
Fitness
$J = L \times X, \cdots L = D / k_{\mathrm{E}}$
ΔW
$k_{\mathrm{E}}\, T_{\mathrm{evo}} = 0.5 \sum_{\mathrm{fit}}$
$\Delta I_{\mathrm{fit}} = \frac{\Delta W}{T_{\mathrm{evo}}}$
Thermo-dynamics
Change in enthalpy
$-\Delta H$
$k_{\mathrm{B}}\, T_{\mathrm{the}}$
$\Delta S_{\mathrm{sur}} = \frac{-\Delta H}{T_{\mathrm{the}}}$

Fig. 3.4 (Upper) A probability density of the change in walker's fitness after a single step of walk from a certain fitness W, as for the case of the sampling-screening. J and Σ_{fit} represent the expectation and standard deviation of the change, respectively. (Middle) The fitness-information change, ΔI_{fit}, is the fitness-change digitized by the fluctuation size, $k_{\mathrm{E}}T_{\mathrm{evo}} = \Sigma_{\mathrm{fit}}/2$. (Bottom) The concept of thermal entropy, $\Delta S_{\mathrm{sur}} = -\Delta H/T_{\mathrm{evo}}$, which is the enthalpy-change digitized by the fluctuation size of thermal energy per degree of freedom, $k_{\mathrm{B}}T_{\mathrm{the}}$. Adapted from Aita and Husimi [4, 7].

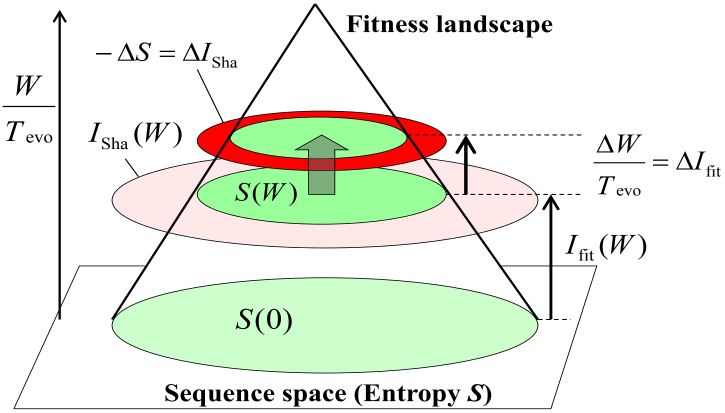

Fitness landscape
$-\Delta S = \Delta I_{\text{Sha}}$
$\frac{W}{T_{\text{evo}}}$
$I_{\text{Sha}}(W)$
$S(W)$
$\frac{\Delta W}{T_{\text{evo}}} = \Delta I_{\text{fit}}$
$I_{\text{fit}}(W)$
$S(0)$
Sequence space (Entropy S)

Fig. 3.5 Schematic representation of the fitness information I_{fit} and Shannon information I_{Sha}. The circular cone represents a fitness landscape schematically. The height represents the fitness W divided by T_{evo}, while the area of the cross section represents the entropy $S(W) \equiv k_{\text{E}} \times \ln \Omega(W)$, where $\Omega(W)$ is the number of all sequences with a given fitness W. $I_{\text{fit}}(W)$ is defined as the change of W/T_{evo} from the foot of the landscape, while $I_{\text{Sha}}(W)$ is defined as the negative change of S from the foot. ΔI_{fit} and ΔI_{Sha} are the changes of these quantities after a single step of walk. Adapted from Aita and Husimi [4, 7].

This equation shows that $k_{\text{E}}T_{\text{evo}}$ is approximately equivalent to the "fitness fluctuation" Σ_{fit}.

Next, we discuss selective enrichment. We introduce the molar energy E_{m} of an ensemble of the evolving molecules of Avogadro's number at a thermodynamic temperature T_{the}. Suppose that all the molecules realize a canonical ensemble, in which $k_{\text{B}}T_{\text{the}}$ is related to the "energy fluctuation". According to thermo-statistical mechanics theory, the standard deviation of E_{m}, denoted by Σ_{ene} ($\equiv \mathbf{SD}[E_{\text{m}}]$), is given by the energy-fluctuation formula [7]. Using this formula combined with Equation (3.28), we obtain

$$k_{\text{E}}T_{\text{evo}} \approx \frac{\Sigma_{\text{ene}}}{r\sqrt{2f}} \quad \text{for selective enrichment,} \tag{3.40}$$

where f is the degree of freedom of the molecules of Avogadro's number.

Then we can say that the fitness fluctuation (Σ_{fit}) originates from the events that the evolving sequences (genotypes) move in sequence space, while the energy fluctuation (Σ_{ene}) originates from the events that the evolving phenotypic molecules move in physical space.

[7] $\Sigma_{\text{ene}} = k_{\text{B}}T_{\text{the}} \times \sqrt{f/2}$.

3.4.2 Biomolecular Information

Suppose an adaptive walk from the foot of the landscape ($W = 0$) up to the stationary state ($W = W^*$). Here, using entropy S in Equation (3.23) and evolutionary potential ϕ in Equation (3.32), we introduce the following three quantities as functions of walker's fitness W:

$$I_{\mathrm{fit}}(W) \equiv \frac{W}{T_{\mathrm{evo}}} \tag{3.41}$$

$$I_{\mathrm{Sha}}(W) \equiv S(0) - S(W) \tag{3.42}$$

$$I_{\mathrm{bio}}(W) \equiv \frac{\phi(W)}{T_{\mathrm{evo}}} - \frac{\phi(0)}{T_{\mathrm{evo}}} \tag{3.43}$$

$$= I_{\mathrm{fit}}(W) - I_{\mathrm{Sha}}(W), \tag{3.44}$$

and let ΔI_{fit}, ΔI_{Sha} and ΔI_{bio} be the changes in I_{fit}, I_{Sha} and I_{bio}, respectively, after a single step of a walk (Figure 3.5). Since Equation (3.42) means the change in entropy S between an initial state and a certain state with fitness W (Figure 3.5), I_{Sha} is interpreted as the "Shannon information" [48].

On the other hand, from Equation (3.39) and Equation (3.40), the meaning of $\Delta I_{\mathrm{fit}} (= \Delta W / T_{\mathrm{evo}})$ is as follows:

$$\frac{\Delta I_{\mathrm{fit}}}{k_{\mathrm{E}}} \approx \frac{\Delta W}{\Sigma_{\mathrm{fit}}} \times 2 \quad \text{for sampling-screening} \tag{3.45}$$

$$\frac{\Delta I_{\mathrm{fit}}}{k_{\mathrm{E}}} \approx \frac{\Delta W}{\Sigma_{\mathrm{ene}}} \times r\sqrt{2f} \quad \text{for selective enrichment.} \tag{3.46}$$

As can be seen in the above equations, ΔI_{fit} is the fitness-change digitized by the fitness fluctuation size Σ_{fit} or energy fluctuation size Σ_{ene} (in the middle of Figure 3.4). Here, the analog-to-digital conversion is realized as the significant figures of the fitness-change observed by the walker with the observation error Σ. This is analogous to the thermal entropy change in the surroundings, $-\Delta H / T_{\mathrm{the}}$, when a system emits the heat $-\Delta H$ to the surroundings (bottom of Figure 3.4) [10]. According to the analogy with thermodynamics, we can interpret $\Delta I_{\mathrm{fit}} (= \Delta W / T_{\mathrm{evo}})$ as the negative entropy that the evolving system absorbs from the surroundings (Figure 3.2 right). Here, the surroundings mean the experimental setup (e.g. a column of affinity-chromatography) around the biopolymer as an evolving entity (Figure 3.1). We designate I_{fit} as the "fitness information". We can say that the evolving entity gains the fitness information from the surroundings (Figure 3.1 and 3.2).

The expectation of ΔI_{bio} is given by

$$\mathbf{E}[\Delta I_{\mathrm{bio}}] = JX \approx LX^2 \geq 0. \tag{3.47}$$

Equation (3.47) proves the theorem that I_{bio} (or ϕ) is a Lyapunov function of the evolution process. Therefore, we conclude that the evolution is driven in the direction in which I_{bio} increases, and then we designate I_{bio} as the "biomolecular information" gained through *in vitro* evolution.

3.4.3 Information-Gain Formula

Based on the classical concept of information [17, 48], we consider the information gained through a single step of the selection process. We denote this information gain by $\varDelta I_{\mathrm{sel}}$. $\varDelta I_{\mathrm{sel}}$ is defined by the following change in classical entropy after a single step of walk:

$$\varDelta I_{\mathrm{sel}} \equiv k_{\mathrm{E}} \ln N - k_{\mathrm{E}} \ln 1 = k_{\mathrm{E}} \ln N \quad \text{for sampling-screening} \tag{3.48}$$

$$\varDelta I_{\mathrm{sel}} \equiv k_{\mathrm{E}} \sum_{s=1}^{N_d} -p_0(s) \ln p_0(s) - k_{\mathrm{E}} \sum_{s=1}^{N_d} -p_1(s) \ln p_1(s) \quad \text{for selective enrichment,} \tag{3.49}$$

where s ($s = 1 \sim N_d$) represents an arbitrary mutant among all conceivable d-fold point mutants of a parent sequence, and N_d is given by Equation (3.6). The $p_0(s)$ and $p_1(s)$ represent the occurrence probability of the sequence s in the mutant population before selective enrichment and that after selective enrichment, respectively, with

$$p_0(s) = \frac{1}{N_d} \quad \text{and} \quad p_1(s) = \frac{\exp(rW(s)/k_{\mathrm{B}} T_{\mathrm{the}})}{\sum_{s=1}^{N_d} \exp(rW(s)/k_{\mathrm{B}} T_{\mathrm{the}})}, \tag{3.50}$$

where $W(s)$ is the fitness of the sequence s.

Interestingly, for both cases, the I_{fit}, I_{Sha} and I_{bio} at the attractor $W = W^*$ are related with $\varDelta I_{\mathrm{sel}}$ by

$$I_{\mathrm{bio}}(W^*) = I_{\mathrm{Sha}}(W^*) = \frac{I_{\mathrm{fit}}(W^*)}{2} \tag{3.51}$$

$$= \frac{\mathscr{V}}{2k_{\mathrm{E}} T_{\mathrm{evo}}^2} \approx t^* \times \varDelta I_{\mathrm{sel}}, \tag{3.52}$$

where t^* represents the mean generation (or mean step number) from the foot of the landscape up to the stationary state. Precisely, the t^* is defined as a characteristic generation that the expectation of the walker's fitness becomes equal to $W^*(1 - e^{-2})$. The t^* is approximately given by $t^* \approx 2\nu/d(1+k)$. Equation (3.52) is reasonable in that I_{bio} in the stationary state is approximately equivalent to the sum of the information gained by the selection process ($\varDelta I_{\mathrm{sel}}$) over generations ($t^*$) up to the stationary state. Figure 3.6 shows the relationship among I_{bio}, I_{fit}, I_{Sha} and $\varDelta I_{\mathrm{sel}}$ gained through an adaptive walk process.

The analogy between the concepts in *in vitro* evolution and those in thermodynamics is compiled in Table 3.1. Additionally, in Table 3.2, we show a comparison between the two different selection protocols: the sampling-screening and the selective enrichment. For the description in the two tables, the protocol of the sampling-screening is more generalized as follows: the adaptive walk is performed by M walkers as parent sequences, which are selected as the 1st-Mth fittest mutants from among the N offspring [7].

Table 3.1 An analogy between thermodynamics and *in vitro* evolution

Thermodynamics	*In vitro* Evolution
Number of moles: M	Number of walkers: M
Molar energy or enthalpy: $E_{\mathrm{m}}, H_{\mathrm{m}}$	Mean fitness of walkers: W
Boltzmann constant: k_{B}	Evolutionary Boltzmann constant: k_{E}
Molar entropy: $S_{\mathrm{m}} \equiv k_{\mathrm{B}} \ln \Omega$	Walker's entropy: $S \equiv k_{\mathrm{E}} \ln \Omega$
Temperature: $T_{\mathrm{the}} \equiv \partial E_{\mathrm{m}} / \partial S_{\mathrm{m}}$	Evolutionary temperature: $T_{\mathrm{evo}} \equiv -\mathrm{d}W/\mathrm{d}S$
Chemical potential: $G_{\mathrm{m}} \equiv H_{\mathrm{m}} - T_{\mathrm{the}}\, S_{\mathrm{m}}$	Evolutionary potential: $\phi \equiv W + T_{\mathrm{evo}}\, S$
Free energy: $G \equiv G_{\mathrm{m}} \times M$	Free fitness: $G \equiv \phi \times M$
Generalized force: $X \equiv \mathrm{d}(G_{\mathrm{m}}/T_{\mathrm{the}})/\mathrm{d}x$	Evolutionary force: $X \equiv \mathrm{d}(\phi/T_{\mathrm{evo}})/\mathrm{d}W$
Generalized flux: J	Fitness flux: $J \equiv \mathbf{E}[\Delta W]$
$J \approx LX$	$J \approx LX$
Diffusion coefficient: D	Diffusion coefficient: $D \equiv k_{\mathrm{L}}^2 d/M$
Einstein relation: $L = MD/k_{\mathrm{B}}$	$L = MD/k_{\mathrm{E}}$
Energy fluctuation: $\Sigma_{\mathrm{ene}} \equiv \mathbf{SD}[E_{\mathrm{m}}] \propto k_{\mathrm{B}} T_{\mathrm{the}}$	Fitness fluctuation[a]: $\Sigma_{\mathrm{fit}} \equiv \mathbf{SD}[\Delta W] \propto k_{\mathrm{E}} T_{\mathrm{evo}}$
$\Delta S_{\mathrm{system}}/M = \Delta S_{\mathrm{m}}$	Shannon information gain : $\Delta I_{\mathrm{Sha}} = -\Delta S$
$\Delta S_{\mathrm{surroundings}}/M = -\Delta H_{\mathrm{m}}/T_{\mathrm{the}}$	Fitness information gain: $\Delta I_{\mathrm{fit}} = \Delta W/T_{\mathrm{evo}}$
$\Delta S_{\mathrm{universe}}/M = -\Delta G_{\mathrm{m}}/T_{\mathrm{the}}$	Biomolecular information gain: $\Delta I_{\mathrm{bio}} = \Delta\phi/T_{\mathrm{evo}}$
$\Delta S_{\mathrm{universe}} = \Delta S_{\mathrm{surroundings}} + \Delta S_{\mathrm{system}}$	$\Delta I_{\mathrm{bio}} = \Delta I_{\mathrm{fit}} - \Delta I_{\mathrm{Sha}}$

a): For the case of the sampling-screening. As for the protocol of the sampling-screening, the adaptive walk is performed by M walkers as parent sequences, which are selected as the 1st-Mth fittest mutants from among the N offspring. Adapted from Aita and Husimi [4, 7].

Table 3.2 A comparison between two different selection protocols: the sampling-screening and the selective enrichment

Selection scheme	Sampling-screening[a]	Selective enrichment[b]
Evolutionary Boltzmann constant: k_{E}	Landscape constant: $k_{\mathrm{L}} \equiv \sigma\sqrt{1+k}$	Boltzmann constant: k_{B}
Evolutionary temperature: T_{evo}	$\sqrt{d/4\ln(N/M)}$	$T_{\mathrm{the}}/2r$
Biophysical meaning[c] of $k_{\mathrm{E}} T_{\mathrm{evo}}$	Fitness fluctuation: $\mathbf{SD}[\Delta W] \times (\sqrt{M}+1)/4$	Energy fluctuation: $\mathbf{SD}[E]/r\sqrt{2f}$
$\Delta I_{\mathrm{sel}}/k_{\mathrm{E}}$	$\ln N - \ln M$	$-\sum_{s=1}^{N_d} p_0(s) \ln p_0(s)$ $+\sum_{s=1}^{N_d} p_1(s) \ln p_1(s)$

a): Random sampling of N mutants from a mutant library and screening of the best M mutants from among the N mutants. b): Each mutant with fitness W is amplified by the Boltzmann factor $\exp(rW/k_{\mathrm{B}} T_{\mathrm{the}})$, where r is the number of rounds of the selective enrichment.

3.4.4 Extent and Content of Information

According to Eigen [25], the extent of information is related to the constrained volume of the sequence space. Therefore, we explicitly define the "extent of information" in *in vitro* evolution in the following. Let p_s be the probability (or frequency) of occurrence of sequence s in a population. The entropy for this state is given by

$$S = -\sum_{s=1}^{\lambda^{\nu}} p_s \log p_s, \tag{3.53}$$

while the maximum entropy is given by $S_{\max} = \nu \log \lambda$, which is for the case where every sequence in the sequence space occurs with the same probability of $p_s = 1/\lambda^{\nu}$. Particularly, $S_{\max}$ is called the "source entropy". The extent of information is defined as

$$I_{\text{extent}} \equiv S_{\max} - S \tag{3.54}$$

(Figure 3.7) [16, 17]. Several concepts of the "amount of genetic information" [35], "genomic complexity" [1], R_{sequence} [34, 44], "functional information" [53], "functional sequence complexity" [21] and I_{Sha} (Equation (3.42)) can be classified to I_{extent}. For example, Szostak introduced the "functional information", which is defined as $-\log_2$ of the fraction of functional sequences that have fitness values (activity of a biopolymer) greater than a specified value [53]. For example, suppose all possible RNA sequences of 4^{70} and the fraction of functional sequences among

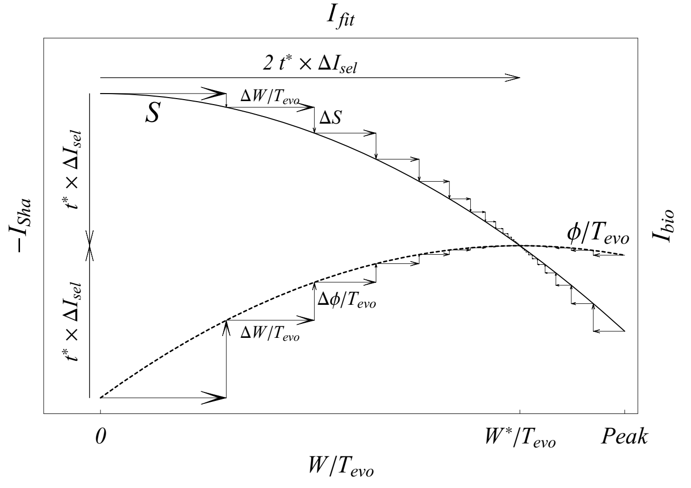


Fig. 3.6 Information gained through the adaptive walk process. The solid curve and dashed curve represent the entropy S and ϕ/T_{evo} as functions of W/T_{evo}, respectively. Each short arrow along these curves represents the expected change after a single step of the walk. $I_{\text{bio}}(W) \equiv \phi(W)/T_{\text{evo}} - \phi(0)/T_{\text{evo}}$. $I_{\text{Sha}}(W) \equiv S(0) - S(W)$. $I_{\text{fit}}(W) \equiv W/T_{\text{evo}}$. At the attractor W^*, $I_{\text{bio}} = I_{\text{Sha}} = I_{\text{fit}}/2$. Adapted from Aita and Husimi [2, 8].

them is 10^{-11}, the functional information in this case is 37 bit compared with 140 bit to specify a unique 70-mer sequence. He suggested the importance of the activity - functional information relationship, which affects the evolvability of the biopolymer. In terms of I_{extent}, [7] discussed the energetic cost and entropic cost for the most efficient search algorithm for several types of fitness landscapes, and concluded that the sum of the two costs is approximately equal to $S_{\text{max}} = \nu \log \lambda$ in any cases.

On the other hand, the content of information means the "meaning (=semantic value) of the message" and the "effectiveness (=pragmatic value) of the message", and it is related with the concept of fitness. Weaver commented three levels of communication problems: Level A (technical problem), level B (semantic problem) and level C (effectiveness problem) [48]. He noticed that Shannon's communication or information theory treated only level A, which is independent of value or content of the messages. This issue is discussed in the following two subsections.

3.4.5 Fitness Information

For adaptation and existence under given conditions, the evolving biopolymers gain high fitnesses from the surroundings. We interpret that the fitness information I_{fit} defined in Equation (3.41) is a measure of the content of information. For example, concerning the emergence of a specific ligand monomeric sequence with high affinity to target-receptor molecules (see Figure 3.1), as is mentioned in Section 3.1, the emerging ligands gain I_{fit} from the receptor under experimental conditions. In this case, fitness should be defined as the natural logarithm of the association

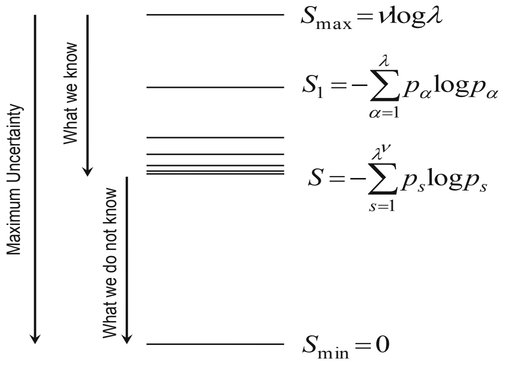

Maximum Uncertainty
What we know
What we do not know
$S_{\text{max}} = \nu\log\lambda$
$S_1 = -\sum_{\alpha=1}^{\lambda} p_\alpha \log p_\alpha$
$S = -\sum_{s=1}^{\lambda^\nu} p_s \log p_s$
$S_{\text{min}} = 0$

Fig. 3.7 Schematic diagram representing the meaning of Shannon's entropy in its meaning as "extent" of information. In S_1, p_α is the probability of appearance of a symbol of class α (e.g. amino acid residues); the sum is taken over all λ classes of symbols. In S, p_s denotes the probability of appearance of a certain sequence s. Here the sum has to be taken over all λ^ν possible sequences of length ν. S takes into account all possible long-range symbol interactions. In a protein chain a given symbol might be in interaction with any other symbol via folding of the chain. Adapted from Eigen [25].

constant between the ligand and the receptor, i.e. $W \equiv \ln K_{\mathrm{a}}$. The value of the content of information is quantified by dividing a fitness change, $\Delta \ln K_{\mathrm{a}}$, by evolutionary temperature T_{evo}. For the sampling-screening, it is done by scaling $\Delta \ln K_{\mathrm{a}}$ by the fitness fluctuation for a single step of walk, $\mathbf{SD}[\Delta \ln K_{\mathrm{a}}]$. By following this operation, the fitness can be handled within the same level of Shannon information, as is shown in Equation (3.44).

Similarly, the concept of fitness information can be applied to the genetic-drift model [13, 32, 47] mentioned in Section 3.3. This model is based on the strong genetic-drift effect with weak mutational effect in a finite population. In Equation (3.8), *Fitness* corresponds to the population average of the selection coefficient, $\langle$*selection coefficient*$\rangle$, and the *Temperature-like parameter* corresponds to the inverse of the double population size, $1/2N$. The $1/4N$ is known as a critical value of the selection coefficient between selective neutrality and selective advantage [36]. That is, $1/4N \sim 1/2N$ can be regarded as a resolving power of the selection coefficient. Therefore, the quantity of $\langle$*selection coefficient*$\rangle$ divided by $1/2N$ is similar to the fitness information mentioned above.

In vitro evolution of artificial selection-type (or directed evolution), the functional property of the evolving biopolymers, that is, its fitness is regarded as a meaning of the message in a monomeric sequence. Thus, the fitness information is a measure of the semantic value of information (level B). The pragmatic value of information (level C) depends on the utilization of the biopolymers by the experimenter of directed evolution, and beyond the scope of our analysis.

On the other hand, *in vitro* evolution of natural selection-type, the fitness is the reproduction rate constant and the evolving biopolymers effectively reproduce by themselves. Thus, the pragmatic value of information corresponds to the content of information (the semantics is in a black box in this case), and the effectiveness of the messages is measured by the results of reproduction based on the fitnesses. An example of this measure is introduced in the next subsection.

3.4.6 Pragmatic Information

Assuming that the pragmatic value of information stems from its usefulness in making an informed decision, Weinberger proposed "pragmatic information", which quantifies the impact of a message on the receiver's actions [56]. His theory is based on a communication system consisting of a "decision maker" and an "effector". Figure 3.8 shows a conceptual framework for pragmatic information.

Suppose that the decision maker, in some current state $\mathscr{S}$, receives a set of $\mathscr{M}$ messages ($m = 1,2,3,\cdots,\mathscr{M}$) and chooses a message m among the set with the probability of φ_m. Subsequently, based on the practical meaning of the chosen message m, the effector assigns a probability value to each of the $\mathscr{N}$ outcomes ($o = 1,2,3,\cdots,\mathscr{N}$). The probability of an outcome o without any message is given by q_o, and the conditional probability of an outcome o with a message m is given by $P_{o|m}$. The pragmatic information of the message ensemble is defined by

$$I_{\text{pra}} \equiv \sum_{m=1}^{\mathscr{M}} \varphi_m \sum_{o=1}^{\mathscr{N}} P_{o|m} \log_2 \left(\frac{P_{o|m}}{q_o} \right). \tag{3.55}$$

Therefore, the pragmatic information is the average of the relative entropy between $\{P_{o|m}\}$ and $\{q_o\}$ over the message ensemble.

Weinberger demonstrated the pragmatic information is a global Lyapunov function for the quasi-species model (Section 3.2.1). Here, the environment (= setup of the flow reactor) effectively decides the fitness of phenotype corresponding to each given genotype, where a phenotype's fitness is defined to be its reproduction rate. At each time t, the environment receives "messages" about the fitness of a particular replicator via the number of copies of that replicator's genome, where the only messages received are measurements at various times. In Equation (3.55), $\mathscr{M} = 1$ and then $\varphi_m = 1$, and an outcome o corresponds to a species s. Prior to receipt of the messages, the initial probability of selecting a species s at random from the system is $q_s = x_s(0)$. The probability of selecting a species s at subsequent time t is $P_{s|m} = x_s(t)$. Then, the pragmatic information for the quasi-species model is given by

$$I_{\text{pra}} = \sum_{s=1}^{n} x_s(t) \log_2 \left(\frac{x_s(t)}{x_s(0)} \right). \tag{3.56}$$

Regardless of the arbitrary initial distribution, $\{x_s(0)\}$, $\mathrm{d}I_{\text{pra}}/\mathrm{d}t > 0$ holds for all finite times. The pragmatic information is generated through the process of evolution for the quasi-species. In Table 3.3, we compiled the list of several aspects of information introduced in this chapter.

Table 3.3 List of several aspects of information introduced in this chapter

	Extent of information	Content of information		Lyapunov function
	$I_{\text{extent}}(\equiv S_{\max} - S)$	Semantic value	Pragmatic value	
Natural selection-type	I_{extent}	?	I_{pra}	I_{pra}
Artificial selection-type	I_{Sha}	I_{fit}	?	$I_{\text{bio}}(\equiv I_{\text{fit}} - I_{\text{Sha}})$

The motivation of establishing pragmatic information is very sound, however, we are uncertain whether Equation (3.56) reflects the effects of absolute fitness values on the resulting probability distribution $\{x_s(t)\}$. In evolution as the fitness gaining process, the absolute fitness value that the evolving biopolymer stores seems essential for its pragmatic value. For example, consider two fitness landscapes with a common shape but different absolute heights and the quasi-species on each landscape. We can expect that the resulting probability distribution of the quasi-species seems to be identical with each other. However, the absolute fitness values that the quasi-species store are different from each other.

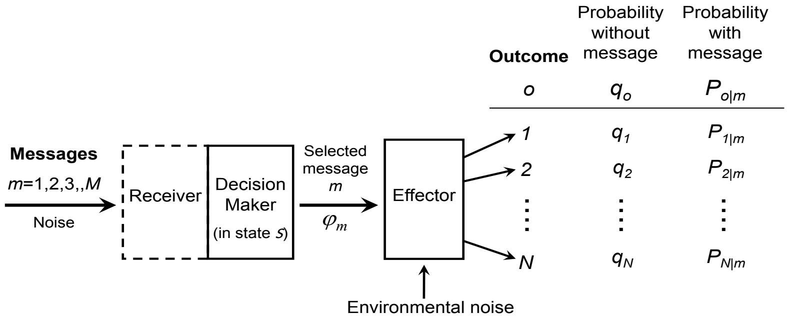


Fig. 3.8 Conceptual framework for pragmatic information. Adapted from Weinberger [56] and Aita Husimi [7].

3.5 Conclusion

As an analogy between *in vitro* evolution and thermodynamics, some studies introduced the concept of "free fitness" which is analogous to free energy, and then proposed that evolution is driven in the direction in which the free fitness increases. Although the mathematical definition of the free fitness is slightly different by their evolution models, in their common view, the free fitness is given as *Free fitness* $\equiv$ *Fitness* + *Temperature-like parameter* × *Entropy*, where *Fitness* is sometimes population average of log fitness or selection coefficient. Particularly, the *Temperature-like parameter* is dependent on the models. For example, it is given as a function of population size in several studies, or given as a function of population size (library size) and mutation rate (mutation distance) in other studies. The mathematical definition of the free fitness remains to be unified.

In a thermodynamic system, a spontaneous process is driven in the direction in which the total entropy of the universe increases. Namely, under a constant temperature and constant external pressure, $-\Delta G/T_{\rm the} = -\Delta H/T_{\rm the} + \Delta S \geq 0$ (G: Gibbs free energy of the system; $T_{\rm the}$: thermodynamic temperature; H: enthalpy of the system; S: entropy of the system). The quantities $-\Delta H/T_{\rm the}$ and $-\Delta G/T_{\rm the}$ represent the thermal entropy that the system emits to the surroundings and the change in total entropy of the universe, respectively. Then, returning to *in vitro* evolution, we focus on the novel quantity of *Fitness* divided by *Temperature-like parameter*, and regard this quantity as the content of information, while we regard *Entropy* as the extent of information. Namely, the evolving biopolymer stores the extent and content of information through the evolution process (Table 3.3).

The extent of information is related to the constrained volume of the sequence space and is defined as $I_{\rm extent} \equiv S_{\rm max} - S$ (Equation (3.54)). This is comprehended within the framework of the Shannon's information theory. Therefore, we designated $I_{\rm extent}$ as the Shannon information $I_{\rm Sha}$ in our mathematical framework (Equation (3.42)).

On the other hand, the content of information is a controversial issue. In our theory based on Kauffman's NK landscape, we introduced the "fitness information" as a measure of the content of information. The fitness information $I_{\rm fit}$ is defined by dividing the fitness W by evolutionary temperature $T_{\rm evo}$ (Equation (3.41)). Particularly, for the sampling-screening type, the quantity $k_{\rm E}T_{\rm evo}$ ($k_{\rm E} \equiv$ the "evolutionary Boltzmann constant") is nearly equal to the fitness fluctuation for a single step of walk, $\mathbf{SD}[\Delta W]$ ($\equiv$ the standard deviation of a fitness change ΔW for a single step of walk) (Equation (3.39)). Then, we can interpret that the fitness information represents the scaled fitness by the fitness fluctuation $\mathbf{SD}[\Delta W]$. Furthermore, the "biomolecular information" $I_{\rm bio}$ is defined by dividing the evolutionary potential ϕ (Equation (3.32)) by evolutionary temperature $T_{\rm evo}$ (Equation (3.43)), and then consists of the content ($I_{\rm fit}$) and extent ($I_{\rm Sha}$) of information (Equation (3.44) and Figure 3.5). $I_{\rm bio}$ is a Lyapunov function of the *in vitro* evolution process (Equation (3.47)).

From the viewpoint of the effectiveness of messages (=information) on the receiver's actions, Weinberger introduced pragmatic information $I_{\rm pra}$ (Equation (3.55)), which is defined as the mean relative entropy between probability distribution of actions without and with messages. The pragmatic information he defined for the quasi-species model is a Lyapunov function. While the motivation of establishing the pragmatic information is very sound, we can not properly evaluate the validity of his definition (Equation (3.56)).

In addition, these concepts should be related with the dissipative structure concept in nonequilibrium thermodynamics, especially, thermodynamic entropy production in the evolving system and the surroundings (see Nicolis and Prigogine [41] or Smith [49, 50, 51].

References

[1] Adami, C., Ofria, C., Collier, T.C.: Evolution of biological complexity. Proc. Natl. Acad. Sci. USA 97, 4463–4468 (2000)

[2] Aita, T., Morinaga, S., Husimi, Y.: Thermodynamical interpretation of evolutionary dynamics on a fitness landscape in an evolution reactor, I. Bull. Math. Biol. 66, 1371–1403 (2004)

[3] Aita, T., Morinaga, S., Husimi, Y.: Thermodynamical interpretation of evolutionary dynamics on a fitness landscape in an evolution reactor, II. Bull. Math. Biol. 66, 1371–1403 (2005)

[4] Aita, T., Husimi, Y.: An interpretation of evolutionary dynamics in in vitro molecular evolution from thermodynamical and informational viewpoint. Seibutsu Butsuri (Biophysics) 46, 137–143 (2006) (in Japanese with English abstracts)

[5] Aita, T., Hayashi, Y., Toyota, H., Husimi, Y., Urabe, I., Yomo, T.: Extracting characteristic properties of fitness landscape from in vitro molecular evolution: A case study on infectivity of fd phage to E.coli. J. Theor. Biol. 246, 538–550 (2007)

[6] Aita, T.: A trade-off relationship between energetic cost and entropic cost for in vitro evolution. Biosystems 101, 194–199 (2010)

[7] Aita, T., Husimi, Y.: Biomolecular information gained through in vitro evolution. Biophys. Rev. 2, 1–11 (2010)

[8] Aita, T., Husimi, Y.: Biophysical connection between evolutionary dynamics and thermodynamics in vitro evolution. in vitro 294, 122–129 (2012)

[9] Ao, P.: Emerging of stochastic dynamical equalities and steady state thermodynamics from Darwinian dynamics. Communications in Theoretical Physics 49, 1073–1090 (2008)

[10] Atkins, P.W.: Physical Chemistry. Oxford University Press, Oxford (1978)

[11] Barton, N.H., Coe, J.: On the application of statistical physics to evolutionary biology. J. Theor. Biol. 259, 317–324 (2009)

[12] Barton, N.H., de Vladar, H.P.: Statistical mechanics and the evolution of polygenic quantitative traits. Genetics 181, 997–1011 (2009)

[13] Berg, J., Willmann, S., Lässig, M.: Adaptive evolution of transcription factor binding sites. BMC Evol. Biol. 4, 42-1-12 (2004)

[14] Biebricher, C.K., Eigen, M., Luce, R.: Product analysis of RNA generated de-novo by phage Q-beta replicase. J. Mol. Biol. 148, 369–390 (1981)

[15] Blackburne, B.P., Hirst, J.D.: Population dynamics simulations of functional model proteins. J. Chemical Physics 123, 154907 (2005)

[16] Brillouin, L.: Negentropy principle of information. J. of Applied Physics 24, 1152–1163 (1953)

[17] Brillouin, L.: Science and Information Theory. Academic Press, New York (1956)

[18] Demetrius, L.: Directionality principles in thermodynamics and evolution. Proc. Natl. Acad. Sci. USA 94, 3491–3498 (1997)

[19] Demetrius, L.: Thermodynamics and evolution. J. Theor. Biol. 206, 1–16 (2000)

[20] de Vladar, H.P., Barton, N.H.: The contribution of statistical physics to evolutionary biology. Trends Ecol. Evol. 26, 424–432 (2011)

[21] Durston, K.K., Chiu, D.K., Abel, D.L., Trevors, J.T.: Measuring the functional sequence complexity of proteins. Theor. Biol. Med. Model. 4, 47-1-14 (2007)

[22] Eigen, M.: Self-organization of matter and the evolution of biological macromolecules. Naturwissenschaften 58, 465–523 (1971)

[23] Eigen, M., Schuster, P.: The Hypercycle. Springer, Berlin (1979)

[24] Eigen, M., Gardiner, W.: Evolutionary Molecular Engineering Based on RNA Replication. Pure and Appl. Chem. 56, 967–978 (1984)

[25] Eigen, M.: Natural selection: a phase transition? Biophysical Chemistry 85, 101–123 (2000)

[26] Einstein, A.: Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen (On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat.). Ann. D. Phys. 17, 549–560 (1905)

[27] Ellington, A.D., Szostak, J.W.: in Vitro selection of RNA molecules that bind specific ligands. selection of RNA molecules that bind specific ligands 346, 818–822 (1990)

[28] Fisher, R.A.: The Genetical Theory of Natural Selection. Clarendon, Oxford (1930)

[29] Husimi, Y., Nishigaki, K., Kinoshita, Y., Tanaka, T.: Cellstat-a continuous culture system of a bacteriophage for the study of the mutation rate and the selection process of the DNA level. Rev. Sci. Instrum. 53, 517–522 (1982)

[30] Husimi, Y.: Selective value landscape on the base sequence space and concept of free selective value: a model of quasi-species. Viva Origino 16, 136–141 (1988) (in Japanese with English abstracts)

[31] Iguchi, K.: Reciprocal relations in evolutionary processes. Prog. Theor. Phys. Suppl. 173, 235–242 (2008)

[32] Iwasa, Y.: Free fitness that always increases in evolution. J. Theor. Biol. 135, 265–281 (1988)

[33] Kauffman, S.A.: The Origin of Order. Oxford University Press, Oxford (1993)

[34] Kim, J.T., Martinetz, T., Polani, D.: Bioinformatic principles underlying the information content of transcription factor binding sites. J. Theor. Biol. 220, 529–544 (2003)

[35] Kimura, M.: Natural selection as the process of accumulating genetic information in adaptive evolution. Genet. Res. 2, 127–140 (1961)

[36] Kimura, M.: Genetic variability maintained in a finite population due to mutational production of neutral or nearly neutral isoalleles. Genet. Res. Camb. 11, 247–269 (1968)

[37] Maynard-Smith, J.: Natural selection and the concept of a protein space. Nature 225, 563–564 (1970)

[38] Maynard-Smith, J.: The concept of information in biology. Philos. Sci. 67, 177–194 (2000)

[39] Mills, D.R., Peterson, R.L., Spiegelman, S.: An extracellular Darwinian experiment with a self-duplicating nucleic acid molecule. Proc. Natl. Acad. Sci. USA 58, 217–224 (1967)

[40] Nemoto, N., Miyamoto-Sato, E., Husimi, Y., Yanagawa, H.: In vitro virus: bonding of mRNA bearing puromycin at the 3'-terminal end to the C-terminal end of its encoded protein on the ribosome in vitro. FEBS Lett. 414, 405–408 (1997)

[41] Nicolis, G., Prigogine, I.: Self-Organization in Nonequilibrium Systems: From Dissipative Structures to Order through Fluctuations. Wiley, Chichester (1977)

[42] Pande, V.S., Grosberg, A.Y., Tanaka, T.: Statistical mechanics of simple models of protein folding and design. Biophys. J. 73, 3192–3210 (1997)

[43] Sato, K., Ito, Y., Yomo, T., Kaneko, K.: On the relation between fluctuation and response in biological systems. Proc. Natl. Acad. Sci. USA 100, 14086–14090 (2003)

[44] Schneider, T.D.: Evolution of biological information. Nucleic Acid Research 28, 2794–2799 (2000)

[45] Schuster, P., Swetina, J.: Stationary mutant distributions and evolutionary optimization. Bull. Math. Biol. 50, 635–660 (1988)

[46] Scott, J.K., Smith, G.P.: Searching for peptide ligands with an epitope library. Science 249, 386–390 (1990)

[47] Sella, G., Hirsh, A.E.: The application of statistical physics to evolutionary biology. Proc. Natl. Acad. Sci. USA 102, 9541–9546 (2005)

[48] Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. Univ. Illinois Press, Champaign (1949)

[49] Smith, E.: Thermodynamics of natural selection I: Energy flow and the limits on organization. J. Theor. Biol. 252, 185–197 (2008)

[50] Smith, E.: Thermodynamics of natural selection II: Chemical Carnot cycles. J. Theor. Biol. 252, 198–212 (2008)

[51] Smith, E.: Thermodynamics of natural selection III: Landauer's principle in computation and chemistry. J. Theor. Biol. 252, 213–220 (2008)

[52] Stemmer, W.P.C.: The evolution of molecular computation. Science 270, 1510 (1995)

[53] Szostak, J.W.: Functional information: Molecular messages. Nature 423, 689 (2003)

[54] Tuerk, C., Gold, L.: Systematic evolution of nucleic acids by exponential enrichment: RNA nucleic acids to bacteriophage T4 DNA polymerase. Science 249, 505–510 (1990)

[55] Voigt, C.A., Kauffman, S., Wang, Z.G.: Rational evolutionary design: the theory of in vitro protein evolution. Advances in Protein Chemistry 55, 79–160 (2000)

[56] Weinberger, E.D.: A theory of pragmatic information and its application to the quasi-species model of biological evolution. Biosystems 66, 105–119 (2002)

[57] Wilke, C.O., Wang, J.L., Ofria, C., Lenski, R.E., Adami, C.: Evolution of digital organisms at high mutation rates leads to survival of the flattest. Nature 412, 331–333 (2001)

[58] Wolynes, P.G., Luthey-Schulten, Z.: The energy landscape theory of protein folding. In: Flyvbjerg, H., Hertz, J., Jensen, M.H., Mouritsen, O.G., Sneppen, K. (eds.) Physics of Biological Systems: From Molecules to Species, pp. 61–79. Springer, Heidelberg (1997)

[59] Wright, S.: Evolution in Mendelian populations. Genetics 16, 97–159 (1931)

[60] Yang, X., Liu, X., Lou, C., Chen, J., Ouyang, Q.: A case study of the dynamics of in vitro DNA evolution under constant selection pressure. J. Mol. Evol. 68, 14–27 (2009)

Part II
Topology, Measures and Problem Hardness

Chapter 4
Fitness Landscape Analysis for Metaheuristic Performance Prediction

Katherine M. Malan and Andries P. Engelbrecht

Abstract. Metaheuristics have become popular for solving complex optimisation problems where classical techniques are either infeasible or perform poorly. Despite many success stories, it is well known that metaheuristics sometimes fail and that researchers and practitioners frequently resort to trial and error to find an appropriate algorithm or setting to solve a given problem. Within the framework of the general algorithm selection problem, this chapter addresses the feasibility of predicting algorithm performance on unknown real-valued problems based on fitness landscape features. Normalized metrics are proposed for quantifying algorithm performance on known problems to generate suitable training data. Performance metrics are tested using a standard particle swarm optimisation algorithm and are investigated alongside three existing fitness landscape measures. This preliminary investigation highlights the need for a shift in focus away from predicting general problem hardness towards characterising problems where each fitness landscape technique has value as a part-predictor of algorithm performance.

4.1 Introduction

Many real-world scenarios require optimisation; finding the most profitable or least expensive or most desirable solution to some problem. Traditional mathematical optimisation techniques, such as gradient-based techniques, use the derivative of the objective function to determine the exact maximum or minimum point of a continuous problem. Many optimisation problems are, however, not able to be solved using such traditional techniques. For example, multimodal problems where gradient information is not sufficient for finding the global optimum, problems with discontinuous search spaces that are non-differentiable, or black-box optimisation problems where there is no objective function in mathematical form to be

Katherine M. Malan · Andries P. Engelbrecht
Department of Computer Science, University of Pretoria, Pretoria 0002, South Africa
e-mail: {kmalan,engel}@cs.up.ac.za

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_4, © Springer-Verlag Berlin Heidelberg 2014

differentiated. For these cases, often the only feasible alternative is to use approaches that find approximate solutions. Metaheuristics is the name given to such approximate optimisation techniques that are general, in that they can be applied to solve any optimisation problem.

There are many different approaches within the field of metaheuristics. Some metaheuristics perform local search, usually with extensions to enhance exploration of the search space, such as simulated annealing [6, 31], iterated local search [35] and tabu search [16, 17]. Other metaheuristics are population-based in that they work on a collection of solutions in parallel, such as evolutionary algorithms [3], ant colony optimisation algorithms [9] and particle swarm optimisation (PSO) algorithms [30]. Many of these population-based metaheuristics are inspired by nature; the way that nature constructs elegant solutions to problems within extremely complex systems has inspired scientists to capture some of this 'magic' in simple forms within algorithms and the results are often surprisingly good. Talbi [56] provides a genealogy of metaheuristics as applied to optimisation from the 1940s to the 1990s containing no less than 24 different kinds of algorithms. Since then many more algorithms have been proposed from more bio-inspired algorithms like firefly [69] and krill herd [14] algorithms to algorithms inspired by the interaction between magnetic particles [57] and even algorithms inspired by the way musicians improvise (harmony search) [15]. Given this plethora of algorithms, the challenge of choosing the most appropriate algorithm for solving a given problem can be a daunting task.

This chapter considers the subset of optimisation problems that involve real-valued variables. More precisely, the scope of the problems under investigation is limited to optimisation problems that are static, bound-constrained, multivariate and continuous and it is assumed that problems are to be minimized. In general such a problem can be defined as

$$\min f(\mathbf{x}), f : \mathbb{R}^n \to \mathbb{R}, \mathbf{x} \in \mathscr{S} \subseteq \mathbb{R}^n$$

where $\mathbf{x}$ is an n-dimensional candidate solution vector and $\mathscr{S}$ defines the feasible subregion of $\mathbb{R}^n$ as defined by the domains of the variables within $\mathbf{x}$. Considering only continuous optimisation problems does not simplify the problem of choosing an appropriate algorithm. Although some algorithms were designed for discrete spaces, there are just as many algorithms that were designed to work in continuous spaces. Examples include PSO [30], differential evolution [43], evolution strategies [45, 47] and cuckoo search [70]. In addition, many metaheuristics originally designed for discrete spaces have been adapted to work in continuous environments such as real-coded genetic algorithms [24], evolutionary programming [32], ant colony optimisation algorithms [4], estimation of distribution algorithms [1] and scatter search [23].

Over the last few decades the focus of research in metaheuristics has been largely on the algorithmic side. Relatively little attention has been paid to the study of the problems. When a publication introduces a new algorithm or variation on an existing algorithm, the approach is typically to demonstrate empirically that the algorithm out-performs other algorithms on a number of selected benchmark problems.

The proposers of a new algorithm will usually neglect to provide any analysis of problems on which the proposed algorithm will perform poorly and why.

There have been some recent theoretical studies attempting to address this gap in understanding of algorithm behaviour on problems. These include: analysing which problems are hard for particular algorithms [7, 22]; analysing which problems require smaller population sizes in evolutionary algorithms [8]; and analysing features of particular problems that make them hard for particular algorithms [40]. Despite these advances in understanding, there is still little practical guidance for researchers and practitioners. Questions such as ‘Which algorithm will most accurately solve my problem?’ or ‘Which algorithm will most quickly produce a reasonable answer to my problem?’ remain unanswered. The most common technique for choosing an appropriate algorithm for a given problem is by trial and error. If there existed one algorithm that out-performed all others in solving optimisation problems, then this ‘super-algorithm’ could be used in all cases. It is well known, however, that no such algorithm can exist as was proved by Wolpert and Macready with their famous ‘No-Free-Lunch’ theorems for search/optimisation [66, 67]. The emphasis is therefore not on finding the best optimisation algorithm in general, but rather on finding the most appropriate search process for solving a particular problem, whether this is a particular algorithm with appropriate settings and parameters or a combination of several different search algorithms.

The focus of this chapter is to propose a way in which the features of a problem, based on analysis of the fitness landscapes of continuous problems in particular, can be used to predict the performance of a given metaheuristic. This is not a new problem or idea. Many have tried to predict problem hardness and many have failed, leading to the conclusion that no satisfactory problem difficulty measure for search heuristics has been found [20, 26]. In fact, trying to find a computationally feasible hardness measure is a futile exercise, since He *et al.* [21] have proved that a predictive version of such a measure, i.e. that runs in polynomial-time, cannot exist (unless P=NP or BPP[1]=NP). Therefore, rather than proposing yet another difficulty measure, this chapter proposes a framework within which existing approximate measures can be used together to solve the problem of performance prediction. There are many proposed measures based on the analysis of fitness landscapes that show some correlation to algorithm performance. Approximating these measures on a selection of benchmark problems and combining this data with actual algorithm performance could form a dataset to be used with one of the many machine learning algorithms to find a mapping from problem features to performance.

The chapter is organised as follows: In the following section, the general algorithm selection problem and related performance prediction problem are described. Section 4.3 proposes performance metrics that are suitable for the algorithm selection framework. Section 4.4 addresses the challenge of feature extraction of continuous optimisation problems and three possible feature metrics are investigated in terms of the link to algorithm performance. The chapter is concluded in Section 4.5.

[1] BPP: bounded-error probabilistic, polynomial time [64].

4.2 The Algorithm Selection Problem

The general problem of selecting an effective or good or best algorithm to solve a given problem was formulated by Rice in the 1970s [46]. One of the models described by Rice is the model where algorithm selection is based on features of the problem. This model has four main characteristics:

- A set of problem instances (problem space P),
- a set of algorithms for solving the problems (algorithm space A),
- measures for comparing the performance of algorithms on a particular problem (performance measure space Y), and
- measurable characteristics of the problem instances (feature space G).

The relationship between these components is illustrated in Figure 4.1 (this diagram is based on Rice's Figure 3 [46] but includes an additional mapping for performance prediction). A given algorithm $a \in A$ can be applied to a problem instance p to produce performance measure(s) $y(a(p))$. In a trial-and-error approach to finding the best algorithm to solve a problem, this process of applying algorithms to a problem is simply repeated until the best algorithm from a set of algorithms is found, based on the given performance measures. The algorithm selection problem, however, involves avoiding this trial-and-error approach by achieving the following:

- Feature extraction: devising a mapping from problem space to feature space, so that any problem instance p can be characterised by features $g(p)$; and
- Algorithm selection: devising a mapping from problem feature space to algorithm space, so that a given problem p, with extracted features $g(p)$, can be matched to the most appropriate algorithm a, such that performance $y(a(p))$ is maximised.

Figure 4.1 also illustrates the related problem of performance prediction, which involves predicting the performance $y(a(p))$ of a given algorithm a applied to problem p based on extracted features $g(p)$. If a solution to the algorithm selection / performance prediction problem is found, it becomes possible to take an unseen optimisation problem, extract its features, and from these features select the most appropriate metaheuristic algorithm from a subset of metaheuristics for solving the problem (the algorithm selection problem) or predict the performance of a given metaheuristic algorithm on the problem (the performance prediction problem).

Smith-Miles [49] used Rice's model [46] to address the algorithm selection problem for a subset of combinatorial optimisation problems, namely quadratic assignment problems. In that study, 28 instances of the problem were used with three metaheuristics (tabu search, iterated local search and min-max ant system). The features of the problem were a combination of measures quantifying the size of the quadratic assignment problem with fitness-distance metrics based on local search runs (requiring knowledge of the global optima). A neural network was used to solve the problem of mapping problem features to performance measures. Although restricted to a specific class of optimisation problems, the study by Smith-Miles demonstrated the potential relevance of using such an approach. This chapter proposes how Rice's model [46] can be applied to continuous optimisation problems

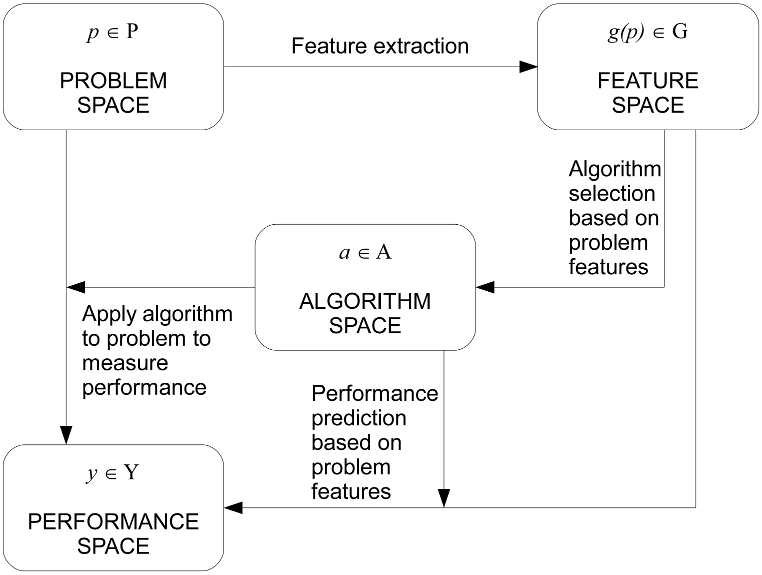

$p \in$ P
PROBLEM SPACE
Feature extraction
$g(p) \in$ G
FEATURE SPACE
Algorithm selection based on problem features
$a \in$ A
ALGORITHM SPACE
Apply algorithm to problem to measure performance
Performance prediction based on problem features
$y \in$ Y
PERFORMANCE SPACE

Fig. 4.1 A framework for describing the general problems of algorithm selection and performance prediction based on problem features (based Rice's model [46])

and algorithms, where the features of the problem are based on the analysis of fitness landscapes. The features should be computable on unseen problems (without prior knowledge of the problem, such as knowledge of the global optima).

The following are prerequisites to solving the algorithm selection problem [50] in general:

(1) A large number of problem instances with different levels of difficulty;
(2) A large number of different algorithms for solving these problem instances;
(3) Metrics for evaluating the performance of algorithms; and
(4) The existence of features that can be used to suitably characterise the properties of problems.

Items 1 and 2 are met, given the extensive range of benchmark optimisation problems in the literature and the many algorithms for solving continuous problems, as discussed in Section 4.1. Requirement 3 is addressed in the next section and requirement 4 is discussed in Section 4.4.

4.3 Performance Metrics

Solving the algorithm selection problem in general requires suitable metrics for evaluating the performance of algorithms on a given set of known problems. The performance metrics are used to generate a data set that can be used as the basis for finding a mapping from feature space to algorithm or performance space, as illustrated in Figure 4.1. For any two problems, the metrics should distinguish the relative difficulty of solving the problems by a given algorithm. Equally, given two algorithms, the metrics should distinguish the relative difficulty of the algorithms solving the same problem. This section reviews some existing approaches to measuring performance of optimisation algorithms and then proposes the use of three normalised metrics that can be used for quantifying the performance of algorithms on different problems.

4.3.1 Existing Approaches to Measuring Performance

The most common way of measuring the performance of an optimisation algorithm is in terms of the quality of the solution found in comparison to the quality of the solution found by some other algorithm. Given sufficient independent runs of both algorithms, it is reported whether there is a statistically significant difference in the quality of the solutions found by the algorithms. In the case of benchmark problems with known optimal solutions, the measure of the quality is usually simply the difference in fitness value between the solution found by the algorithm and the optimal solution (sometimes called the fitness error value). To ensure reasonably fair comparisons between different algorithms, the fitness error value is usually based on the best or average solution found after a set number of function evaluations by the algorithm. Using an absolute fitness error value as a measure of performance is suitable when comparing algorithms on the same problem, but cannot be used when comparing the performance of algorithms on different problems with varying fitness ranges. For example, a mean error value of 0.8 on one problem could be regarded as a high performance result, whereas a mean error value of 0.003 may be regarded as a low performance result for a different problem.

Another common approach to measuring performance is to quantify the percentage of successful runs over a number of runs (frequently called success rate). What constitutes a ‘successful run’ has to be properly defined. Some arbitrary cutoff like “within 10^{-8} of the global optimum” could be sensible in the case of a problem with a fitness range of $[0,1]$, but would not make sense in the case of a problem with a fitness range of $[0,10^{30}]$ (such as with the Schwefel 2.22 benchmark function in 30 dimensions). In some cases a fixed error is specified for each benchmark function [42, 53]. For example, Suganthan *et al.* [53] define fixed accuracy levels for each benchmark function, such as 10^{-6} for F_1 (Shifted Sphere Function) and 10^{-2} for F_6 (Shifted Rosenbrock’s Function). They define a successful run as one during which “the algorithm achieves the fixed accuracy level within the Max_FES for the particular dimension”[53], where Max_FES specifies the maximum number of function evaluations and is defined as $10000 \times D$ (the dimension of the problem). In this way, functions with higher dimensions are given more function evaluations to reach the fixed accuracy level.

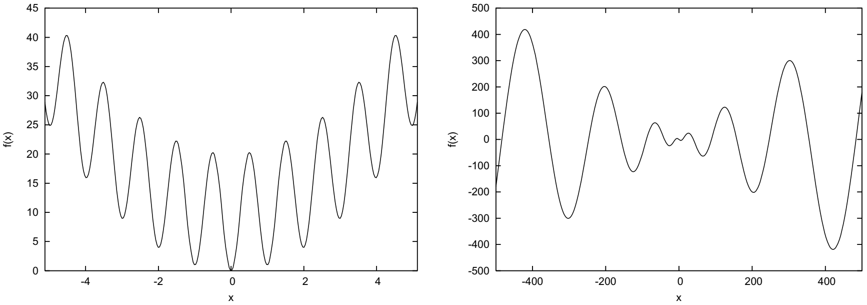
Yet another approach to measuring performance is to quantify how quickly an algorithm is able to find an acceptable solution. For example, the AES measure [12] is defined as the average number of evaluations to a solution, defined over those runs that reach a solution to within a given fixed accuracy level.

4.3.2 Proposed Performance Metrics

This section proposes the use of three normalised algorithm performance measures. Note that the purpose of these measures is to generate data on known problems to be used for the training of a predictor or classifier of algorithm performance on unknown problems. The measures quantify solution quality, rate of success and speed of reaching a solution and can be used to contrast the performance of a single algorithm on multiple problems or multiple algorithms on the same problem. All three measures require knowledge of the range of fitness values of a problem in order for the measures to be normalised across different problems.

4.3.2.1 Estimating Fitness Range

The range of fitness values for benchmark problems is not always known. In the case of simple functions like the Spherical function, it is obvious that the maximum values lie on the boundaries of the search space. The range of fitness values is then simply the difference between the fitness values on the boundary and the fitness of the known optimal solution. In the case of many other benchmark problems, however, the maximum point is somewhere else in the search space. For example, Figure 4.2 illustrates two common benchmark functions in one dimension. A simple visual inspection of a plot of the functions in one dimension can be used to estimate the range of fitness values of the function. However, in higher dimensions it is not as easy to estimate the fitness range, as the position of the maximum point cannot be assumed to be at the same x position as for the one dimensional version of the function.



(a) Rastrigin function

(b) Schwefel 2.26 function

Fig. 4.2 Example one-dimensional minimisation benchmark functions

A simple way of estimating the maximum fitness value of a benchmark problem is to optimise the minimisation problem as a maximisation problem using a suitable optimisation algorithm. The result of the optimisation is the estimated maximum fitness value, called $\hat{f}$. It is important that during the maximisation, the algorithm be confined to the bounds of the search space, because many of the benchmark functions (like those illustrated in Figure 4.2) continue increasing outside the bounds of the problem domain. For a benchmark function, given an estimated maximum fitness $\hat{f}$ and a known minimum fitness value, f^*, the estimated range of fitness values is then defined as $\hat{f} - f^*$.

4.3.2.2 Determining Fixed Accuracy Levels

Measures based on the success rate and speed of an optimisation algorithm are dependent on deciding whether a run is successful or not. Similar to [42] and [53] the notion of a successful run is dependent on a fixed accuracy level for a function. A function with a smaller range of fitness values should have a smaller fixed accuracy level than a function with a larger fitness range. However, rather than using a single fixed accuracy level for each benchmark function (as in [42, 53]), it is proposed that a fixed accuracy level is defined for each function/dimension combination. If the range of fitness values of the same benchmark function increases as the dimension increases, then the fixed accuracy level should also increase. The proposed method for determining the fixed accuracy level of a given benchmark function and dimension is as follows:

(1) The estimated fitness range of the problem $(\hat{f} - f^*)$ is rounded down to the nearest 10^n (called the *fitness range order*), where n is an integer. Rounding the fitness range down (rather than up) results in a smaller fixed accuracy level and hence higher requirement in terms of accuracy of solution.

(2) The *fixed accuracy level* is computed as the fitness range order multiplied by 10^{-8}. The use of 10^{-8} is specifically chosen to align with error values in other sources [42, 53]. The Spherical benchmark function in one dimension with domain $[-100, 100]$ would result in a fitness range of 10^2. Multiplying this by 10^{-8} would result in a fixed accuracy of 10^{-6}, which is equal to the error value specified for Sphere in [42] as well as the fixed accuracy level specified for the Shifted Sphere Function in [53].

Some example benchmark problems with proposed fixed accuracy levels are listed in Table 4.1. The definition of these functions is given in Table 4.2. Notice how with some functions, such as Ackley, the fixed accuracy levels stays the same as the dimension increases, but in the case of other functions, such as Griewank, the fitness range increases with an increase in dimension, resulting in a decrease in the fixed accuracy level.

4.3.2.3 QMetric

Given a run of an optimisation algorithm on benchmark function f with resulting best fitness found f^{min}, the distance of the best found solution from the optimal solution is quantified as $f^{min} - f^*$. This distance is an absolute measure of fitness error, where 0 is the minimum error and corresponds with the highest possible solution quality. To convert the fitness error into a positive measure of quality, the

Table 4.1 Some example benchmark problems (defined in Table 4.2) in different dimensions with estimated maximum fitness ($\hat{f}$), known minimum (f^*), fitness range order and fixed accuracy level

Function	**Dimension**	$\hat{f}$	f^*	**Fitness Range Order**	**Fixed Accuracy Level**
Ackley	1	22.31	0	10^1	10^{-7}
Ackley	15	22.31	0	10^1	10^{-7}
Ackley	30	22.31	0	10^1	10^{-7}
Griewank	1	92.00	0	10^1	10^{-7}
Griewank	15	1351.00	0	10^3	10^{-5}
Griewank	30	2701.00	0	10^3	10^{-5}
Rosenbrock	2	3905.93	0	10^3	10^{-5}
Rosenbrock	15	54682.97	0	10^4	10^{-4}
Rosenbrock	30	113271.86	0	10^5	10^{-3}
Schwefel 2.26	1	418.98	-418.98	10^2	10^{-6}
Schwefel 2.26	15	6284.74	-6284.74	10^4	10^{-4}
Schwefel 2.26	30	12569.49	-12569.49	10^4	10^{-4}

Table 4.2 Benchmark Functions (D is the dimension of the problem)

Function	**Definition, domain and global optimum (f^*)**
Ackley	$f_1(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$ $x_i \in [-32,32], \qquad f_1^* = f_1(0,\dots,0) = 0$
Griewank	$f_2(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{D}x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ $x_i \in [-600,600], \qquad f_2^* = f_2(0,\dots,0) = 0$
Quadric	$f_3(\mathbf{x}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i}x_j\right)^2$ $x_i \in [-100,100], \qquad f_3^* = f_3(0,\dots,0) = 0$
Rana (expanded)	$f_4(\mathbf{x}) = \sum_{i=1}^{D}x_i\sin(\alpha)\cos(\beta) + \left(x_{(i+1)\bmod D} + 1\right)\cos(\alpha)\sin(\beta), \quad D > 1,$ where $\alpha = \sqrt{\lvert x_{i+1} + 1 - x_i\rvert}$ and $\beta = \sqrt{\lvert x_i + x_{i+1} + 1\rvert}$ $x_i \in [-512,512], \qquad f_4^* = f_4(-512,\dots,-512)$
Rastrigin	$f_5(\mathbf{x}) = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$ $x_i \in [-5.12,5.12], \qquad f_5^* = f_5(0,\dots,0) = 0$
Rosenbrock (generalized)	$f_6(\mathbf{x}) = \sum_{i=1}^{D-1}\left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right), \quad D > 1$ $x_i \in [-2.048,2.048], \qquad f_6^* = f_6(1,\dots,1) = 0$
Salomon	$f_7(\mathbf{x}) = -\cos\left(2\pi\sum_{i=1}^{D}x_i^2\right) + 0.1\sqrt{\sum_{i=1}^{D}x_i^2} + 1$ $x_i \in [-100,100], \qquad f_7^* = f_7(0,\dots,0) = 0$
Schwefel 2.26	$f_8(\mathbf{x}) = -\sum_{i=1}^{D}\left(x_i\sin(\sqrt{\lvert x_i\rvert})\right)$ $x_i \in [-500,500], \qquad f_8^* = f_8(420.9687,\dots,420.9687)$
Spherical	$f_9(\mathbf{x}) = \sum_{i=1}^{D}x_i^2$ $x_i \in [-100,100], \qquad f_9^* = f_9(0,\dots,0) = 0$
Step	$f_{10}(\mathbf{x}) = \sum_{i=1}^{D}\left(\lfloor x_i + 0.5\rfloor\right)^2$ $x_i \in [-20,20], \qquad f_{10}^* = f_{10}(0,\dots,0) = 0$

found solution, f^{min}, is subtracted from the estimated maximum, f^* and scaled by the estimated range of the problem as follows:

$$q = \frac{\hat{f} - f^{min}}{\hat{f} - f^*} \; . \tag{4.1}$$

The normalized measure q is a value in the range $[0,1]$ where 1 indicates the highest quality, where the found solution exactly matches the known optimal solution and 0 indicates the worst possible quality of finding the maximum fitness. In order to better distinguish between q values closer to 1, the value of q is scaled exponentially to produce the proposed QMetric measure as follows:

$$\text{QMetric} = 2^{q^{10^4}} - 1 \; . \tag{4.2}$$

Figure 4.3 illustrates the relationship between q and QMetric. Given a problem with a fitness range of $[0,1]$ and a found solution of 10^{-8}, the value of QMetric will be 1.000 (rounded to 3 decimal places), indicating a solution quality that is within the acceptable error margin of the global optimum. On the other hand, a solution of 10^{-5} would result in a q value of 0.99999 and an associated QMetric value of 0.872, indicating a lower solution quality. Any solution with fitness 0.001 and larger will result in a QMetric value of 0 (rounded to 3 decimal places).

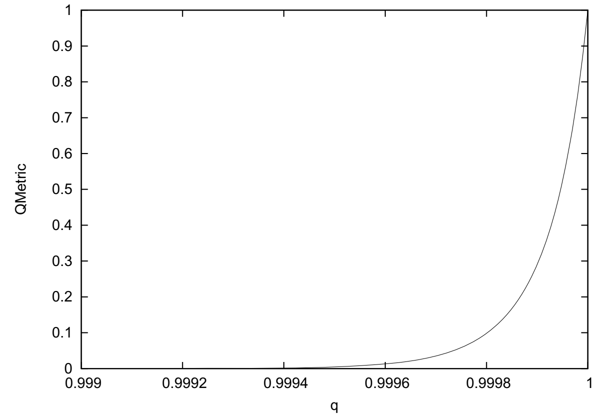


Fig. 4.3 Function used to scale fitness quality measure q to QMetric

4.3.2.4 SRate

To compare the rate of success of different algorithms on different problems, each problem/algorithm combination is run using a maximum number of function evaluations (MaxFES) as the terminating condition. For all problems, the value of MaxFES is set to $10000 \times D$, where D is the dimension of the problem. A PSO algorithm with

50 particles will therefore run for a maximum of 200 iterations on a one-dimensional problem. A run is regarded as successful if the run reaches the global optimum (within the associated fixed accuracy level) before the MaxFES of the problem are exceeded. The success rate (SRate) is defined as the number of successful runs that reach a solution within the fixed accuracy level of the global optimum divided by the total number of runs [53]. Like the QMetric, SRate is a value in the range $[0,1]$ where 1 indicates the highest possible rate of success.

4.3.2.5 SSpeed

The number of function evaluations taken to reach the global optimum (within the fixed accuracy level) for a given run r is known as FES_r. A proposed metric called the success speed of a run r (SSpeed_r) is defined as:

$$\text{SSpeed}_r = \begin{cases} 0 & \text{if the run is not successful} \\ \frac{\text{MaxFES}-(\text{FES}_r-1)}{\text{MaxFES}} & \text{otherwise.} \end{cases} \tag{4.3}$$

The metric SSpeed_r is a value in the range $[0,1]$. The highest value for SSpeed_r can only be obtained if the global minimum is reached in the first function evaluation (if FES_r is 1) and this would indicate the highest possible performance in terms of speed. The success speed (SSpeed) over ns successful runs, is defined as:

$$\text{SSpeed} = \begin{cases} \frac{\sum_{r=1}^{ns} \text{SSpeed}_r}{ns} & \text{if } ns > 0 \\ 0 & \text{if } ns = 0\,. \end{cases} \tag{4.4}$$

4.3.3 Performance Metrics for PSO

To illustrate the use of the proposed three performance metrics, Table 4.3 shows the results of minimizing a number of benchmark functions using a traditional gbest PSO algorithm [10]. The following parameter values were used: 50 particles, 1.496 for both the cognitive and social acceleration constants and 0.7298 for the inertia weight [11]. All algorithms were implemented using CIlib, an open source library and framework of computational intelligence algorithms.[2] Benchmark functions and dimensions were selected to illustrate features of the metrics. For each benchmark function/dimension combination, 30 independent runs of the algorithm were performed. The QMetric values are the means of the 30 runs and the value for MaxFES was set at $10000 \times D$, or $200 \times D$ iterations for 50 particles.

In Table 4.3, the easiest function to minimize is the simple unimodal Spherical function and this is reflected in the high values for all performance metrics. The PSO algorithm achieved a QMetric mean value of 1 in dimensions 1, 15 and 30, indicating that the average of the best fitness values found was of the highest possible quality. Similarly, the SRate of 1 indicates that the algorithm found the optimal solution (to within the fixed accuracy level) in all 30 runs. The high SSpeed

[2] See `http://cilib.net/`

Table 4.3 Results of minimizing benchmark problems using a standard PSO algorithm, based on 30 runs

Function	Dimension	QMetric	SRate	SSpeed
Ackley	1	1.000	1.000	0.471
Ackley	15	0.933	0.933	0.855
Ackley	30	0.300	0.300	0.862
Griewank	1	1.000	1.000	0.721
Griewank	10	0.352	0.033	0.019
Griewank	15	0.691	0.100	0.910
Griewank	30	0.902	0.367	0.921
Rosenbrock	2	1.000	1.000	0.843
Rosenbrock	5	0.931	0.067	0.321
Rosenbrock	15	0.888	0.000	0.000
Rosenbrock	30	0.482	0.000	0.000
Salomon	1	1.000	1.000	0.574
Salomon	5	0.000	0.000	0.000
Schwefel 2.26	1	1.000	1.000	0.816
Schwefel 2.26	5	0.400	0.400	0.826
Schwefel 2.26	15	0.000	0.000	0.000
Spherical	1	1.000	1.000	0.845
Spherical	15	1.000	1.000	0.941
Spherical	30	1.000	1.000	0.940

values indicates that the algorithm found the solution quickly (needing relatively few function evaluations). Recall that the SSpeed metric is a measure of how quickly the solution is found in relation to the maximum number of function evaluations (MaxFES), which increases with dimension (MaxFES is set at $10000 \times D$). For example, an SSpeed value of 0.941 for Spherical in 15 dimensions does not imply that the solution was found in fewer iterations than for 1 dimension (with an SSpeed of 0.845), but rather that the solution was found in a smaller percentage of the maximum number of iterations allowed for that dimension.

From the results, it can be seen that the Spherical problem did not become harder for the PSO algorithm as the dimensions increased. In contrast, the PSO algorithm found it harder to minimize the multimodal Ackley function as the dimensions increased. In 30 dimensions, the algorithm was only able to find the optimal solution in 9 out of the 30 runs (SRate of 0.3).

In many cases the QMetric and SRate have the same value. This indicates that the quality of the solutions and the success rate are essentially reflecting the same information. If the quality of the solution is low, it will result in a QMetric value of 0 and this will also be reflected in an unsuccessful run. There are cases, however, where the QMetric differs from the SRate measure. Consider for example the Rosenbrock function in 15 dimensions. None of the 30 runs of the PSO algorithm found the optimal solution, but a QMetric value of 0.888 indicates that the best solutions found were still of a relatively good quality (relatively close to the global optimum fitness value).

It is interesting to notice how in the case of the Griewank benchmark function, the performance metrics are high in 1 dimension, decrease in 10 dimensions, but then increase in 15 dimensions and increase further in 30 dimensions. This indicates that the problem becomes easier in higher dimensions, which is a known characteristic of the Griewank benchmark function [34].

The three normalized metrics proposed in this section make it possible to compare data from different benchmarks and algorithms. The following section discusses the challenge of extracting features of problems with the aim of finding a mapping from feature to performance space.

4.4 Feature Extraction for Continuous Optimisation Problems

Part of the challenge of solving the algorithm selection / performance prediction problem for continuous optimisation lies in finding a set of features of problems that can be used as the basis for mapping from feature space to algorithm performance. Given the fitness function of an optimisation problem, the aim is to develop a *problem characteriser* blackbox, as illustrated in Figure 4.4, that can produce as output a number of characteristics of the problem based on analysis of fitness landscapes of the problem.

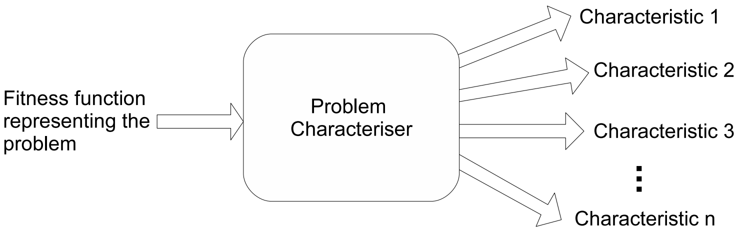


Fig. 4.4 A *problem characteriser* blackbox that can take as input a real-encoded fitness function and produce as output a number of approximate characteristics of the problem, based on analysis of fitness landscapes

To be applicable to the algorithm selection problem, each technique within the *problem characteriser* should have the following features:

(1) Each technique should measure characteristics that in some way correlate with performance of algorithms. Without some correlation, a mapping from feature space to performance space (as illustrated in Figure 4.1) will not be achievable.

(2) For the technique to be useful on unseen problems, it is assumed that there is no information on the nature of the problem beforehand other than the fitness function and the domains of the variables of the problem. For example, a technique that requires knowledge of the global optima would not be appropriate.

(3) Each technique should be analytical and result in numerical output values to facilitate automated analysis of generated data.

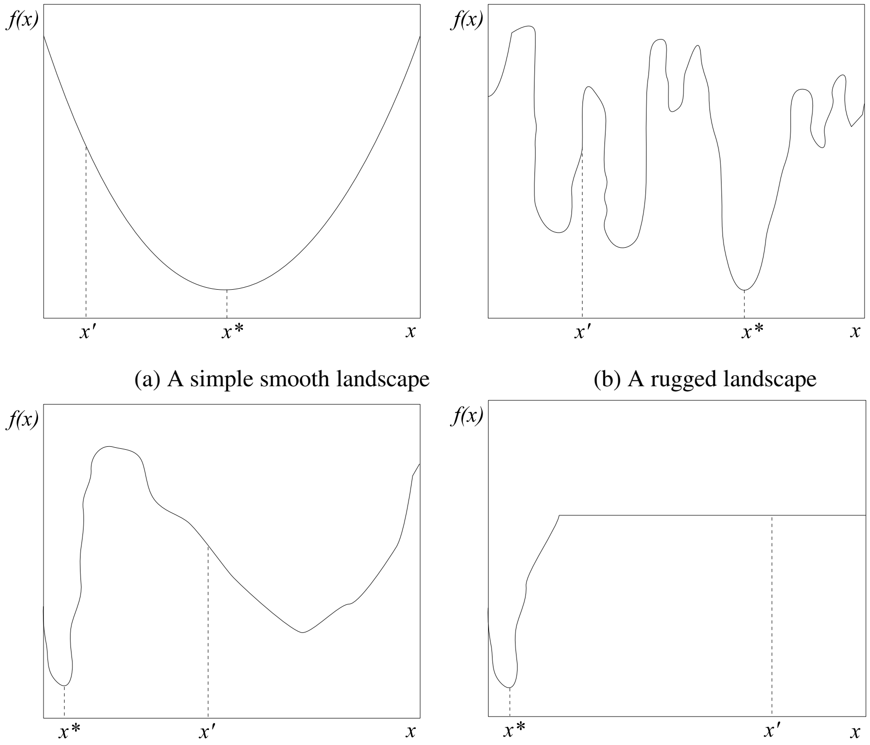
(4) The computational work required in executing the technique should be significantly less than the computational work required to solve the problem using a typical search algorithm. In other words, characterising a problem should be less computationally intensive than solving the problem with multiple algorithms using a trial and error approach.

Although requirement 4 above states that the numerical effort of probing and characterising a problem in multiple ways should be significantly less than the numerical effort in using a trial and error approach with multiple algorithms, one could argue that this is not an essential feature. A trial and error approach to solving an unknown problem has no guarantee of producing a good solution to the problem. On the other hand, characterising a problem should lead to a deeper understanding of the problem and better choices of algorithms and therefore have an increased chance of producing a solution of higher quality than the uninformed application of multiple search algorithms.

The following section discusses some of the features of fitness landscapes that could contribute to problem hardness and in Section 4.4.2 three specific techniques are described as examples of the kinds of measures that could be used to partly characterise problems.

4.4.1 What Makes an Optimisation Problem Hard?

What are the properties or features of optimisation problems that could determine the degree of difficulty in solving them? Consider the fitness landscapes of simple one-dimensional continuous problems as illustrated in Figure 4.5, where x' is a candidate solution found by a search process and x^* is the global optimum solution. A search algorithm would use information from the search space to decide how to proceed – information such as the gradient of the fitness function, or fitness values of solutions in the neighbourhood of x', or a whole population of alternative solutions with associated fitness values. Whichever approach is used, a simple function such as the one illustrated in Figure 4.5a, clearly provides good information to guide search towards the global optimum. In contrast, the rugged landscape in Figure 4.5b provides very little useful information to guide search towards the global optimum. In addition, the vertical gradients in places (e.g. at point x') make it difficult for some search algorithms, either because the gradient is not defined or because the same solution can have multiple fitness values. Ruggedness clearly affects problem difficulty and many studies on problem hardness have focussed on ruggedness as the main determining factor [33, 39, 52]. The ruggedness of a fitness landscape is, however, not the only factor affecting problem hardness. Consider for example the problem in Figure 4.5c. This landscape would not be regarded as rugged compared with the landscape in Figure 4.5b, but presents misleading information for a local search algorithm. Starting at position x', a local search algorithm would typically be guided away from the global optimum at x^*. Problems such as these that present an algorithm with misleading information are known as deceptive problems and many studies on problem hardness have focussed on deception as the main determining factor [5, 18, 19, 27]. Neutrality is yet another factor that can have an influence on problem difficulty [59, 60, 63, 55, 41]. This phenomenon is illustrated in Figure 4.5d, where there is a lack of information around the candidate solution x' for guiding search towards the global optimum.



(a) A simple smooth landscape

(b) A rugged landscape

(c) A deceptive landscape

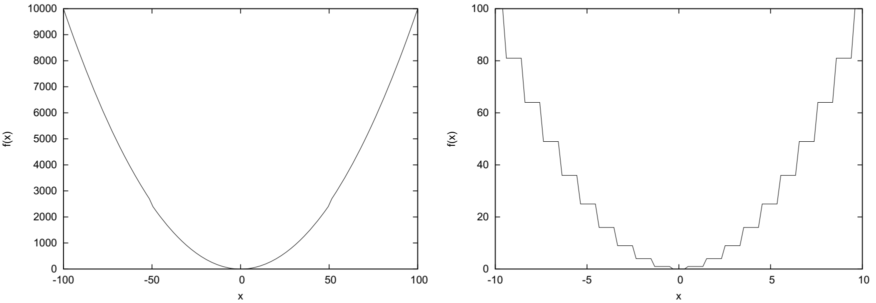
d) A neutral landscape

Fig. 4.5 Simple one-dimensional fitness landscapes to be minimised where x' is a candidate solution to the problem, $f(x')$ the fitness of solution x' and x^* the optimal solution

It is important to note that features such as ruggedness, deception or neutrality are not features of a fitness function, but rather features of a fitness landscape. The same fitness function can generate many different landscapes depending on the way in which the function is sampled or explored by a search algorithm. Consider for example the Step benchmark function in D dimensions:

$$f(\mathbf{x}) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2 \,. \tag{4.5}$$

The same function in one dimension is plotted in Figure 4.6 at different resolutions, rendering a seemingly smooth landscape in the case of Figure 4.6a and a landscape with high neutrality (many flat sections) in Figure 4.6b. In a similar way, a search process that samples the Step function by taking bigger step sizes may result in a smooth landscape, whereas a search process that samples the Step function by taking smaller steps may result in a landscape with high neutrality. This relationship between fitness functions and landscapes was highlighted by Jones [28] for combinatorial problems where he emphasized that different search operators on the same problem generate different landscapes. Therefore, viewing a continuous problem at different levels of granularity, or exploring a multi-dimensional space in different ways, can lead to very different fitness landscapes and hence potentially different views on features such as ruggedness, deception or neutrality.



(a) Larger domain

b) Smaller domain

Fig. 4.6 One-dimensional Step benchmark function viewed in at different resolutions rendering two different fitness landscapes

This section discussed three commonly used landscape features, namely ruggedness, deception and neutrality, as features that can affect problem hardness. This list is not complete and there are many other features that can affect the hardness of problems. We know, however, that a complete characterisation of any problem in terms of hardness is computationally infeasible [21]. The aim is therefore to characterise a problem using as wide a range of features as possible, while still maintaining a relatively low computational budget.

4.4.2 Possible Techniques for Feature Extraction

There are a large number of proposed techniques for fitness landscape analysis. Many of these techniques were published in literature as problem hardness measures and for many of these, subsequent publications can be found providing evidence of counter-examples for which the proposed measure of hardness does not hold. Although many proposed techniques have therefore been 'shot down' as difficulty measures, it is proposed that some techniques can still be useful if viewed as measures of particular problem characteristics, rather than of problem difficulty. With a shift in focus away from predicting hardness, each technique has its own place within a set of techniques for characterising problems. Three possible techniques for characterising continuous optimisation problems are briefly described in this section and in the next section the features are investigated alongside performance metrics for a PSO algorithm.

4.4.2.1 First Entropic Measure of Ruggedness

Weinberger [65] proposed using an autocorrelation method for characterising the ruggedness of a fitness landscape. His technique, and later variations [25, 39, 51], have been widely used to characterise the ruggedness of discrete landscapes, but have had limited success in predicting problem difficulty because they do not consider neutrality in the landscape [13, 44, 63]. An alternative approach to analysing the ruggedness of landscapes with respect to neutrality using information entropy, called the First Entropic Measure (FEM), was proposed by Vassilev *et al.* [62, 61, 63]. Based on a random walk, a sequence of three-point objects are generated. These objects are classified as rugged, smooth or neutral, based on the change in fitness values between neighbouring points. The ruggedness of the landscape is estimated using a measure of entropy with respect to the probability distribution of the rugged elements within the sequence. The result is a graph illustrating how ruggedness changes with an increase in landscape neutrality. Malan and Engelbrecht [37] proposed adaptations to this approach for continuous domains and defined a single measure of ruggedness in the range $[0,1]$ where a higher value is indicative of more ruggedness in the landscape. This single measure of ruggedness is referred to as FEM.

4.4.2.2 Fitness Distance Correlation

Jones and Forrest [29] introduced fitness distance correlation (FDC) as a way of predicting the performance of a genetic algorithm on problems with known global optima. The basic premise of FDC is that for a landscape to be easy to search, fitness values should increase as distance to the optimum decreases in the case of maximisation problems and decrease as distance to the optimum decreases in the case of minimisation problems. In continuous domains, the Euclidean metric can be used for the calculation of distances. For FDC to be applicable as a feature extraction technique for the algorithm selection / prediction problem, it must be adapted to be used without knowledge of the global optima. For this reason, it is proposed that the calculation of FDC be based on a fittest point from a sample in the place of a global optimum. Instead of estimating how well or badly the problem guides search towards the optimum, the modified FDC quantifies how well or badly the problem guides search towards a place of better fitness. This is equivalent to a shift in focus from measuring difficulty to measuring evolvability (the concept of evolvability is described in [2, 58]). To emphasize this difference, FDC based on the fittest point from a sample is referred to as FDC$_e$ (FDC evolvability metric). More formally, given a sample of n points, with associated fitness values for each point $F = f_1,\ldots,f_n$, the fittest point in the sample is determined. Distances $D = d_1,\ldots,d_n$

are calculated from each point in the sample to the fittest point in the sample. FDC$_e$ is calculated as the covariance of F and D divided by the product of the standard deviation of F and standard deviation of D:

$$FDC_e = \frac{\mathrm{Cov}(F,D)}{\sigma(F)\sigma(D)} \,. \tag{4.6}$$

The FDC$_e$ measure takes on values from –1 (perfect anti-correlation) to +1 (perfect correlation), where low values are regarded as desirable for maximisation problems and high values desirable for minimisation problems.

4.4.2.3 Dispersion Metric

A funnel in a landscape is a global basin shape that consists of clustered local optima [54]. The Rastrigin benchmark function illustrated in Figure 4.2a is an example of a single funnel landscape, because although the landscape is multimodal, there is an underlying unimodal global structure. In contrast, Schwefel 2.26 in Figure 4.2b is multi-funnelled. Multi-funnel landscapes can present problems for search, particularly in the case of algorithms that rely on local information, as they may become trapped in sub-optimal funnels [54, 68]. A technique for estimating the presence of funnels in a fitness landscape is Lunacek and Whitley's dispersion metric[36]. Given a sample of points below a fitness threshold, if a decrease in threshold (assuming a minimisation problem) results in an increase in the dispersion of the points from the sample that are below the threshold (in solution space), then this indicates the presence of multiple funnels in the landscape. More precisely, the approach used in this chapter to calculate the dispersion metric is as follows:

(1) Draw a uniformly random sample S of 1000 points (position vectors) from the n-dimensional search space.
(2) Normalise the position vectors in S to produce set S', so that the domain of the search space is $[0,1]$ for all n.
(3) Calculate the dispersion of set S', $disp(S')$, as the average pair-wise distance between points in S'.
(4) Determine the subset S^* of S', such that S^* consists of the fittest 10% of the points in S'.
(5) Calculate $disp(S^*)$ as the average pair-wise distance between points in S^*.
(6) Calculate the dispersion metric, $DM = disp(S^*) - disp(S')$.

The dispersion metric (referred to as DM) is a single value, where a positive value indicates the presence of funnels.

4.4.3 Features and Algorithm Performance

This section presents values for feature metrics FEM, FDC$_e$ and DM alongside performance metrics (based on a traditional PSO algorithm) for a sample of benchmark functions and dimensions. The purpose is to show how fitness landscape metrics could be analysed alongside performance metrics. Although some preliminary observations are made regarding the predictive value of the sample metrics for PSO performance, further investigation is needed before any general conclusions can be drawn.

4.4.3.1 Benchmark Problems

Data was generated for each benchmark problem listed in Table 4.2 for dimensions 1, 2, 5, 15 and 30. These functions cover a range of characteristics. All functions are multimodal, except for Spherical, Quadric and Rosenbrock for dimensions 1 to 3. (Note that although the Rosenbrock function is widely stated as unimodal, it has been shown to be multimodal for dimensions of 4 and higher [48].) Functions Griewank, Quadric, Rana, Rosenbrock and Salomon are non-separable. In addition Rana and Schwefel 2.26 are multi-funnelled.

4.4.3.2 Experimental Setup

For the FEM measure, the approach described in [37] was used with the total number of points sampled through random walks equal to $1000 \times D$. For the FDC$_e$ calculations, uniform random samples of $500 \times D$ were used (reduced to compensate for the computational load of calculating the distance to the best point for each sampled solution). The DM measure was based on uniform random samples of 1000 points with the dispersion difference based on the 100 (10%) fittest solutions from the sample. For the performance metrics a traditional gbest PSO algorithm [10] was used to solve each problem with 50 particles, 1.496 for both the cognitive and social acceleration constants and 0.7298 for the inertia weight [11]. The terminating condition was the maximum number of function evaluations and was set to $10000 \times D$, where D is the dimension of the problem. Note that the computational budget given to solving a problem is approximately 10 times the computational budget given to probing the same problem to calculate the FEM and FDC$_e$ fitness landscape features. The computation of the DM metric involves calculating the pairwise distance between 100 points (the pairwise distance between the full sample of 1000 points is not part of the computational load, as it can be pre-calculated for each dimension, since the distances are normalised), but does not increase as the dimension increases.

4.4.3.3 Landscape Metric Results

Results are listed in Table 4.4. All fitness landscape metric values are reported as means of 30 independent runs and the standard deviations are shown in brackets. The three landscape metrics are discussed briefly in this section in terms of the predictability and reliability of the results generated.

The FEM metric is an estimation of landscape ruggedness and is a value in the range $[0,1]$, where 1 indicates maximal ruggedness. Considering the values of FEM in Table 4.4, Ackley and Salomon have the highest values, while functions like Quadric and Rosenbrock have lower values, which is as expected. Note that the

Table 4.4 Benchmark functions and dimension (D) with feature metrics (FEM, DM, FDC$_e$) and performance metrics based on a traditional PSO algorithm

Function	D	FEM	DM	FDC$_e$	QMetric	SRate	SSpeed
Ackley	1	0.876 (±0.009)	-0.296 (±0.003)	0.793 (±0.008)	1.000	1.000	0.471
Ackley	2	0.854 (±0.008)	-0.357 (±0.008)	0.774 (±0.010)	1.000	1.000	0.634
Ackley	5	0.865 (±0.004)	-0.333 (±0.017)	0.701 (±0.025)	1.000	1.000	0.782
Ackley	15	0.870 (±0.002)	-0.288 (±0.018)	0.506 (±0.025)	0.933	0.933	0.855
Ackley	30	0.870 (±0.001)	-0.270 (±0.018)	0.431 (±0.018)	0.300	0.300	0.862
Griewank	1	0.786 (±0.011)	-0.289 (±0.003)	0.967 (±0.002)	1.000	1.000	0.721
Griewank	2	0.665 (±0.043)	-0.358 (±0.011)	0.966 (±0.007)	0.915	0.767	0.669
Griewank	5	0.479 (±0.042)	-0.358 (±0.012)	0.903 (±0.034)	0.560	0.067	0.211
Griewank	15	0.348 (±0.010)	-0.338 (±0.013)	0.653 (±0.024)	0.691	0.100	0.910
Griewank	30	0.292 (±0.007)	-0.328 (±0.015)	0.567 (±0.027)	0.902	0.367	0.921
Quadric	1	0.464 (±0.014)	-0.297 (±0.003)	0.968 (±0.002)	1.000	1.000	0.942
Quadric	2	0.505 (±0.058)	-0.329 (±0.010)	0.648 (±0.012)	1.000	1.000	0.893
Quadric	5	0.447 (±0.038)	-0.213 (±0.019)	0.342 (±0.030)	1.000	1.000	0.906
Quadric	15	0.368 (±0.018)	-0.087 (±0.021)	0.115 (±0.013)	1.000	1.000	0.869
Quadric	30	0.320 (±0.012)	-0.058 (±0.021)	0.071 (±0.010)	1.000	1.000	0.711
Rana	2	0.430 (±0.082)	0.051 (±0.014)	0.017 (±0.063)	0.187	0.000	0.000
Rana	5	0.679 (±0.011)	0.069 (±0.017)	0.012 (±0.035)	0.000	0.000	0.000
Rana	15	0.711 (±0.006)	0.044 (±0.021)	0.008 (±0.019)	0.000	0.000	0.000
Rana	30	0.741 (±0.003)	0.032 (±0.021)	0.005 (±0.011)	0.000	0.000	0.000
Rastrigin	1	0.540 (±0.012)	-0.212 (±0.010)	0.708 (±0.015)	1.000	1.000	0.814
Rastrigin	2	0.588 (±0.027)	-0.224 (±0.015)	0.641 (±0.062)	1.000	1.000	0.792
Rastrigin	5	0.601 (±0.012)	-0.239 (±0.016)	0.499 (±0.077)	0.533	0.533	0.731
Rastrigin	15	0.601 (±0.008)	-0.232 (±0.014)	0.393 (±0.042)	0.000	0.000	0.000
Rastrigin	30	0.586 (±0.005)	-0.227 (±0.016)	0.358 (±0.023)	0.000	0.000	0.000
Rosenbrock	2	0.356 (±0.054)	-0.220 (±0.015)	0.546 (±0.022)	1.000	1.000	0.843
Rosenbrock	5	0.467 (±0.027)	-0.311 (±0.014)	0.687 (±0.063)	0.931	0.067	0.321
Rosenbrock	15	0.416 (±0.010)	-0.280 (±0.014)	0.555 (±0.081)	0.888	0.000	0.000
Rosenbrock	30	0.361 (±0.007)	-0.273 (±0.014)	0.477 (±0.074)	0.482	0.000	0.000
Salomon	1	0.887 (±0.008)	-0.277 (±0.005)	0.971 (±0.002)	1.000	1.000	0.574
Salomon	2	0.888 (±0.005)	-0.346 (±0.007)	0.960 (±0.009)	1.000	1.000	0.592
Salomon	5	0.890 (±0.004)	-0.350 (±0.016)	0.872 (±0.048)	0.000	0.000	0.000
Salomon	15	0.889 (±0.002)	-0.324 (±0.011)	0.627 (±0.037)	0.000	0.000	0.000
Salomon	30	0.886 (±0.001)	-0.318 (±0.013)	0.534 (±0.017)	0.000	0.000	0.000
Schwefel 2.26	1	0.483 (±0.013)	-0.003 (±0.030)	0.317 (±0.032)	1.000	1.000	0.816
Schwefel 2.26	2	0.539 (±0.028)	0.035 (±0.018)	0.300 (±0.057)	0.967	0.967	0.819
Schwefel 2.26	5	0.567 (±0.018)	0.038 (±0.017)	0.171 (±0.113)	0.400	0.400	0.826
Schwefel 2.26	15	0.582 (±0.010)	0.021 (±0.017)	0.080 (±0.078)	0.000	0.000	0.000
Schwefel 2.26	30	0.582 (±0.005)	0.024 (±0.022)	0.065 (±0.043)	0.000	0.000	0.000
Spherical	1	0.466 (±0.014)	-0.296 (±0.002)	0.968 (±0.001)	1.000	1.000	0.845
Spherical	2	0.550 (±0.066)	-0.358 (±0.010)	0.971 (±0.003)	1.000	1.000	0.890
Spherical	5	0.474 (±0.026)	-0.355 (±0.011)	0.900 (±0.031)	1.000	1.000	0.916
Spherical	15	0.350 (±0.009)	-0.338 (±0.010)	0.666 (±0.024)	1.000	1.000	0.941
Spherical	30	0.292 (±0.006)	-0.328 (±0.014)	0.566 (±0.016)	1.000	1.000	0.940
Step	1	0.516 (±0.007)	-0.293 (±0.003)	0.966 (±0.002)	1.000	1.000	0.994
Step	2	0.698 (±0.025)	-0.359 (±0.010)	0.967 (±0.005)	1.000	1.000	0.988
Step	5	0.813 (±0.013)	-0.357 (±0.015)	0.894 (±0.038)	1.000	1.000	0.977
Step	15	0.707 (±0.011)	-0.336 (±0.011)	0.665 (±0.045)	1.000	1.000	0.972
Step	30	0.619 (±0.010)	-0.330 (±0.012)	0.564 (±0.014)	0.924	0.900	0.857

FEM value for Griewank decreases as the dimension increases, which is consistent with the explanation that Griewank becomes a ‘simpler’ function with increased dimension [34]. However, some values of FEM are not quite as expected. For example, Spherical in 2 dimensions has an FEM value of 0.550, which is higher than the FEM value of Schwefel 2.26 in 2 dimensions. For many of the functions, the standard deviation of the FEM mean is higher in 2 dimensions than in other dimensions. These anomalies affect the reliability of the FEM metric and would need to be investigated further.

The DM metric is an estimate of the presence of funnels in the fitness landscape. Negative values for DM indicate a simpler global topology, while larger values (positive values) are indicative of multi-funnels. All values for DM in Table 4.4 are negative, except for Rana and Schwefel 2.26 (above 1 dimension), which are the only multi-funnelled benchmark functions. The metric DM therefore seems to predict the presence of funnels fairly well. Of concern are the relatively high standard deviations in some cases. For example, the standard deviation of the DM measure for Schwefel 2.26 in 1 dimension implies that the DM measure could very likely result in a positive value for a different set of 30 runs.

The FDC$_e$ metric quantifies how well distances to the fittest solution in a sample are correlated with fitness values. In the case of a simple smooth landscape, like Spherical, the fitness values should decrease steadily as the distance to fitter solutions decrease. FDC$_e$ is therefore a measure of evolvability for a local search algorithm, where higher values indicate better evolvability. The values of FDC$_e$ in Table 4.4 range from values close to 1 for Spherical in low dimensions, to as low as 0.005 for Rana in 30 dimensions. For most functions the value of FDC$_e$ decreases as the dimension increases. This is most probably because the search space increases exponentially and so distances can also become very large. Further investigation is needed on whether the FDC$_e$ measure is a reasonable predictor of evolvability in higher dimensions.

Figure 4.7 shows the FDC$_e$ values plotted against the DM values for all problems in Table 4.4. The scatterplot shows that there is a strong negative correlation (Spearman correlation coefficient value of -0.789) between the two feature metrics. This shows that although the two metrics have different focus areas, they are capturing similar information: the presence of funnels (high DM values) seems to imply lower evolvability (low FDC$_e$ values), alternatively, lower evolvabilty could imply the presence of funnels. Further investigation into this relationship should be done to provide insight into the metrics and ways in which the approaches to calculating the metrics can be improved. Figure 4.8 plots the FDC$_e$ values against the FEM ruggedness values, showing a very weak correlation (Spearman correlation coefficient value of 0.169). This seems to indicate that the two metrics are capturing different information on the problems.

4.4.3.4 Correlation to Performance Metrics

For a feature metric to be useful it should show some correlation (or anti-correlation) to performance. However, as discussed in Section 4.4.2, no one technique can serve

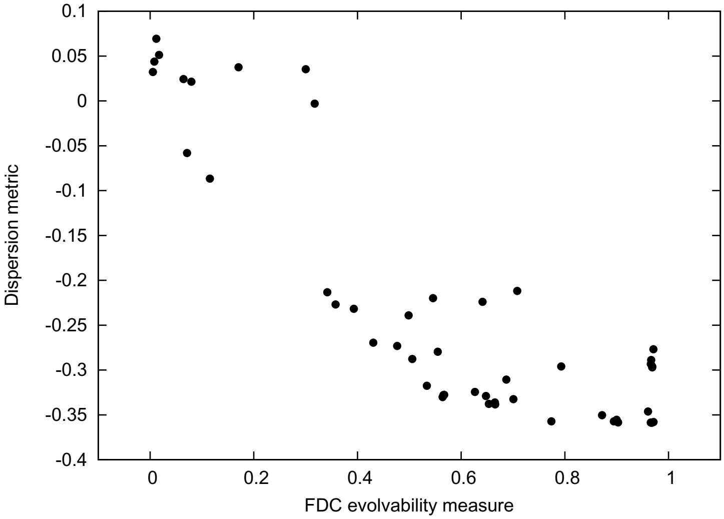


Fig. 4.7 A scatterplot of FDC$_e$ and associated dispersion metric (DM) values based on the data shown in Table 4.4

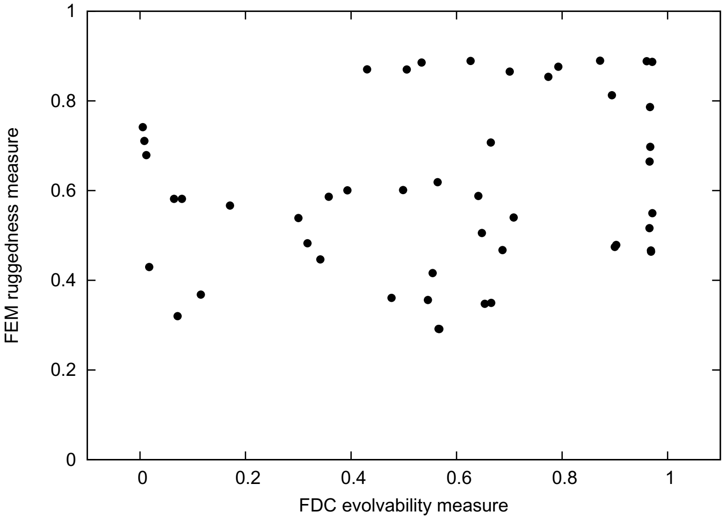


Fig. 4.8 A scatterplot of FDC$_e$ and associated FEM ruggedness values based on the data shown in Table 4.4

as a predictor of hardness. To illustrate this, consider Figure 4.9 showing a scatterplot of the FDC$_e$ and associated QMetric values. Recall that for minimization problems larger values of FDC$_e$ are indicative of higher evolvability and larger values of QMetric are indicative of better performance by the algorithm. Although the data shows only a moderate correlation (Spearman correlation coefficient value of 0.563), there does seem to be some value in FDC$_e$ as a part-predictor of performance. On the scatterplot in Figure 4.9, one group of points on the bottom right that show no positive correlation, correspond to the Salomon benchmark function for dimensions 5 and higher. These points have a minimum QMetric value of 0 (indicating the algorithm's failure), but also have relatively high FDC$_e$ values. In the case of the Salomon function, FDC$_e$ is therefore a misleading predictor of difficulty. The data in Table 4.4, however, shows that the FEM ruggedness values of Salomon are particularly high. In addition, there are other factors that could affect difficulty that should be considered. For example, the Salomon function has extremely steep gradients, which could be a factor contributing to the difficulty for a PSO algorithm [38].

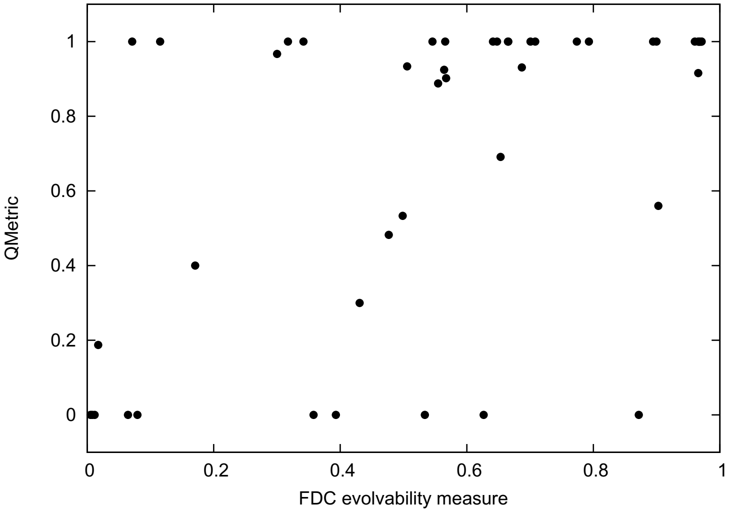


Fig. 4.9 A scatterplot of FDC$_e$ and associated QMetric values based on the data shown in Table 4.4

The scatterplot in Figure 4.9 has a large proportion of values at the top and at the bottom with a few points scattered in between. This is indicative of distinct groups of problems based on success or failure of the algorithm in solving the problem. Figure 4.10 shows a different visualisation of the link between FDC$_e$ and algorithm performance, where the problems are plotted by dimension and the performance is discretised into four groups:

- *Always solved*: problems with an SRate of 1, indicating that the solution was found for all 30 runs of the PSO algorithm.
- *Sometimes solved*: problems with an SRate less than 1, but greater than 0, indicating that the solution was found for some of the runs.
- *Almost solved*: problems with an SRate of 0, but a QMetric value greater than 0, indicating that although none of the runs found the solution to within the required fixed accuracy level, a solution was sometimes found that was very close to the optimum.
- *Not solved*: problems with all performance metric values equal to 0.

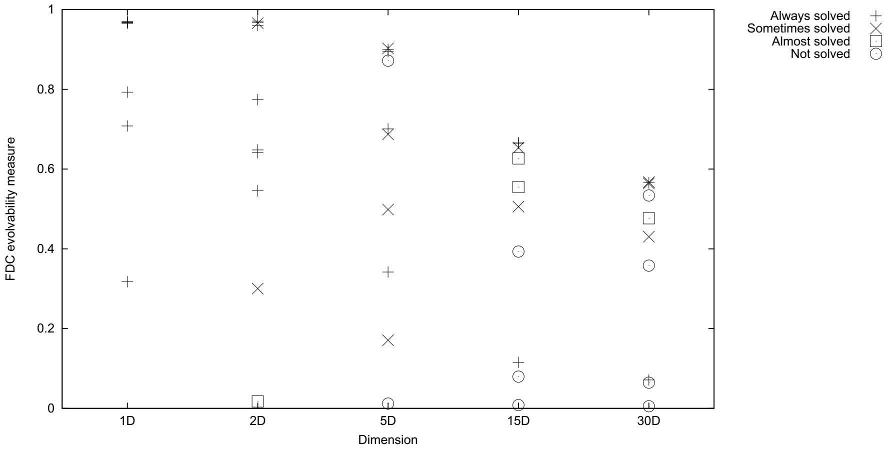


Fig. 4.10 Discretised performance plotted against FDC$_e$ values based on the data shown in Table 4.4

It can be seen from Figure 4.10 that there is some value in the FDC$_e$ metric as a predictor of PSO performance. Although there are exceptions, in each dimension column the symbols tend to match the order of the symbols in the legend. For example, consider the 2D column: with the exception of one high cross, the plus signs (always solved) are above the cross (sometimes solved) and the square (almost solved). Similarly, in 15D and 30D, the circles are lower down in the columns. Figure 4.11 shows the same discretised performance values against DM values. Note that the legend is plotted in the opposite order as in Figure 4.10, since the symbols are expected to appear in that order if correlated with DM. Similarly to FDC$_e$, there seems to be some value in the dispersion metric as a predictor of PSO performance. It would seem, however, that DM is a slightly better predictor in higher dimensions (given the bigger gap between the circles and plus signs at high values of DM in 15 and 30 dimensions) and that FDC$_e$ is a slightly better predictor in lower dimensions.

In Figure 4.10 and 4.11 it can also be seen that all one-dimensional problems are always solved by the PSO algorithm. The difference in the performance of the

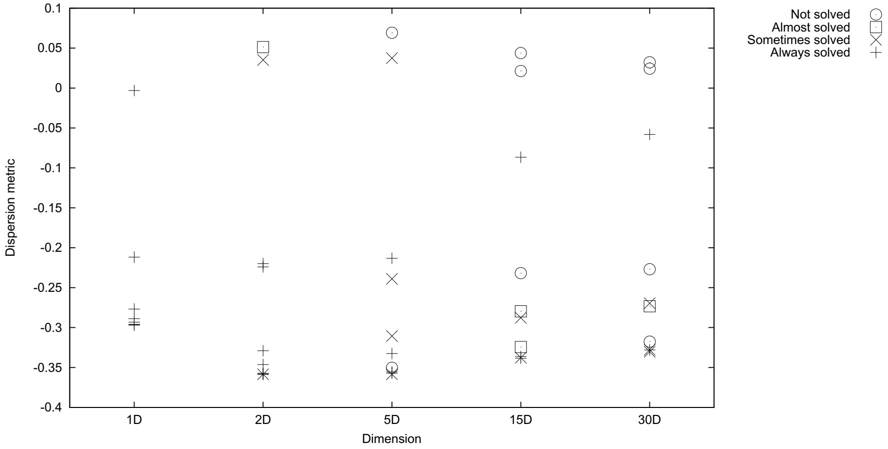


Fig. 4.11 Discretised performance plotted against dispersion metric (DM) values based on the data shown in Table 4.4

algorithm on these problems is in terms of the speed with which the algorithm finds the solution. This is captured in the SSpeed metric. It can be shown, however, that FDC$_e$ and DM show no correlation with the SSpeed metric for the given data. The feature metric which seems to show value in these cases is the FEM ruggedness measure. Figure 4.12 shows the relationship between the FEM metric and SSpeed of the subset of data entries from Table 4.4 that have a SRate value of 1. This figure shows that in the case of problems for which the PSO algorithm was able to find the global minimum, the ruggedness seems to have a moderate effect on the speed with which the PSO algorithm is able to find the optimum (Spearman correlation coefficient value of -0.406).

4.4.4 Discussion

The previous section presented results from three techniques for characterising fitness landscapes alongside performance metrics of a traditional PSO algorithm. One measure (FDC$_e$) focussed on quantifying evolvability for local search, one focussed on quantifying the ruggedness based on random walks and the third focussed on estimating the presence of funnels in the fitness landscape. On their own, these measures cannot be used to predict PSO performance. Each, however, provides some information that could form part of a complex multi-dimensional prediction problem. There are many more characteristics of problems that should be considered, such as neutrality, steepness of gradients and evolvability in relation to particular search operators. More work is needed in the design of suitable techniques and the adaptation of existing techniques for quantifying these and other fitness landscape characteristics.

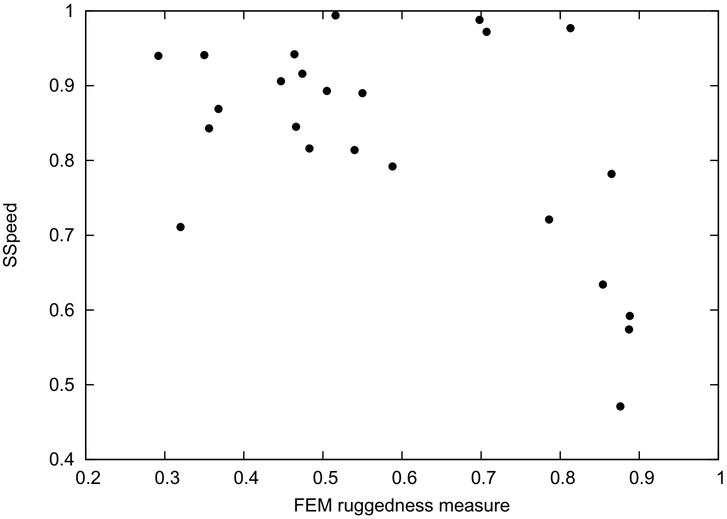


Fig. 4.12 A scatterplot of the FEM ruggedness metric and associated SSpeed values for data in Table 4.4 with SRate values of 1

What remains to be achieved in solving the performance prediction problem is finding a mapping from feature space to performance space. Given a good data set for training, this should be achievable using one of the many data mining techniques. Although it is not possible to measure how good a data set is, a much wider range of sample benchmark problems would go some of the way to achieving a better training set. This study considered one PSO algorithm. More work is also needed on different algorithms and algorithm variations, since a good set of features for predicting PSO performance might be an inappropriate set of features for some other algorithm. Solving the performance prediction problem for a range of different algorithms would not only be useful, but could ultimately lead to a better understanding of the behaviour of the algorithms themselves.

4.5 Conclusion

This chapter discussed the algorithm selection and performance prediction problems for metaheuristics and continuous optimisation. One of the requirements of the model is the existence of suitable metrics for evaluating the performance of algorithms on known problems to generate suitable data for training. New normalized metrics were proposed that can be used to compare performance of algorithms on problems with vastly different fitness ranges. The model also requires suitable metrics for characterising the features of problems. Three metrics were investigated in relation to a traditional PSO algorithm. A number of issues were highlighted regarding the reliability of measures and the usefulness as predictors of PSO performance. The conclusion is that feature metrics have to be viewed together as a multi-dimensional prediction problem and that a wide range of different features are needed to properly characterise problems.

References

[1] Ahn, C.W., Ramakrishna, R.S.: On the Scalability of Real-Coded Bayesian Optimization Algorithm. IEEE Trans. Evol. Comp. 12(3), 307–322 (2008)

[2] Altenberg, L.: The Evolution of Evolvability in Genetic Programming. In: Kinnear, K. (ed.) Advances in Genetic Programming, pp. 47–74. MIT Press, Cambridge (1994)

[3] Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Oxford University Press, Bristol (1997)

[4] Bilchev, G., Parmee, I.C.: The Ant Colony Metaphor for Searching Continuous Design Spaces. In: Fogarty, T.C. (ed.) AISB-WS 1995. LNCS, vol. 993, pp. 25–39. Springer, Heidelberg (1995)

[5] Borenstein, Y., Poli, R.: Information Landscapes and Problem Hardness. In: GECCO 2005: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 1425–1431. ACM Press, New York (2005)

[6] Cerny, V.: Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. J. Optim. Theory Appl. 45(1), 41–51 (1985)

[7] Chen, T., Tang, K., Chen, G., Yao, X.: Analysis of Computational Time of Simple Estimation of Distribution Algorithms. IEEE Trans. Evol. Comp. 14(1), 1–22 (2010)

[8] Chen, T., Tang, K., Chen, G., Yao, X.: A Large Population Size Can Be Unhelpful in Evolutionary Algorithms. Theor. Comput. Sci. 436, 54–70 (2012)

[9] Dorigo, M.: Optimization, Learning and Natural Algorithms. Ph.D. thesis, Politecnico di Milano, Italy (1992) (in Italian)

[10] Eberhart, R., Kennedy, J.: A New Optimizer using Particle Swarm Theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, pp. 39–43 (1995)

[11] Eberhart, R., Shi, Y.: Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, pp. 84–88 (2000)

[12] Eiben, A.E., Jelasity, M.: A Critical Note on Experimental Research Methodology In EC. In: Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), pp. 582–587. IEEE Press (2002)

[13] Fontana, W., Stadler, P.F., Bornberg-Bauer, E.G., Griesmacher, T., Hofacker, I.L., Tacker, M., Tarazona, P., Weinberger, E.D., Schuster, P.: RNA Folding and Combinatory Landscapes. Phys. Rev. E 47, 2083–2099 (1993)

[14] Gandomi, A.H., Alavi, A.H.: Krill herd: A New Bio-inspired Optimization Algorithm. Commun. Nonlinear Sci. Numer. Simul. 17(12), 4831–4845 (2012)

[15] Geem, Z.W., Kim, J.H., Loganathan, G.V.: A New Heuristic Optimization Algorithm: Harmony Search. Simulation 76(2), 60–68 (2001)

[16] Glover, F.: Tabu Search – Part I. INFORMS J. Comput. 1(3), 190–206 (1989)

[17] Glover, F.: Tabu Search – Part II. INFORMS J. Comput. 2(1), 4–32 (1990)

[18] Goldberg, D.E.: Simple Genetic Algorithms and the Minimal Deceptive Problem. In: Davis, L. (ed.) Genetic Algorithms and Simulated Annealing, ch. 6, pp. 74–88. Pitman, London (1987)

[19] Goldberg, D.E.: Genetic Algorithms and Walsh Functions: Part II, Deception and Its Analysis. Complex Sys. 3, 153–171 (1989)

[20] Guo, H., Hsu, W.H.: GA-Hardness Revisited. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 1584–1585. Springer, Heidelberg (2003)

[21] He, J., Reeves, C., Witt, C., Yao, X.: A Note on Problem Difficulty Measures in Black-Box Optimization: Classification, Realizations and Predictability. Evol. Comput. 15(4), 435–443 (2007)

[22] He, J., Yao, X.: A Study of Drift Analysis for Estimating Computation Time of Evolutionary Algorithms. Nat. Comput. 3(1), 21–35 (2004)

[23] Herrera, F., Lozano, M., Molina, D.: Continuous Scatter Search: An Analysis of the Integration of Some Combination Methods and Improvement Strategies. Eur. J. Oper. Res. 169(2), 450–476 (2006)

[24] Herrera, F., Lozano, M., Verdegay, J.L.: Tackling Real-coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. Artif. Intell. Rev. 12(4), 265–319 (1998)

[25] Hordijk, W.: A Measure of Landscapes. Evol. Comput. 4(4), 335–360 (1996)

[26] Jansen, T.: On Classifications of Fitness Functions. In: Kallel, L., Naudts, B., Rogers, A. (eds.) Theoretical Aspects of Evolutionary Computing, pp. 371–385. Springer, London (2001)

[27] Jelasity, M., Tóth, B., Vinkó, T.: Characterizations of Trajectory Structure of Fitness Landscapes Based on Pairwise Transition Probabilities of Solutions. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, pp. 623–630. IEEE Press (1999)

[28] Jones, T.: Evolutionary Algorithms, Fitness Landscapes and Search. Phd thesis, The University of New Mexico (1995)

[29] Jones, T., Forrest, S.: Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. In: Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 184–192. Morgan Kaufmann (1995)

[30] Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1942–1948. IEEE Press (1995)

[31] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)

[32] Lee, C.Y., Yao, X.: Evolutionary Programming using Mutations based on the Levy Probability Distribution. IEEE Trans. Evol. Comput. 8(1), 1–13 (2004)

[33] Lipsitch, M.: Adaptation on Rugged Landscapes generated by Iterated Local Interactions of Neighboring Genes. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the 4th International Conference on Genetic Algorithms, pp. 128–135. Morgan Kaufmann, San Diego (1991)

[34] Locatelli, M.: A Note on the Griewank Test Function. J. Glob. Optim. 25, 169–174 (2003)

[35] Lourenço, H.R., Martin, O., Stützle, T.: Iterated Local Search. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 57, pp. 321–353. Kluwer Academic Publishers (2002)

[36] Lunacek, M., Whitley, D.: The Dispersion Metric and the CMA Evolution Strategy. In: GECCO 2006: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 477–484. ACM, New York (2006)

[37] Malan, K.M., Engelbrecht, A.P.: Quantifying Ruggedness of Continuous Landscapes using Entropy. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009, pp. 1440–1447 (2009)

[38] Malan, K.M., Engelbrecht, A.P.: Steep Gradients as a Predictor of PSO Failure. In: GECCO 2013: Proceedings of the Fifteenth International Conference on Genetic and Evolutionary Computation Conference Companion 2013, pp. 9–10 (2013)

[39] Manderick, B., de Weger, M.K., Spiessens, P.: The Genetic Algorithm and the Structure of the Fitness Landscape. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 143–150. Morgan Kaufmann (1991)

[40] Oliveto, P.S., He, J., Yao, X.: Analysis of the (1 + 1)-EA for Finding Approximate Solutions to Vertex Cover Problems. IEEE Trans. Evol. Comp. 13(5), 1006–1029 (2009)

[41] Owen, A., Harvey, I.: Adapting Particle Swarm Optimisation for Fitness Landscapes with Neutrality. In: IEEE Swarm Intelligence Symposium, SIS 2007, pp. 258–265 (2007)

[42] Price, K.V., Storn, R.M., Lampinen, J.A.: Appendix A.1: Unconstrained Uni-Modal Test Functions. In: Differential Evolution A Practical Approach to Global Optimization. Natural Computing Series, pp. 514–533. Springer, Berlin (2005)

[43] Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Natural Computing Series. Springer-Verlag New York, Inc., Secaucus (2005)

[44] Rand, W.M.: Controlled Observations of the Genetic Algorithm in a Changing Environment: Case Studies using the Shaky Ladder Hyperplane-defined Functions. Ph.D. thesis, University of Michigan, Ann Arbor, MI, USA, chair-Holland, John H. and Chair-Riolo, Rick L (2005)

[45] Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der Biologischen Evolution. Frommann-Holzboog (1973)

[46] Rice, J.R.: The Algorithm Selection Problem. Adv. Comput. 15, 65–118 (1976)

[47] Schwefel, H.P.: Evolutionsstrategie und numerische Optimierung. Ph.D. thesis, Technical University of Berlin (1975)

[48] Shang, Y.W., Qiu, Y.H.: A Note on the Extended Rosenbrock Function. Evol. Comput. 14, 119–126 (2006)

[49] Smith-Miles, K.: Towards Insightful Algorithm Selection for Optimisation using Meta-learning Concepts. In: IJCNN 2008: Proceedings of the IEEE Joint Conference on Neural Networks, pp. 4118–4124 (2008)

[50] Smith-Miles, K.A.: Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection. ACM Comput. Surv. 6, 1–6 (2008)

[51] Stadler, P.F.: Towards a Theory of Landscapes. In: Lopéz-Peña, R., Capovilla, R., García-Pelayo, R., Waelbroeck, H., Zertuche, F. (eds.) Complex Systems and Binary Networks, vol. 461, pp. 77–163. Springer, New York (1995)

[52] Stadler, P., Schnabl, W.: The Landscape of the Travelling Salesman Problem. Phys. Lett. A 161(4), 337–344 (1992)

[53] Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Tech. rep., Nanyang Technological University, Singapore (2005)

[54] Sutton, A.M., Whitley, D., Lunacek, M., Howe, A.: PSO and Multi-funnel Landscapes: How Cooperation might Limit Exploration. In: GECCO 2006: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 75–82. ACM, New York (2006)

[55] Verel, S., Collard, P., Tomassini, M., Vanneschi, L.: Neutral Fitness Landscape in the Cellular Automata Majority Problem. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 258–267. Springer, Heidelberg (2006)

[56] Talbi, E.G.: Metaheuristics: From Design to Implementation. John Wiley & Sons, Inc., Hoboken (2009)

[57] Tayarani, M.H., Akbarzadeh-Totonchi, M.R.: Magnetic Optimization Algorithms a New Synthesis. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, pp. 2659–2664 (2008)

[58] Turney, P.D.: Increasing Evolvability Considered as a Large-Scale Trend in Evolution. In: Proceedings of 1999 Genetic and Evolutionary Computation Conference Workshop Program (GECCO 1999 Workshop on Evolvability), pp. 43–46 (1999)

[59] Vanneschi, L., Pirola, Y., Collard, P.: A quantitative study of neutrality in GP boolean landscapes. In: GECCO 2006: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 895–902. ACM, New York (2006)

[60] Vanneschi, L., Tomassini, M., Collard, P., Vérel, S., Pirola, Y., Mauri, G.: A Comprehensive View of Fitness Landscapes with Neutrality and Fitness Clouds. In: Ebner, M., O'Neill, M., Ekárt, A., Vanneschi, L., Esparcia-Alcázar, A.I. (eds.) EuroGP 2007. LNCS, vol. 4445, pp. 241–250. Springer, Heidelberg (2007)

[61] Vassilev, V.K.: Fitness Landscapes and Search in the Evolutionary Design of Digital Circuits. Ph.D. thesis, Napier University (2000)

[62] Vassilev, V.K., Fogarty, T.C., Miller, J.F.: Information Characteristics and the Structure of Landscapes. Evol. Comput. 8(1), 31–60 (2000)

[63] Vassilev, V.K., Fogarty, T.C., Miller, J.F.: Smoothness, Ruggedness and Neutrality of Fitness Landscapes: from Theory to Application. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computing: Theory and Applications, pp. 3–44. Springer-Verlag New York, Inc (2003)

[64] Wegener, I.: Complexity Theory – Exploring the Limits of Efficient Algorithms. Springer, Berlin (2005)

[65] Weinberger, E.: Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference. Biol. Cybern. 63(5), 325–336 (1990)

[66] Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Search. Technical Report SFI-TR-95-02-010, Santa Fe Institute (February 1995), http://ideas.repec.org/p/wop/safiwp/95-02-010.html (accessed: November 7, 2011)

[67] Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Optimization. IEEE Trans. Evol. Comput. 1(1), 67–82 (1997)

[68] Xin, B., Chen, J., Pan, F.: Problem Difficulty Analysis for Particle Swarm Optimization: Deception and Modality. In: GEC 2009: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 623–630. ACM, New York (2009)

[69] Yang, X.S.: Firefly Algorithms for Multimodal Optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)

[70] Yang, X.S., Deb, S.: Cuckoo Search via Lévy Flights. In: World Congress on Nature & Biologically Inspired Computing, NaBIC 2009, pp. 210–214 (2009)

Chapter 5
Fitness Landscapes and Problem Difficulty in Evolutionary Algorithms: From Theory to Applications

Guanzhou Lu, Jinlong Li, and Xin Yao

Abstract. Above many successes of evolutionary algorithms in solving computationally hard optimisations problems, a major challenge in practice remains how to select/construct the best suited algorithm when solving a problem. The well-known no free lunch theorem rules out the possibility of developing one best algorithm generally suitable for solving all problems. Within the realm of algorithm selection in general, the problem becomes how can we characterise problem hardness with reference to evolutionary algorithms (EAs). For the first time, this chapter rigorously derives a problem hardness measure from a theoretical difficulty measure widely used in complexity theory of EAs. Furthermore, the proposed measure is applied to construct an offline optimisation algorithm and an online optimisation algorithm. On one hand, the measure is incorporated with a machine learning algorithm for parameter tuning and achieves powerful performance. On the other hand, an adaptive algorithm framework is proposed and shows promising results. We argue that the proposed measure is general, yet powerful as an indicator of EA-hardness, and contribute to the goal of constructing better suited algorithms for solving problems.

Guanzhou Lu · Xin Yao
Joint USTC-Birmingham Research Institute in Intelligent Computation and Its Applications
The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
e-mail: {g.lu,x.yao}@cs.bham.ac.uk

Jinlong Li
Joint USTC-Birmingham Research Institute in Intelligent Computation and Its Applications
The Nature Inspired Computation and Applications Laboratory (NICAL),
School of Computer Science and Technology,
University of Science and Technology of China, Hefei, 230027, China
e-mail: jlli@ustc.edu.cn

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_5, © Springer-Verlag Berlin Heidelberg 2014

5.1 Introduction

How do we determine if a problem is difficult or easy for a given search heuristic? An answer to this question was sought after in the field of evolutionary computation (EC) for over a decade. As of yet, no satisfactory answer has been found.

Many practical attempts have been made to define a measure of problem difficulty for evolutionary algorithms (EAs), for example fitness landscape analysis, fitness-distance correlation [17], correlation length and operator correlation [27], fitness variance [33], epistasis variance [8], amongst others. All of these measures aim to act as a useful difficulty measure generally suitable for all problems. However, they do not always describe problem difficulty correctly and counter-examples have been identified.

From theoretical perspectives, He et al. [13] rigorously define difficulty measures and propose a classification on realisations of difficulty measures. Assuming a worst-case perspective, for both approximate and exact measures, He et al. [13] have proven that realisations of predictive measures, i.e. polynomial-time implementations, do not exist unless $P = NP$ or $BPP = NP$. In other words, to find a useful difficulty measure in general is impossible. However, this does not prevent us from developing a useful measure to estimate the difficulty for a broad class of problem instances in practice.

In this chapter, for the first time we rigorously analyse the relationship between the escape probability and the expected running time, a difficulty measure widely used in complexity theory of EAs [32]. Based on the concept of escape probability, we develop the fitness-probability cloud (fpc) and accumulated escape probability (aep). We argue that aep is the first practical difficulty measure with a solid theoretical basis in complexity theory of EAs.

To build upon the existing work on fitness landscapes and problem difficulty measures, which mostly focus on analysing characteristics of problem structure and algorithm behaviours, we aim to directly apply the difficulty measure to construct better suited algorithms for a particular class of instances. Two distinctive classes of applications (offline and online) are identified for the proposed difficulty measure: The first is to incorporate the measure with a support vector machine (SVM) learning algorithm to automate the parameter tuning of EAs for solving the unique input output sequence problem (UIOP). The second is to build the difficulty measure as the core decision maker into an adaptive EA for online optimisations.

The remainder of this chapter is organised as follows. Section 5.2 briefly reviews previous work on fitness landscapes and problem difficulty measures. The theoretical results on escape probability and expected running time are presented in Section 5.3. Section 5.4 introduces the concept of a fitness-probability cloud and accumulated escape probability. Sections 5.5 and 5.6 present the applications of the difficulty measure in an offline and an online optimisation algorithm, respectively. Finally, we conclude in Section 5.7.

5.2 Background

For over a decade, a considerable amount of research efforts have been devoted to the studies of problem hardness with reference to EAs. The notion of fitness landscapes, originally proposed in [40], underlies a large body of work in the literature. It is generally agreed that the properties associated with fitness landscapes can indicate problem difficulty. For instance, the problem difficulty with reference to EAs has been described using concepts of ruggedness, neutrality [38] and information landscapes [3].

The concept of fitness landscapes provides intuitive explanations of problem hardness, however, it does not explicitly quantify difficulty as a numerical measure. Therefore, it is desirable to have one or more algebraic values to capture key characteristics of fitness landscapes. Along this line of consideration, a significant contribution was made by Jones [17] through the introduction of a measure called fitness distance correlation (fdc), which has been evaluated empirically on a large number of genetic algorithm (GA) and genetic programming (GP) benchmarks showing considerable effectiveness. However, fdc still has some drawbacks with the most severe one being that the global optima have to be known beforehand, preventing fdc from being applied to real-world problems. This limitation has been overcome by the introduction of the fitness cloud (fc) [5] and the negative slope coefficient (nsc) [37] as a measure based on fc. Unfortunately fc also has its own weakness: experimental analysis has shown that *nsc* is drastically influenced by the neighbourhood sample size K in generating it. In practice, *nsc* is unable to serve as an accurate measure unless an appropriate K is selected [24].

5.3 Escape Probability versus Expected Runtime Time

This work aims to address the problem of developing a useful difficulty measure in both theory and practice. In the context of EAs, a predictive difficulty measure is thought to be useful if it can discriminate the hardness of a particular problem with respect to different EAs, which is needed in choosing the best suited algorithm to solve the problem. Apart from many existing practical measures mentioned above, there exists a number of studies into problem hardness in complexity theory [32], a widely used one being the expected running time of the algorithm on the fitness function, which is usually taken as a measure of difficulty of the fitness function for the algorithm.

The notion of escape probability (escape rate) was used by Merz [30] in analysing the use of random walks to escape from the basins of attractions of current local optima. In theoretical runtime analysis of EAs, He and Yao [14] proposed an analytic way to estimate the mean first hitting time of an absorbing Markov chain, in which the transition probability between states were used.

Below we define the concept of the escape probability and carry out a rigorous analysis of the relationship between the escape probability and the expected running time. The results show that the escape probability is positively correlated with the expected running time and therefore can, in theory, reliably indicate the problem difficulty. This chapter provides the first study of linking escape probability with the expected running time.

5.3.1 Preliminaries

We model the evolution of an EA as a time-homogeneous Markov chain. Therefore, the transition matrix P is the same after each step, the $k-step$ transition probability can be computed as the $k-th$ power of the transition matrix P^k. Below is a list of definitions of notations used in the derivation:

- Running time m_i: On a problem instance, m_i is the average time cost of an EA to reach one of the optima when search starts from solution i, $i = 1,2,\dots,n$, where n is the total number of solutions.
- The escape probability,$p_i^e = \sum\limits_{j,F_j>F_i} p_{ij}$, where F is the fitness function, and $i, j = 1,2,\dots,n$, p_i^e, represents the probability of escaping from solution i to a strictly better solution;
- Column sum in probability transition matrix, $\mathbf{P}$, $c_i = \sum_{j=1}^{n} p_{ji}$, where $i = 1,2,\dots,n$;
- Let $\mathbf{P}^{(k)} = \mathbf{P}^k$, and $c_i^{(k)} = \sum_{j=1}^{n} p_{ji}^{(k)}$, where $i, j = 1,2,\dots,n$, and $k = 1,2,3,\dots$

5.3.2 Derivation of the Mathematical Equation between Escape Probability and Expected Running Time

Equation (5.1) below shows that m_i can be computed as the runtime of the search starting from j by m_j times p_{ij} plus one, where i and j are adjacent solutions with distance one:

$$m_i = \sum_{j=1}^{n} m_j p_{ij} + 1, i = 1,2,\dots,n \tag{5.1}$$

Equation (5.1) can be rewritten as:

$$m_i = \sum_{j=1}^{n} (m_j + 1) p_{ij} = \sum_{j \neq i} (m_j + 1) p_{ij} + m_i p_{ii} \tag{5.2}$$

The sum of m_i is:

$$\begin{aligned}\sum_i m_i &= \sum_{i=1}^{n}\sum_j (m_j+1)p_{ij} = \sum_j \sum_{i=1}^{n}(m_j+1)p_{ij}\\ &= \sum_j m_j \sum_{i=1}^{n} p_{ij} + \sum_j \sum_{i=1}^{n} p_{ij}\\ &= \sum_j m_j c_j + n\end{aligned} \tag{5.3}$$

Equation (5.4) below shows the computation of m_i as a time series:

$$\begin{aligned}m_i = {} & 1 * p_i^e + \sum_{j\neq i} m_j p_{ij}^{(1)}\\ & + 2*(1-p_i^e)^1 p_i^e + \sum_{j\neq i} m_j p_{ij}^{(2)}\\ & + \ldots\\ & + k*(1-p_i^e)^{k-1} p_i^e + \sum_{j\neq i} m_j p_{ij}^{(k)}\\ & + \ldots\end{aligned} \tag{5.4}$$

Summing both sides of Equation (5.4) we obtain the following:

$$\begin{aligned}\sum_i m_i &= \sum_i 1/p_i^e + \sum_{k>0}(\sum_j m_j(c_j^{(k)} - p_{jj}^{(k)}))\\ &= \sum_i 1/p_i^e + \sum_j m_j \sum_{k>0}(c_j^{(k)} - p_{jj}^{(k)})\end{aligned} \tag{5.5}$$

By solving the system of equations formed by Equations (5.3) and (5.5):

$$\begin{cases}\sum_i m_i = \sum_i 1/p_i^e + \sum_j m_j \sum_{k>0}(c_j^{(k)} - p_{jj}^{(k)})\\ \sum_i m_i = \sum_j m_j c_j + n\end{cases} \tag{5.6}$$

The equation between escape probability and expected running time can be obtained:

$$\sum_i 1/p_i^e - n = \sum_{k>1}\sum_i m_i(-c_i^{(2)} - c_i^{(3)} - \ldots - c_i^{(k)} + p_{ii}^{(1)} + p_{ii}^{(2)} + \ldots + p_{ii}^{(k)}) \tag{5.7}$$

As we see from the above equation, on the left is the reciprocal sum of the escape probabilities minus a constant n. On the right is the expected running time; the coefficients $\sum_{k>1}\sum_i m_i(-c_i^{(2)} - c_i^{(3)} - \ldots - c_i^{(k)} + p_{ii}^{(1)} + p_{ii}^{(2)} + \ldots + p_{ii}^{(k)})$ constitute for the weights of m_i for computing the expected value of m_i, namely, the expected running time.

According to the relationship between escape probability and the expected running time established in Equation (5.7), the larger the reciprocal sum of the escape probability, the larger the expected running time. In sum, we can use the escape probability as an indication of the expected running time, as well as estimating the expected running time based on values of escape probability.

5.4 Fitness-Probability Cloud

From the results presented in Section 5.3, the escape probability is a theoretically reliable difficulty measure derived from the expected running time, a difficulty measure widely used in complexity theory of EAs. However, the mere definition of the escape probability is not sufficient for it to be used as an effective difficulty measure in practice. In this section we build on the definition of escape probability and propose the fitness-probability cloud and accumulated escape probability as the first difficulty measure with solid theoretical basis in complexity theory of EAs.

5.4.1 Definition of Fitness-Probability Cloud

5.4.1.1 Escape Probability

$F = \{f_0, f_1, \dots, f_L \quad | \quad f_0 < f_1 < \cdots < f_L\}$ denotes the distinctive set of all possible fitness values of the fitness function. Let us partition the entire set of solutions in the search space into $L+1$ sets according to their fitness values. For a solution with fitness value f_i, S_i denotes the average number of steps required for it to reach a solution with better fitness. The escape probability $P(f_i)$ is defined as follows:

$$P(f_i) = \frac{1}{S_i}. \tag{5.8}$$

The greater the escape probability is for a particular f_i, the easier it is to reach a better fitness from the corresponding fitness level.

5.4.1.2 Fitness-Probability Cloud

We can extend the definition of escape probability to be on a set of fitness values. P_i denotes the average escape probability for individuals of fitness value equal to or above f_i and is defined as:

$$P_i = \frac{\sum_{f_j \in C_i} P(f_j)}{|C_i|}, \tag{5.9}$$

where $C_i = \{f_j | j \geq i\}$. If we take into account all the P_i for a given problem, this would be a good indication of the degree of evolvability of the problem. The fitness-probability cloud (fpc) is defined as:

$$fpc = \{(f_0, P_0), \dots, (f_L, P_L)\}. \tag{5.10}$$

5.4.1.3 Accumulated Escape Probability

Based on the concept of fpc, a numerical measure called accumulated escape probability (aep) is defined:

$$aep = \frac{\sum_{f_i \in F} P_i}{|F|}. \tag{5.11}$$

The aep is actually the average value of P_i across F. By definition of escape probability $P(f_i)$ and P_i, it is clear to see that aep can measure the problem hardness in the following way: the larger the value of aep, the easier the problem is with respect to the given EA.

5.4.2 Methodology for Generating fpc

Here we describe the methodology in generating the fpc for a given problem and an operator. The size of the search space is exponentially large and does not allow consideration of all individuals, therefore sampling is required. Since not all solutions in the search space are equally important, it is preferred to sample the space according to a distribution that gives higher weight to individuals of higher fitness values. In fact, this can be achieved by using a Metropolis method or any other equivalent method [26]. In our case, we chose to use the Metropolis-Hastings sampling method described in [36].

For each sampled point, the escape probability is estimated by computing the proportion of potential better moves out of the entire neighbour set generated by one application of the genetic operator. The larger the number of neighbours sample, the more accurate the estimated escape probability would be. Hereinafter we refer to F as the set of distinctive fitness values of the sampled individuals obtained with the Metropolis-Hastings method, and for each $f_i \in F$, P_i is the estimated average escape probability computed from the sampled neighbourhood set.

5.4.3 Test Problems

5.4.3.1 Unitation Functions

Three unitation functions: OneMax, Trap, OneMix [29] are used.

Definition 5.1. Let s be a bit string of length l, the unitation u(s) of s is a function defined as: $u(s) = \sum_{i=1}^{l} s_i$.

OneMax functions are generalisations of the unitation $u(s)$ of a bit string s:

$f(s) = d \cdot u(s)$, where d is 1.

The Trap function [9] is defined as follows:

$$f(s) = \begin{cases} \frac{a}{z}(z - u(s)), & if \quad u(s) \leq z \\ \frac{b}{l-z}(u(s) - z), & otherwise \end{cases} \tag{5.12}$$

where a represents a local optimum and b is a global optimum, z is a slope-change location.

The OneMix function is a mixture of the OneMax function and the ZeroMax function, which is formally defined as:

$$f(s) = \begin{cases} (1+a)(\frac{l}{2} - u(s)) + \frac{l}{2}, & if \quad g(s) \\ u(s), & otherwise \end{cases} \tag{5.13}$$

where a represents a constant above zero and $g(s)$ is equal to 1 when $u(s)$ is even and $u(s) < \frac{l}{2}$.

5.4.3.2 Subset Sum Problem

The Subset Sum problem is a constrained optimisation problem. Given a set of n items each with an associated weight w, the problem is to select a subset out of n items, where the weighted sum is maximised and does not exceed the budget W. Mathematically this problem is formulated as follows:

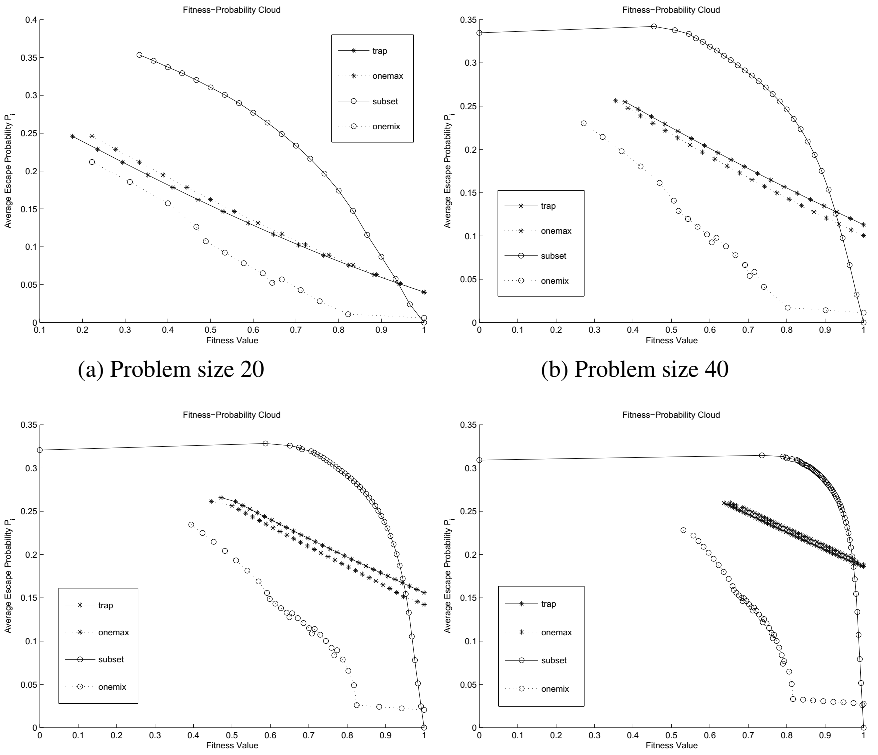
$$\text{Maximise} \quad \sum_{i=1}^{n} w_i x_i, \tag{5.14}$$

$$\text{Subject to} \quad \sum_{i=1}^{n} w_i x_i \leq W, \quad x_i \in \{0,1\}, \quad W = \frac{\sum_{i=1}^{n} w_i}{2}. \tag{5.15}$$

5.4.4 Experimental Results

Once a measure of problem hardness and the way to compute it have been chosen, the problem remains to find a means to validate the prediction of the measure with respect to the problem instance and the algorithm. The easiest way is to use a performance measure [31]. Since the optimal solution to practical problems are unknown, we use the number of fitness evaluations until a certain stopping criterion is satisfied as the performance measure.

Then we evaluate the effectiveness of the fitness-probability cloud and the problem hardness measure accumulated escape probability (*aep*) on four different test problems: OneMax, Trap, OneMix and Subset Sum. For each test problem, four problem instances are generated with problem size varying from 20 to 200. To experimentally confirm the predictions given by the *aep* measure, we use the mutation-based (μ + λ)EA (μ denotes the number of parents, λ the number of offspring) with the following characteristics: mutation operator with flip probability $1/n$ for each bit (bitwise mutation), the algorithm stopped if there is no improvement after 500 fitness evaluations. For each problem instance, 100 independent executions were performed.



(a) Problem size 20

(b) Problem size 40

(c) Problem size 80

(d) Problem size 200

Fig. 5.1 Plot of Fitness-Probability Cloud for Four Test Problems of Problem Sizes 20, 40, 80 and 200

We applied the approach described in Section 5.3 to generate fpc and calculate the value of the corresponding aep. For each problem instance, 1000 samples were obtained using the Metropolis-Hastings sampling method. For each sampled point, the bitwise mutation operator was used to generate 10000 neighbours in order to estimate the escape probability. Under the above parameter settings, we generate the fitness-probability clouds for four test problems of problem sizes 20, 40, 80 and 200. The results are illustrated in Figure 5.1(a)to Figure 5.1(d), respectively. With the fitness-probability cloud generated, we can then apply the method defined in Section 5.3 to compute the accumulated escape probability (aep). We then contrast predictions of aep to the performance from actual runs defined above. For the sake of comparison, we also compute the values of nsc [37]. If $nsc = 0$, the problem is easy; if $nsc < 0$ the problem is difficult and the value of nsc quantifies this difficulty: the smaller its value, the more difficult the problem. The experimental results are summarised in Table 5.1.

Table 5.1 *aep* Predictions vs. Actual Performance for Four Problems of Size 20, 40, 80 and 200. Column 3 to 5 are the number of fitness evaluations taken by three different $(\mu+\lambda)$ EAs.

Problem	Problem Size	(1+1) EA	(3+7) EA	(7+3) EA	*aep*	*nsc*
OneMax	20	641	1166	1110	0.135	0
Trap	20	627	1158	1105	0.135	0
OneMix	20	745	1375	1330	0.09	-8.1932
Subset Sum	20	548	1009	928	0.22	-1.1572
OneMax	40	821	1434	1430	0.175	-0.333
Trap	40	829	1422	1438	0.182	0
OneMix	40	1028	1776	1728	0.105	-16.3114
Subset Sum	40	533	1009	928	0.239	-6.818
OneMax	80	1267	2002	2134	0.202	-0.5
Trap	80	1273	2004	2115	0.209	-0.25
OneMix	80	1609	2608	2678	0.121	-20.4879
Subset Sum	80	547	1015	936	0.246	-7.5286
OneMax	200	2640	3848	4221	0.225	-3
Trap	200	2590	3860	4242	0.222	0
OneMix	200	3070	4724	4952	0.121	-30.175
Subset Sum	200	534	1021	945	0.252	-8.6169

In terms of the defined performance measure, the relevant problem hardness of the four test problems remains the same across problem sizes of 20, 40, 80, and 200. Among different problems, the order of problem difficulty indicated by the performance measure is: Subset Sum $<$ OneMax $\approx$ Trap $<$ OneMix. If we look at the order of problem hardness indicated by the *aep* measure, by definition of the *aep*, the smaller the *aep* value, the more difficult the problem is, it is clear that *aep* consistently orders the hardness of the four test problems across all four problem sizes, with results in qualitative agreement with the actual performance. However, the *aep* is unable to quantify the magnitude of the difference in problem hardness among those problems. In contrast to the correct predictions made by the *aep* measure, as we can see from Table 1, the results given by *nsc* do not correspond to the actual performance. Consequently, the *nsc* fails to correctly predict the relevant problem hardness among OneMax, Trap, OneMix and Subset Sum.

5.5 Parameter Learning Method Using Fitness-Probability Cloud for the Unique Input Output Sequence Problem

This section shows how the difficulty measure *aep* is incorporated within a machine learning method to construct an offline algorithm for tuning EAs to solve the unique input output sequence problem.

5.5.1 Motivation

Finite state machines (FSMs) have been usually used to model software, communication protocols and circuits [19]. The unique input output sequence (UIOS) problem is mostly used for testing finite state machines [11, 12]. To determine whether a given state has an UIOS or not is an NP-hard problem [19]. The UIOS problem has been reformulated as an optimisation problem, and several EAs have been developed to tackle it [10, 11].

It is widely acknowledged that good parameter values are essential for good EA performance, in the meantime, tuning EA parameters stands as one of the persisting challenges in the field of EC. Previous work revealed that 90% of the time is spent on fine-tuning algorithm parameter settings [1]. Most existing approaches for parameter tuning attempt to find one best parameter setting for an EA to solve all instances of the same problem or at least a class of instances [2, 16, 28]. Others used problem dependent features to characterise the problem instances and the feature selection process relies heavily on the domain knowledge. For example, SATzilla [41] uses 48 features mostly specified to the SAT problem to construct per-instance algorithm portfolios. A problem-independent feature represented by a behaviour sequence of a local search procedure is used to perform instance-based automatic parameter tuning [23].

The fitness-probability cloud (fpc) has demonstrated to be an appropriate characterisation of fitness landscapes and does not require any a priori knowledge. We incorporate the fpc within a support vector machine(SVM) learning algorithm to automate the parameter tuning of EAs for solving the UIOS problem [22].

5.5.2 Preliminaries

Definition 5.2. (Finite State Machine). A finite state machine (FSM) is a quintuple, $\mathbf{M} = (\mathbf{S},\mathbf{X},\mathbf{Y},\delta,\lambda)$, where $\mathbf{X},\mathbf{Y}$ and $\mathbf{S}$ are finite and nonempty sets of input symbols, output symbols, and states, respectively; $\delta : \mathbf{S} \times \mathbf{X} \longrightarrow \mathbf{S}$ is the state transition function, and $\lambda : \mathbf{S} \times \mathbf{X} \longrightarrow \mathbf{Y}$ is the output function.

Definition 5.3. (Unique Input Output Sequence). An unique input output sequence for a given state s_i is an input/output sequence x/y, where $x \in \mathbf{X}^*, y \in \mathbf{Y}^*, \forall s_j \neq s_i$, $\lambda(s_i,x) \neq \lambda(s_j,x)$ and $\lambda(s_i,x) = y$.

To generate an UIOS using an EA, candidate solutions are represented by input strings restricted to $\mathbf{X}^n = \{0,1\}^n$, where n is the number of states of the FSM. In general, the length of the shortest UIOS is unknown. Assume the objective is to search for an UIOS of input string length n for state s_1 in all FSM instances. The fitness function is defined as a function of the state partition tree [11, 20, 21].

Definition 5.4. (UIOS fitness function [20, 21]). For a FSM M with n states, the fitness function $f : \mathbf{X}^n \longrightarrow \mathbb{N}$ is defined as $f(x) := n - \gamma_M(s,x)$, where s is the initial state for which we want to find an UIOS, and $\gamma_M(s,x) := |\{t \in \mathbf{S} | \lambda(s,x) = \lambda(t,x)\}|$.

There are $|\mathbf{X}|^n$ candidate solutions with $n-1$ different fitness values. A candidate solution x^* is a global optimum if and only if x^* produces an UIOS and $f(x^*) = n-1$.

5.5.2.1 Target Algorithm and Its Parameters

The $(\mu+\lambda)-EAs$ described in *Algorithm 1* are employed to solve the UIOP, with three parameters to tune: population size, neighbourhood operator, and selection operator. Their candidate settings are listed below:

- Population size: We provide 3 different $(\mu+\lambda)$ options: $\{(4+4),(7+3),(3+7)\}$.
- Neighbourhood operator $\mathscr{N}_j,(j=1,2,\ldots,12)$: There are 3 types of neighborhood operators with different mutation probabilities:
  - $\mathscr{N}_1(x) \sim \mathscr{N}_5(x)$: Bit-wised mutation, flip each bit with probability $p = c/n$, where $c \in \{0.5, 1, 2, n/2, n-1\}$;
  - $\mathscr{N}_6(x) \sim \mathscr{N}_9(x)$: flip *c-th* bits, where $c = \{1, 2, n/2, n-1\}$;
  - $\mathscr{N}_{10}(x) \sim \mathscr{N}_{12}(x)$: Non-uniform mutation [4], for each bit $i, 1 \leq i \leq n$, flip it with probability $\chi(i) = c/(i+1)$, where $c = \{0.5, 1, 2\}$.

  In total 12 neighbourhood operators are used, and then 12 fitness-probability clouds are generated to characterize an UIOS instance.
- Selection operator $\mathscr{S}_i,(i=1,2)$: Two selection schemes are considered in this paper:
  - Truncation Selection: Sort all individuals in $\mathbf{P}^{(k)}$ and $\mathbf{P}_m^{(k)}$ by their fitness values, then select μ best individuals as the next generation $\mathbf{P}^{(k+1)}$.
  - Roulette Wheel Selection: Retain all the best individuals in $\mathbf{P}^{(k)}$ and $\mathbf{P}_m^{(k)}$ directly, and the rest of the individuals of the population are selected by roulette wheel selection.

Algorithm 1. $(\mu+\lambda)$- Evolutionary Algorithms

```
Choose $\mu$ initial solutions $\mathbf{P}^{(0)} = \{x_1^{(0)}, x_2^{(0)}, \ldots, x_\mu^{(0)}\}$ uniformly at random from $\{0,1\}^n$
$k \longleftarrow 0$
While termination criterion is not met $\mathbf{P}_m^{(k)} \longleftarrow \mathscr{N}_j(\mathbf{P}^{(k)})$ %%mutation operator
$\mathbf{P}^{(k+1)} \longleftarrow \mathscr{S}_i(\mathbf{P}^{(k)}, \mathbf{P}_m^{(k)})$        %%selection operator
$k \longleftarrow k+1$
```

For a given UIOS instance, there are 72 different combinations of parameters that are viewed as 72 different EA parameters settings, and our goal is to find 'good' settings for a given UIOS instance. The stopping criterion used in *Algorithm 1* is satisfied when an UIOS has been found.

5.5.3 *fpc-Based Parameter Learning Method*

The framework of selecting an EA parameter setting consists of two phases. In brief, an EA parameter setting predictor is trained using the support vector machine (SVM) learning algorithm at first. In the second phase, when a new UIOS instance is fed into the predictor, the features are generated and fed to the predictor in order to find good parameter settings for the instance. This two-phase framework is illustrated in Figure 5.2.

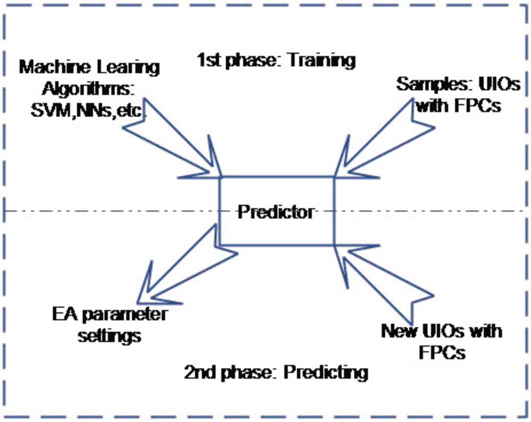


Fig. 5.2 The two-phase framework in parameters tuning

5.5.3.1 First Phase: Training the Predictor

First of all, the structure of the training data is introduced and denoted by a tuple $\mathbf{D} = (\mathbf{F}, \mathbf{PC}, \mathbf{L})$. In tuple **D**, **F** represents the features of a problem instance. For an UIOS instance, the features are a vector of *aep* values [24]. One neighbourhood operator produces one distinctive fitness landscape as well as a fitness-probability cloud; the more neighbourhood operators applied to a problem instance, the more features of it can be generated. This paper adopts 12 mostly used neighbourhood operators in the literature to generate 12 fitness-probability clouds for characterizing an UIOS instance.

PC of tuple **D** denotes the ID of an EA parameter setting. Each problem instance represented by its features **F** is solved by the target algorithm with 72 parameter settings. The performances of these settings are measured by the number of function evaluations. The number of function evaluations of parameter setting j on problem instance i is denoted by $\mathbf{E}_{ij}$, $j = 1, 2, \ldots, 72$.

L is a binary variable, **L** = $\{$good, bad$\}$. **L** indicates whether the parameter setting is good or not, which is determined by a threshold value v. L is good if the number of function evaluations is below v.

To generate training data, we randomly select some problem instances at first. $\mathbf{P} = \{p_1, p_2, \ldots, p_m\}$ denotes the set of m problem instances. For each problem instance, we have a set of neighbourhood operators $\mathscr{N}_i, i = 1, 2, \ldots, 12$ to generate a set of *aep* values as its features. $\mathbf{L}$ is then instantiated by executing EAs with different parameter settings on the set of problem instances.

The data sets used are imbalanced and with small sample sizes. In light of these characteristics, the choice of the machine learning algorithm becomes clear. The support vector machine is a popular machine learning algorithm which handles small samples well. It is therefore employed to train the predictor of good EA parameter setting of the UIOS instances.

5.5.3.2 Second Phase: Predicting Good EA Parameter Settings

When a new UIOS instance comes in, its features $(aep_1, aep_2 \ldots, aep_{12})$ are generated and fed to the predictor to find good EA parameter settings for the corresponding instance.

5.5.4 Experimental Results

In order to test our framework on the UIOP, 24 UIOS instances have been generated at random. The problem size, n, of all instances is 20. We applied the approach described in Section 5.3 to generate the training data. In the first phase of the training predictor, we ran and EA with each parameter setting on each UIOS instance for 100 times and take the average number of function evaluations. The stopping criterion is when an UIOS is found. For each UIOS instance, 72 training samples are generated from 72 different parameter settings. In total we have 1728 samples. The 10 × 10-fold cross validation was adopted to evaluate our method.

We are interested in ‘good’ EA parameter settings (gEAPC) for every instance, and the best EA parameter setting having the smallest fitness evaluations on an instance was labeled ‘good’ in our experiments, the remaining 71 settings were labelled ‘good’ or ‘bad’ depending on the differences between their fitness evaluations and the threshold value v. We let $v = pr \times \bar{\mathbf{E}}_i$, where $\bar{\mathbf{E}}_i$ is the mean value of fitness evaluations on instance i and pr replaces v to regulate the number of gEAPC.

As shown in Table 5.2, the number of gEAPC (2nd column ‘#gEAPC’) in all 1728 samples was decreasing while we were reducing pr. For an UIOS instance there exists at least one gEAPC, the ideal output of the predictor would be a single best gEAPC. The number of gEAPC is influenced by the value of pr. If it is large, almost half of the settings are labelled ‘good’. The smaller the value of pr, the less gEAPC we will have, but the correct rate of predicting gEAPC, denoted by s_g in Table 5.1, is decreasing when pr is smaller than 0.1. Furthermore, we found out that more and more instances have no gEAPC predicted when decreasing the value of pr. The 4th column of Table 5.1 is ‘no’ if there existed any testing instance without a predicted gEAPC. Table 5.1 shows that the best value of pr was 0.11 and there are about 267 gEAPC and all instances will have at least one predicted gEAPC.

Table 5.2 Correct rates of predicting gEAPC with different values of pr. Values of s_g in 3rd column, the average of 10×10 fold cross validation, is the proportion of gEAPC that have been correctly classified.

pr	#gEAPC	s_g	gEAPC found?
0.7	1180	0.500	yes
0.6	1115	0.510	yes
0.5	1007	0.709	yes
0.45	955	0.690	yes
0.4	874	0.689	yes
0.35	806	0.685	yes
0.3	716	0.726	yes
0.25	604	0.689	yes
0.2	489	0.653	yes
0.18	441	0.656	yes
0.16	391	0.709	yes
0.15	377	0.694	yes
0.14	343	0.698	yes
0.135	328	0.632	yes
0.13	326	0.687	yes
0.125	306	0.875	yes
0.12	299	0.764	yes
0.115	286	0.903	yes
0.11	267	0.933	yes
0.1	237	0.925	no
0.09	200	0.861	no
0.08	177	0.864	no
0.05	71	0.782	no
0.01	50	0.620	no

5.5.5 Discussions

EA parameters setting significantly affects the performance of the algorithm. The fpc-based parameter learning method is presented here. The framework is instance-based, through which the EA can be tuned with good parameter settings for a new problem instance based on its features characterised by fitness-probability clouds. On the UIOP the experimental results showed that by properly setting the values of v or pr, a small set of good settings could be discriminated and for each instance there exists at least one good parameter setting.

Future work includes to test the proposed method on a wider range of problems, and to investigate more machine learning algorithms other than the support vector machine when the training sample is small. Furthermore, an in-depth analysis of the time cost of the proposed method would be carried out.

5.6 A fpc-Based Online Adaptive Evolutionary Algorithm

This section proposes to incorporate the difficulty measure *aep* with adaptive EAs to perform online optimisation.

5.6.1 Motivation and Background

It has long been acknowledged that the choice of operator settings has a significant impact upon evolutionary algorithm (EA) performance. However, finding a good choice is somewhat difficult. In contrast, there is evidence, both empirical and theoretical, that the most effective operator settings do vary during the course of an EA run [35]. The problem of devising such a schedule to adapt the operator settings along the search progress, is almost certainly as hard as finding a good static set of operator settings. One key problem concerns the criterion used to judge the performance of candidate operator settings at a given point in the search process.

Looking at the literature, the problem of developing an effective adaptive algorithm has long been recognised. In general, two classes of adaptation methods have been distinguished [34]:

- Self-adaptation: the values of the operator probabilities are directly encoded in the representation of the individual solutions.
- Adaptive Rules: the search strategy is dependent on the state of the EA run.

Self-adaptation is particularly applied for numerical optimisation. Adaptive rules are more interesting as they can be applied to solve combinatorial optimisation problems as well. Looking at the literature it becomes clear that most of the existing adaptive rules belong to the probability matching type [6, 15, 35, 39]. Alternative rules include adaptive pursuit method [34] and the operator selection mechanism using the multi-armed bandit paradigm [7].

Previous work has seen some success along the years like in [18]. However, we have identified three major limitations of the existing approaches: First, they all work with a static set of candidate operator settings, whilst adapting the probabilities of applying them, therefore limiting the adaptivity that can be provided. Furthermore, the use of the search state information by their decision-making mechanisms are efficient but not sufficient. Apart from the aforementioned, most previous mechanisms assume operators that produced better offspring in the past are supposed to perform better in the future. This assumption seems to lack sufficient justification in that past performance does not necessarily reflect the potential of operators to further improve the fitness quality.

5.6.2 Algorithm Description

In the proposed fpc-based adaptive evolutionary algorithm, we introduce a variable set of candidate operator settings to allow greater adaptivity. To address other issues mentioned above, it seems promising to introduce the difficulty measure into the algorithm, for the reason that as the search proceeds, the difficulty measure can exploit the information about the evolutionary progress, and indicate the most suited operator with respect to the local landscape. Therefore, the fpc-based adaptive rules are very promising in navigating the adaptation correctly [25].

For the first time, we have incorporated the problem difficulty measure for EAs with the online optimisation by introducing the fpc into an adaptive evolutionary algorithm. The proposed algorithm is illustrated in Algorithm 2.

Algorithm 2. fpc-based Online Adaptive Evolutionary Algorithm

```
Require: Select an initial operator $op = op_{ini}$ from the candidate set $OP$
  while No improvement of quality of the best solution for $G_a$ generations do
    while No improvement of quality of the best solution for $G_b$ generations do
      Apply the operator $op$ to obtain the new population $P$
    end while
    Select a subset of operators from $OP$
    $\forall op \in OP$, $\forall p \in P$, compute the number of steps $K_{p,op}$ required to find an improving
    move, and record the set of better solutions obtained
    Compute the measure $m_{op}$ and update the operator $op$ with the operator obtain the best
    value of the measure. Also update the population $P$.
  end while
```

5.7 Conclusion

In this chapter, for the first time we have proven the correlation between the escape probability and the expected running time, a difficulty measure widely used in complexity theory of EAs. Furthermore, we have proposed the fitness-probability cloud and accumulated escape probability to implement escape probability as a difficulty measure applicable in practice. Experimental results on unitation functions and the subset sum problem showed that the proposed measure is able to discriminate the relevant problem hardness with respect to mutation-based EAs.

More importantly, promising applications of the proposed difficulty measure have been found in both offline and online optimisations. On one hand, the fpc-based parameter learning method for UIOS has demonstrated its effectiveness through the experiments carried out on 24 UIOS instances. On the other hand, we have proposed the fpc-based adaptive EA with the difficulty measure in its core for decision-making.

Future work includes further investigation of the correlation between the escape probability computed theoretically and its estimations in practice on problems with known structure. Also to analyse the time complexity of the proposed adaptive EAs.

Acknowledgements. This work was partially supported by the European Union Seventh Framework Programme under Engineering Proprioception in Computing Systems (EPiCS) (257906), the EU FP7-PEOPLE-2009-IRSES project under Nature Inspired Computation and its Applications (NICaiA) (247619), and the EPSRC grant Dynamic Adaptive Automated Software Engineering (DAASE) (No. EP/J017515/1).

References

[1] Adenso-Diaz, B., Laguna, M.: Fine-Tuning of Algorithms Using Fractional Experimental Design and Local Search. Operations Research 54(1), 99–114 (2006)

[2] Birattari, M., Stuzle, T., Paquete, L., Varrentrapp, K.: A Racing Algorithm for Configuring Metaheuristics. In: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, GECCO 2002, pp. 11–18. Morgan Kaufmann, San Francisco (2002)

[3] Borenstein, Y., Poli, R.: Information Landscapes and Problem Hardness. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, GECCO 2005, pp. 1425–1431. ACM, New York (2005)

[4] Cathabard, S., Lehre, P.K., Yao, X.: Non-uniform Mutation Rates for Problems with Unknown Solution Lengths. In: Beyer, H.G., Langdon, W.B. (eds.) Foundations of Genetic Algorithms (FOGA) XI, pp. 173–180. ACM, New York (2011)

[5] Collard, P., Vérel, S., Clergue, M.: Local Search Heuristics: Fitness Cloud versus Fitness Landscape. In: Proceedings of the 16th European Conference on Artificial Intelligence, ECAI, pp. 973–974. IOS Press, Amsterdam (2004)

[6] Corne, D.W., Oates, M.J., Kell, D.B.: On Fitness Distributions and Expected Fitness Gain of Mutation Rates in Parallel Evolutionary Algorithms. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN VII 2002. LNCS, vol. 2439, pp. 132–141. Springer, Heidelberg (2002)

[7] DaCosta, L., Fialho, A., Schoenauer, M., Sebag, M.: Adaptive Operator Selection with Dynamic Multi-armed Bandits. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008, pp. 913–920. ACM, New York (2008)

[8] Davidor, Y.: Epistasis Variance: A Viewpoint on GA-hardness. In: Rawlins, G.J.E. (ed.) Foundations of Genetic Algorithms (FOGA), pp. 23–35. Morgan Kaufmann, San Francisco (1991)

[9] Deb, K., Goldberg, D.E.: Analyzing Deception in Trap Functions. In: Whitley, L.D. (ed.) Foundations of Genetic Algorithms (FOGA) II, pp. 93–108. Morgan Kaufmann, San Francisco (1993)

[10] Derderian, K., Hierons, R.M., Harman, M., Guo, Q.: Automated Unique Input Output Sequence Generation for Conformance Testing of FSMs. The Computer Journal 49 (2006)

[11] Guo, Q., Hierons, R.M., Harman, M., Derderian, K.: Computing Unique Input/Output Sequences Using Genetic Algorithms. In: Petrenko, A., Ulrich, A. (eds.) FATES 2003. LNCS, vol. 2931, pp. 164–177. Springer, Heidelberg (2004)

[12] Guo, Q., Hierons, R., Harman, M., Derderian, K.: Constructing Multiple Unique Input/Output Sequences Using Metaheuristic Optimisation Techniques. IET Software 152(3), 127–140 (2005)

[13] He, J., Reeves, C., Witt, C., Yao, X.: A Note on Problem Difficulty Measures in Blackbox Optimization: Classification, Realizations and Predictability. Evol. Comput. 15(4), 435–443 (2007)

[14] He, J., Yao, X.: Towards an Analytic Framework for Analysing the Computation Time of Evolutionary Algorithms. Artificial Intelligence 145, 59–97 (2003)

[15] Hong, T., Wang, H., Chen, W.: Simultaneously Applying Multiple Mutation Operators in Genetic Algorithms. Journal of Heuristics 6, 439–455 (2000)

[16] Hutter, F., Hoos, H.H., Leyton-brown, K., Stuetzle, T.: ParamILS: An Automatic Algorithm Configuration Framework. Journal of Artificial Intelligence Research 36, 267–306 (2009)

[17] Jones, T., Forrest, S.: Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. In: Proceedings of the 6th International Conference on Genetic Algorithms, pp. 184–192. Morgan Kaufmann, San Francisco (1995), `http://portal.acm.org/citation.cfm?id=645514.657929`

[18] Jong, K.D.: Parameter Setting in EAs: a 30 Year Perspective, pp. 1–18. Springer (2007)

[19] Lee, D., Yannakakis, M.: Testing Finite-State Machines: State Identification and Verification. IEEE Transactions on Computers 43(3), 30–320 (1994)

[20] Lehre, P.K., Yao, X.: Runtime Analysis of (1+1) EA on Computing Unique Input Output Sequences. In: IEEE Congress on Evolutionary Computation, 2007, pp. 1882–1889 (2007)

[21] Lehre, P.K., Yao, X.: Crossover Can Be Constructive When Computing Unique Input Output Sequences. In: Li, X., et al. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 595–604. Springer, Heidelberg (2008)

[22] Li, J., Lu, G., Yao, X.: Fitness Landscape-based Parameter Tuning Method for Evolutionary Algorithms for Computing Unique Input Output Sequences. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part II. LNCS, vol. 7063, pp. 453–460. Springer, Heidelberg (2011)

[23] Lindawati, Lau, H.C., Lo, D.: Instance-Based Parameter Tuning via Search Trajectory Similarity Clustering. In: Coello, C.A.C. (ed.) LION 5 2011. LNCS, vol. 6683, pp. 131–145. Springer, Heidelberg (2011)

[24] Lu, G., Li, J., Yao, X.: Fitness-probability Cloud and a Measure of Problem Hardness for Evolutionary Algorithms. In: Hao, J.-K. (ed.) EvoCOP 2011. LNCS, vol. 6622, pp. 108–117. Springer, Heidelberg (2011)

[25] Lu, G., Li, J., Yao, X.: Embrace the New Trend in SBSE with Fitness-Landscape Based Adaptive Evolutionary Algorithm. In: Fast Abstracts of the 4th Symposium on Search Based Software Engineering (2012)

[26] Madras, N.: Lectures on Monte Carlo Methods. American Mathematical Society, Rhode Island (2002)

[27] Manderick, B., Weger, M.K., Spiessens, P.: The Genetic Algorithm and the Structure of the Fitness Landscape. In: ICGA 1991, pp. 143–150 (1991)

[28] Maturana, J., Lardeux, F., Saubion, F.: Autonomous Operator Management for Evolutionary Algorithms. Journal of Heuristics 16, 881–909 (2010)

[29] Mengshoel, O.J., Goldberg, D.E., Wilkins, D.C.: Deceptive and Other Functions of Unitation as Bayesian Networks. In: Symposium on Genetic Algorithms, SGA (1998)

[30] Merz, P.: Advanced Fitness Landscape Analysis and the Performance of Memetic Algorithms. Evol. Comput. 12, 303–325 (2004)

[31] Naudts, B., Kallel, L.: A Comparison of Predictive Measures of Problem Difficulty in Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation 4(1), 1–15 (2000)

[32] Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover, Mineola (1998)

[33] Radcliffe, N.J., Surry, P.D.: Fitness Variance of Formae and Performance Prediction. In: Whitley, L.D., Vose, M.D. (eds.) Foundations of Genetic Algorithms (FOGA) 3, pp. 51–72 (1995)

[34] Thierens, D.: Adaptive Strategies for Operator Allocation. In: Lobo, F., Lima, C., Michalewicz, Z. (eds.) Parameter Setting in Evolutionary Algorithms, vol. 54, pp. 77–90. Springer, Heidelberg (2007)

[35] Tuson, A., Ross, P.: Adapting Operator Settings in Genetic Algorithms. Evol. Comput. 6(2), 161–184 (1998)

[36] Vanneschi, L., Clergue, M., Collard, P., Tomassini, M., Vérel, S.: Fitness Clouds and Problem Hardness in Genetic Programming. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 690–701. Springer, Heidelberg (2004)

[37] Vanneschi, L., Tomassini, M., Collard, P., Vérel, S.: Negative Slope Coefficient: A Measure to Characterize Genetic Programming Fitness Landscapes. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 178–189. Springer, Heidelberg (2006)

[38] Vassilev, V.K., Fogarty, T.C., Miller, J.F.: Smoothness, Ruggedness and Neutrality of Fitness Landscapes: from Theory to Application, pp. 3–44. Springer, New York (2003)

[39] Whitacre, J., Pham, T., Sarker, R.: Credit Assignment in Adaptive Evolutionary Algorithms. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO 2006, pp. 1353–1360. ACM, New York (2006)

[40] Wright, S.: The Roles of Mutation, Inbreeding, Crossbreeding, and Selection in Evolution. In: Proc. 6th Congr. Genetics, vol. 1, p. 365 (1932)

[41] Xu, L., Hutter, F., Hoos, H., Leyton-Brown, K.: SATzilla: Portfolio-based Algorithm Selection for SAT. Journal of Artificial Intelligence Research 32, 565–606 (2008)

Chapter 6
Geometry and Coarse-Grained Representations of Landscapes

Konstantin Klemm, Jing Qin, and Peter F. Stadler

Abstract. Basic geometric notions describing the structure of landscapes as well as the dynamics of local search on them include basins, saddles, reachability and funnels. We focus on discrete, combinatorial landscapes and emphasize the complications arising from local degeneracies. Local search in such landscapes is well described by adaptive walks, which we use to define reachability of a target from an initial configuration. Reachability introduces a topological structure on the configuration space. Combinatorial vector fields (CVFs) provide a more powerful mathematical framework in which the subtleties of local degeneracy can be conveniently formalized. Stochastic search dynamics has a direct representation as a probability space over the set of CVFs with the given landscape as a Lyapunov function. This ensemble of CVFs is amenable to the framework of standard statistical mechanics. The implications of landscape structure on search dynamics are elucidated further by the fact that the set of all CVFs on a landscape has a product structure, factorizing over extended plateaus (so called shelves) of the landscape. Finally, we discuss the coarse graining of landscapes from two perspectives. Traditionally, a partitioning (e.g. by gradient basins) of a given landscape is used to obtain a landscape with

Konstantin Klemm · Peter F. Stadler
Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstrasse 16-18, 04107 Leipzig, Germany
e-mail: klemm@bioinf.uni-leipzig.de

Jing Qin · Peter F. Stadler
Max Planck Institute for Mathematics in the Sciences,
Inselstraße 22, 04103 Leipzig, Germany
e-mail: qin@bioinf.uni-leipzig.de

Peter F. Stadler
Fraunhofer Institut for Cell Therapy and Immunology,
Perlickstraße 1, 04103 Leipzig, Germany
Institute for Theoretical Chemistry, University of Vienna,
Währingerstrasse 17, A-1090 Vienna, Austria
Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM87501, U.S.A.
e-mail: studla@bioinf.uni-leipzig.de

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_6, © Springer-Verlag Berlin Heidelberg 2014

fewer configurations called macrostates. A reverse, and less investigated, view on coarse graining considers finer landscapes, with a larger number of configurations than the original one and a non-injective mapping into the original configuration space. Such encodings of landscapes, when suitably defined, turn out advantageous for optimization by adaptive walks.

6.1 Introduction

Combinatorial landscape theory provides a framework for the description of the thermodynamics and kinetics of a large class of complex systems. It has proven to be a valuable concept in evolutionary biology, combinatorial optimization and the physics of disordered systems.

The notion of a "fitness landscape" originated in theoretical biology as a technique to visualize evolutionary adaption in 1932 [39]. The basic ingredients are a set of discrete genetic structures, a fitness function used to evaluate every possible structure and a "mutation" function measuring the feasibility of transitions between pairs of different structures. Due to the combined effects of mutation and selection, a population moves uphill/downhill on the landscape, which provides evolutionary information in the form of accessibility or reachability. The rationale behind this view of evolution on a landscape gives rise to the inception of evolutionary algorithms for global search or solving combinatorial optimization tasks such as the traveling salesman problem. The equivalent notion of "energy landscapes" arose in physics as a natural description of disordered systems. In spin glasses, for instance, each spin configuration is assigned an energy describing its Hamiltonian which specifies the model [1]. In theoretical chemistry, energy landscapes are viewed as discrete models to approximate the smooth potential energy surfaces [24]. In structural biology, energy landscapes are used to understand the folding of biopolymers such as RNAs and proteins into their three-dimensional structures [7].

In formal terms, a *(combinatorial) landscape* consists of a *search space* or *configuration space* $\mathbb{X} = (V, \mathscr{T})$ and a fitness or energy function $f : V \to \mathbb{R}$ that evaluates each configuration. In general, $\mathscr{T}$ denotes a (generalized) topological structure on V. In this contribution we will restrict ourselves to the simplest case, namely undirected finite graphs $G = (V, E)$ as search spaces. Similarly, we will assume that the values of f are real numbers. We refer to [9] for some insights into landscapes over recombination spaces and to [33] for landscapes whose values are elements of a partially ordered set. For the sake of clarity we adopt the picture of physics and interpret f as an energy function. Optimization thus seeks low energy configurations by dynamics that tends to minimize f.

In this contribution we focus on geometric and topological features of landscapes, i.e., on properties that arise from the interplay of the structure of G with the function f. These are of particular interest for an understanding of processes on combinatorial landscapes that are governed by local transitions, including in particular a wide variety of heuristic optimization algorithms from simulated annealing to genetic algorithms. Although the relationship between dynamical processes on combinatorial landscapes and geometric properties of the landscape itself has been a long-standing research problem, we still lack a satisfactory theory [27]. Some progress has been made, however, in the analysis of the landscape structure itself. The hierarchical structure of local minima and the barriers between their basins of attraction plays a crucial role in this context.

6.2 Two Examples

Before we proceed, let us briefly introduce two famous examples of combinatorial landscapes:

(A) TSP landscapes. The traveling salesman problem (TSP) is probably the most frequently studied combinatorial optimization problem. Each potential solution to the TSP is a cyclic permutation among n cities, each city occurs once. The configuration space of the TSP landscape consists of all potential solutions of TSP. Each configuration is evaluated by the value of the distance of the total route. Two potential solutions are adjacent in the underlying graph of the TSP landscape if their corresponding permutations can be transformed from each other by exchanging the positions of two cities.

(B) RNA landscapes. The *RNA landscape* may serve as a prototype for biophysically interesting landscapes. An RNA sequence can be viewed as a string over the alphabet over four *bases* $\{\mathbf{A},\mathbf{U},\mathbf{G},\mathbf{C}\}$ of length n. Given an RNA sequence s, an RNA secondary structure is identified as a simple graph with vertex set $\{1,\ldots,n\}$, whose edge set consists of the edges $\{\{i,i+1\}|1 \leq i \leq n-1\}$, together with a further collection of edges P_s such that if $\{i,j\}$, $\{k,\ell\} \in P_s$ with $i<j$ and $k<\ell$ then (i) the particular base combinations at pairing position i and j (k and ℓ) must be **AU**, **GU**, or **GC**; (ii) $i=k$ if and only if $j=\ell$; (3) $k \leq j$ implies that $i<k<\ell<j$. An edge $\{i,j\}$ contained in P_s is called a *base pair*. Those vertices not contained in a base pair are called unpaired. Condition (i) implies that each vertex is allowed to belong to at most one base pair. Condition (ii) excludes the formation of crossing base pairs, i.e. *pseudoknots*. For a given RNA sequence, the number of all valid secondary structures grows exponentially with the sequence length n. Its configuration space of the RNA folding landscape consists of all the valid secondary structures. Each secondary structure is called a configuration in the landscape. The energy of a secondary structure is calculated by `RNAeval` in the `Vienna Package` [14] based on the *loop energy model* [21]. Two configurations x,y are adjacent in the underlying graph, if y can be derived from x by adding or removing a base pair in x (see Figure 6.1(A)).

6.3 Local Minima, Walks, and Degeneracy

In this section we introduce the basic notations and concepts. Throughout, we consider a landscape (G,f) on a finite, undirected, connected graph $G=(V,E)$ with a real-valued energy function $f: V \to \mathbb{R}$. We reserve calligraphic letters for systems of subsets of V and set-valued set functions on V, i.e., maps $2^V \to 2^V$.

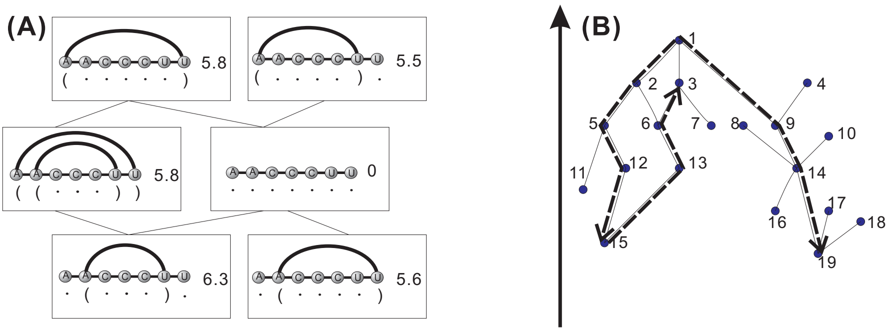


Fig. 6.1 Energy landscapes. **(A)** The configuration space $G = (V,E)$ of the RNA sequence AACCCUU consists of six secondary structures. Two structures $x,y \in V$ are adjacent, $\{x,y\} \in E$, if x can be obtained from y by adding or removing a single base pair. The folding energy $f(x)$ of a configuration x is displayed in the box next to the structure. **(B)** Three types of walks defined on the landscape: a gradient walk $1 \to 9 \to 14 \to 19$ in which 9 is the unique gradient neighbor of 1, an adaptive walk $1 \to 2 \to 5 \to 12 \to 15$ and a hill climbing walk $15 \to 13 \to 6 \to 3$. The vertical arrow between panels (A) and (B) points in the direction of increasing energy.

6.3.1 Neighbors and Minima

We write $N(x) = \{y|\{x,y\} \in E\}$ for the graph-theoretical neighborhood of a vertex x in G. A vertex $x \in V$ is a *local minimum* of (G, f) if $f(x) \leq f(y)$ for all $y \in N(x)$. If $f(x) < f(y)$, we call x a *strict local minimum* . A vertex x is a *global minimum*, also called *ground state* if $f(x) \leq f(y)$ for all $y \in V$. Since V is finite, a global minimum exists for all landscapes. It is not necessarily unique, however. The set of local minima of (G, f) is denoted by $M(G, f)$.

A vertex $y \in N(x)$ with $f(x) = f(y)$ is a *neutral neighbor* of x. We write $N^{>}(x) = \{y \in N(x)|f(x) > f(y)\}$ and $N^{>}[x] = N^{>}(x) \cup \{x\}$ and call x a *drainage point* if $N^{>}(x) \neq \emptyset$. Otherwise, x is a local minimum. Furthermore, we set $N^{>}(W) = \bigcup_{z\in W} N^{>}(z)$ for any subset $W \subseteq V$.

We say that $y \in N(x)$ is a *gradient neighbor* of x if $f(y) = \min\{f(z) : z \in N(x)\}$ and $f(z) < f(x)$. Hence x has a gradient neighbor if and only if it is not a local minimum. In general, a configuration can have more than one gradient neighbor.

For computational purposes it is desirable to *define* a unique gradient neighbor for each non-minimal vertex via introducing a total order L related to the energy function f on the vertex set V. To be precise, this total order can be interpreted as an *energy sorted list*, i.e. a bijective mapping $L : \{1,2,\ldots,|V|\} \to V$ such that $f(L(i)) \leq f(L(j))$ for any i, j with $i < j \leq |V|$. Given this total order L, a configuration $x \in V \setminus M(G, f)$ is assigned the unique gradient neighbor $L(i)$ with $i = \min\{j : L(j) \in N(x)\}$, being the neighbor of x appearing earliest in this list.

6.3.2 Walks and Paths

An *adaptive* walk in (G,f) is a sequence of configurations $w_1, w_2, \ldots, w_\ell$ such that $\{w_{i-1}, w_i\} \in E$ and $f(w_{i-1}) \geq f(w_i)$ for all $1 < i \leq \ell$. Adaptive walks are often called *hill-climbing walks* in the context of maximization: A walk is a hill-climbing walk on $(G,-f)$ if and only if it is an adaptive walk on (G,f). Hill climbing walks are useful as a technical device for minimization problems in Section 6.4. A *gradient walk* is an adaptive walk $w_1, w_2, \ldots, w_\ell$ such that w_i is a gradient neighbor of w_{i-1} for $1 < i \leq \ell$. Gradient walks are often called *best-improvement* or *steepest-descent local search*. We illustrate these three types of walks in Figure 6.1(B). A *neutral walk* in (G,f) is an adaptive walk such that w_i is a neutral neighbor of w_{i-1} for $1 < i \leq \ell$. Note that by definition every neutral walk is also an adaptive walk. Conversely, an adaptive walk is neutral if and only if $f(w_1) = f(w_\ell)$.

A path is a walk in which no two vertices are visited twice. In particular, every gradient walk is a path. Furthermore, we note that every walk contains a path that is obtained by removing every part of a walk that leads from a vertex back to itself. Since G is finite, every path is necessarily finite as well.

6.3.3 Degeneracy, Plateaus, and Shelves

Major technical complications in the analysis of discrete landscapes arise from degeneracy, i.e., the presence of distinct vertices with the same value of f. A landscape (G,f) is *non-degenerate* if $f(x) = f(y)$ implies $x = y$. This condition is too strong for most practical applications since many landscape models have symmetries that lead to degeneracies. For instance, the tours in a TSP can start and end in any city along the way without changing the travel cost. Figure 6.2 is an illustration of a highly degenerate landscape.

Denote by $G^f(x)$ the connected component of the induced subgraph with vertex set $\{z \in V | f(z) = f(x)\}$ such that $x \in G^f(x)$. In the local search literature, $G^f(x)$ is often called a *plateau* or *neutral network* [36]. Every neutral walk with starting configuration x is by construction confined to $G^f(x)$. The relation $x \sim_f y$ defined in V by $x \sim_f y \Leftrightarrow y \in G^f(x)$ is an equivalence relation and thus $\Pi = \{G^f(x) | x \in V\}$ is the set of all the equivalence classes in V. Therefore, it forms a partition of V.

A landscape is *locally non-degenerate* or *invertible on edges* if the following three equivalent conditions are satisfied: (i) $G^f(x)$ consists of a single vertex for all $x \in V$; (ii) there are no neutral walks on (G,f); (iii) $f(x) = f(y)$ implies $y \notin N(x)$. Clearly, if (G,f) is non-degenerate, then it is locally non-degenerate also. But the inverse statement is not true.

We note that strict local minima need not exist unless the landscape is locally non-degenerate. In the general case, therefore, we have to work with non-strict local minima and to accommodate neutral walks.

Alternatively, one may consider the locally non-degenerate quotient landscape $(G,f)/\sim_f$ obtained from (G,f) by contracting the plateaus $G^f(x)$ to single vertices. Since local neutrality is an important characteristic of many landscapes (as in the RNA example, for instance [29]), this simplification may be misleading. We therefore focus on the general, degenerate case in most of our discussion below.

For a detailed investigation into the structure of adaptive walks, we will need the neutral components together with their downhill neighbors. To this end, we define for every subgraph H of G the subgraph $\overrightarrow{H}$ with vertex set $V(\overrightarrow{H}) = \bigcup_{x\in V(H)} N^{>}[x]$ and edge set $E(\overrightarrow{H}) = E(H) \cup \big\{\{x,y\} \in E | x \in V(H), y \in N^{>}(x)\big\}$. In the particular case that $H = G^f(x)$, we call a graph $\overrightarrow{G^f(x)}$ a *shelf* of (G,f). For every shelf $A = \overrightarrow{G^f(x)}$ of (G,f) we distinguish between the *flat surface* $A^{\text{flat}} = V(G^f(x))$ of the shelf, i.e., the vertices of the plateau $G^f(x)$, and its *exit points* $A^{>} = \{y \in N^{>}(x') | x' \in V(\overrightarrow{G^f(x)})\}$.

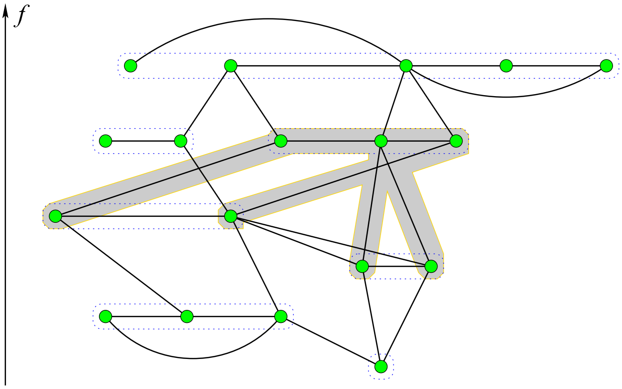


Fig. 6.2 Example of a small highly degenerate landscape. Vertices of G are arranged according to the fitness values $f(x)$. Connected components $G^f(x)$ are indicated by dotted boxes. For one of them, the corresponding shelf-graph $\overrightarrow{G^f(x)}$ is highlighted in gray.

Shelves are constructed such that their flat surfaces form the partition of the vertex set of G into the plateaus of (G,f) while their edge sets form a partition of the edge set of G [31]. In locally non-degenerate landscapes, the flat surfaces consist of single points so that each shelf consists of a vertex and its downhill neighbors.

A shelf A is *locally minimal* if $A^{>} = \emptyset$. In this case $A^{>} \subseteq M(G,f)$, i.e., all vertices of locally minimal shelves are local minima. The converse is not true: shelves with exit points may also contain local minima. All strict local minima, of course, correspond to locally minimal shelves that consist of a single vertex only.

6.3.4 Reachability

The concept of adaptive walks implies a simple concept of reachability among the vertices of G: y is *reachable* from x, $x \rightsquigarrow y$, if there is an adaptive walk (and hence an adaptive path) starting at x that contains y. If A and B are shelves, we say that B is *reachable* from A if there is an $x \in A$ and a $y \in B$ so that $x \rightsquigarrow y$. Every vertex of the flat surface of a shelf is reachable from every other vertex of the flat surface. Thus, if B is reachable from A then every $x \rightsquigarrow y$ for all $x \in A$ and $y \in B$.

Naturally, one considers the system of sets

$$\mathscr{C}(x) = \{y \in V | x \rightsquigarrow y\} \tag{6.1}$$

on (G, f). By construction $x \in \mathscr{C}(x)$ and transitivity of $\rightsquigarrow$ immediately implies that $\mathscr{C}(y) \subseteq \mathscr{C}(x)$ whenever $x \rightsquigarrow y$. Furthermore, let us consider set-wise reachability

$$\mathscr{C}(W) = \bigcup_{x \in W} \mathscr{C}(x) \tag{6.2}$$

so that $y \in \mathscr{C}(W)$ if there is a point $z \in W$ such that $z \rightsquigarrow y$. As shown in [31], the function $\mathscr{C} : 2^V \to 2^V$ satisfies Kuratowski's closure axioms and hence defines a reachability topology $\tau_{\mathscr{C}}$ on V.

Once we decide to consider the finite topological space $(V, \tau_{\mathscr{C}})$ instead of the graph G we have to clearly distinguish between graph-theoretical properties of G and topological properties of $(V, \tau_{\mathscr{C}})$. Connectedness of sets, for instance becomes a property derived from the topology on V. It can be shown that $A \subset V$ is connected in a reachability topology $\tau_{\mathscr{C}}$ if and only if any two points $x, y \in A$ are connected by a path $x = x_0, x_1, \ldots, x_l = y$ such that $x_i \in \mathscr{C}(x_{i-1})$ or $x_{i-1} \in \mathscr{C}(x_i)$ [31]. For the notion of reachability defined above, however, it is easy to see that topological and graph theoretical connectedness coincide: If $u \in \mathscr{C}(v)$, there is a path in G connecting u and v, hence the reachable sets are connected in the graph-theoretic sense. A general result from topology ensures closures of points are topologically connected. Conversely, we have $u \in \mathscr{C}(v)$ or $v \in \mathscr{C}(u)$ for adjacent vertices in G by construction of our notion of reachability. Thus pairs of adjacent vertices form $\tau_{\mathscr{C}}$-connected sets. Note that this would no longer be true for a more restrictive definition of accessibility requiring, e.g., strictly adaptive walks or even more extremely, gradient walks. We will briefly return to this point when discussing combinatorial vector fields in Section 6.7.

We remark that the minimal closed sets (w.r.t. set inclusion) in the reachability topology coincide with the locally minimal shelves and hence with the connected components of $M(G, f)$.

Let W be a closed connected set. We say that W is a *valley* (w.r.t. the reachability topology $\tau_{\mathscr{C}}$) if there is no connected closed set W' so that (i) $W \subset W'$ and (ii) W and W' contain the same minimal closed sets. In other words, any closed connected superset W' of a valley W contain a local minimum not contained in W. By construction, therefore, we can characterize the valleys of f by the collection of minimal closed sets (local minima) that they contain (see Figure 6.3). A more detailed discussion of valleys can be found in [31].

6.3.5 Mutual Accessibility

We say that x and y are *mutually accessible at level h* , in symbols $x \hookleftarrow\!\underline{\;\,h\;\,}\!\hookrightarrow y$, if there is a walk P starting at x ending at y such that $f(z) \leq h$ for all $z \in P$. It is convenient to define the sets

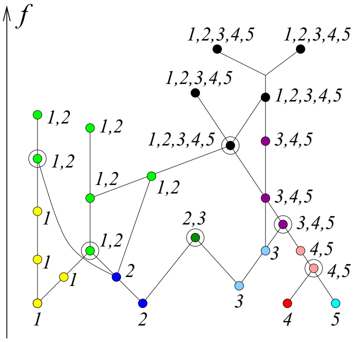


Fig. 6.3 The local minima in this simple landscape are labeled 1 through 5. Each vertex of (G, f) is annotated with the list of reachable local minima that are reachable from it. These labels, i.e., $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{1,2\}$, $\{2,3\}$, $\{4,5\}$, $\{3,4,5\}$, and $\{1,2,3,4,5\}$ identify the valley. Vertices that connect valleys are highlighted by circles. The valleys do not form a hierarchical structure: For instance, $W_{\{1,2\}}$ and $W_{\{2,3\}}$ share the local minimum 2 but both 1 and 3 are contained in only one of the two sets.

$$\mathscr{B}_h(x) = \{y \in V | x \overset{h}{\longleftrightarrow} y\} \tag{6.3}$$

for every $h \geq f(x)$. Trivially, $x \rightsquigarrow y$ implies $x \overset{h}{\longleftrightarrow} y$ for all $h \geq f(x)$. Thus we have $\mathscr{C}(y) \subseteq \mathscr{B}_h(x)$ for all $x \in V$, $y \in \mathscr{B}_h(x)$, and $h \geq f(x)$. The sets $\mathscr{B}_h(x)$ are therefore closed in the reachability topology. By construction, they are connected. Hence there is a unique minimal valley W that contains $\mathscr{B}_h(x)$.

The sets $\mathscr{B}_h(x)$ also have an alternative interpretation: they are the $\tau_{\mathscr{C}}$-connected components of the *level sets* $V_h := \{x \in V | f(x) \leq h\}$, which appears to be the much more natural definition.

6.4 Basins and Saddles

For each local minimum $x \in M(G, f)$ we define the gradient basin $\mathscr{G}(x)$ as the set of configurations $z \in V$ so that the unique gradient walk with starting point in z ends in x. Analogously, we write $\mathscr{G}(L) = \bigcup_{x \in L} \mathscr{G}(x)$ for the gradient basin of the minimal shelf L. We note for later reference that $\{\mathscr{G}(x) | x \in M(G, f)\}$ forms a partition of V.

Analogously, we define the adaptive basin $\mathscr{A}(x) = \{z \in V | z \rightsquigarrow x\}$ for all local minima $x \in M(G, f)$. In contrast to the gradient basins, the adaptive basins form a covering of V that in general will not be a partition. By construction we have $x \in \mathscr{G}(x) \subseteq \mathscr{A}(x)$.

For $x, y \in M(G, f)$ and $s \in V$, we say that s is a *gradient saddle* between x and y if (i) there are neighbors $s' \in N(s) \cap \mathscr{G}(x)$ and $s'' \in N(s) \cap \mathscr{G}(y)$ with $f(s'), f(s'') \leq f(s)$; and (ii) s is a configuration with minimal energy fulfilling property (i). In this case, we set the *gradient saddle height* $\mathsf{GS}(x, y) = f(s)$. We set $\mathsf{GS}(x, y) = \infty$ if x and y are not connected by a gradient saddle.

A *direct saddle* is defined analogously involving adaptive basins. We say that $s \in V$ is a direct saddle point between x and y if s is an element of $\mathscr{A}(x) \cap \mathscr{A}(y)$ with minimal energy. Equivalently, s is a direct saddle between x and y if (i) $x, y \in \mathscr{C}(s)$ and (ii) there is no point s' with $f(s') < f(s)$ and $x, y \in \mathscr{C}(s')$. For any two local minima we define the *direct saddle height* $\mathsf{DS}(x,y) = f(s)$ if a direct saddle s between x and y exists. Otherwise we set $\mathsf{DS}(x,y) = \infty$.

If s is a direct saddle point between two local minima, it is also a gradient saddle for some, but not necessarily the same two local minima. In general we have $\mathsf{DS}(x,y) \le \mathsf{GS}(x,y)$.

The existence of a direct saddle point s between two local minima x and y implies that there is a path P in G from x to y so that $f(v) \le f(s)$ for all $v \in P$. This is not necessarily the smallest bound on the "peak" of the path, however. The *saddle height* between any two vertices is the minimal value of f at which x and y are mutually reachable, i.e.,

$$\mathsf{S}(x,y) = \min\{h | x \overset{h}{\hookleftarrow\!\!\hookrightarrow} y\} \tag{6.4}$$

A configuration $s \in V$ is a *saddle point* between two distinct local minima $x, y \in M(G,f)$ if (i) $f(s) = \mathsf{S}(x,y)$ and (ii) there is a path P from x to y passing through s so that $f(s) \ge f(z)$ for all $z \in P$. In contrast to gradient saddle points, thus, one can always find a saddle point since G is assumed to be connected. In the degenerate case, it is common that the saddle point for two given local minima x and y is not unique.

It is well known that S is an ultrametric distance measure [26], i.e., for all z

$$\mathsf{S}(x,y) \le \max\{\mathsf{S}(x,z), \mathsf{S}(y,z)\} \ . \tag{6.5}$$

Obviously, we have $\mathsf{S}(x,y) \le \mathsf{DS}(x,y)$. We illustrate the differences between direct saddles, saddles, and gradient saddles in Figure 6.4. We remark that computing saddle heights and saddle points is a difficult task in general. For landscapes of RNA secondary structure, for instance, the problem is NP-hard [22]. An optimized path with minimum saddle height can be obtained from the *barriers* [6, 8] program due to an exhaustive enumeration of the energy landscape. We will give a more detailed description about the paradigm of *barriers* in Section 6.5. In the particular case of RNA-landscapes, most existing approaches utilize heuristics that consider only direct (minimal length) paths between two configurations [25]. Direct path heuristics perform poorly when the two structures are far apart. The algorithm `Pathfinder` [19] solves this problem via constructing intermediate structures between the two reference structures. Therefore it splits the path construction problem into sub-path construction problems using the direct path heuristic from [10]. Their additional constraint on the sum of the Hamming distances between the intermediate structure and the reference structures allows an adjustable exploration of the underlying energy landscape.

The connection between direct saddles and saddles is elucidated in more detail by the following result. Given a path $P = (v_0, v_1, \dots, v_\ell, v_{\ell+1}) \in G$, if $v_k > v_{k+1} = \cdots = v_{l-1} < v_l$, then all the configurations $v_j \in L$ for $k+1 \le j \le l-1$ are called *valley points*. Analogously, *peak points* are the configurations v_j with $k+1 \le j \le l-1$ if $v_k < v_{k+1} = \cdots = v_{l-1} > v_l$. A path $P = (x = w_0, w_1, \dots, w_\ell, w_{\ell+1} = y)$ is a *zig-zag path* on (G,f) if

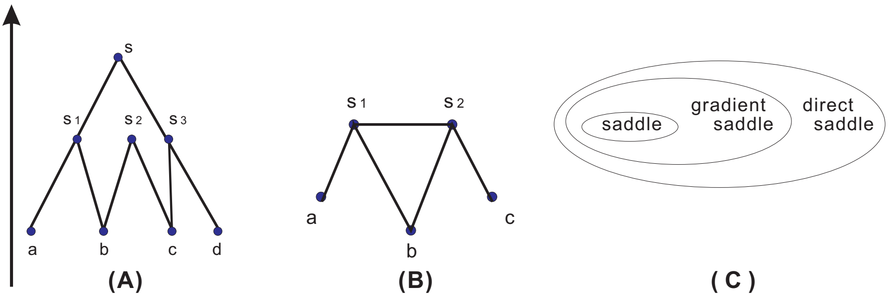


Fig. 6.4 **(A)** Saddles and direct saddles. Given a landscape in which the configuration space consists of $\{a,b,c,d,s_1,s_2,s_3,s\}$, we have $\mathsf{DS}[a,d] = f(s) > \mathsf{S}[a,d] = f(s_1)$. Therefore, s and $\{s_1,s_2,s_3\}$ is the direct saddle and the equivalent class of saddles between a and d, respectively. Furthermore, there exists a relation between saddle height and direct saddle heights given by $\mathsf{S}[a,d] = \mathbf{min}\{\mathbf{max}\{\mathsf{DS}[a,b],\mathsf{DS}[b,c],\mathsf{DS}[c,d]\},\mathsf{DS}[a,d]\}$. **(B)** Direct saddles and gradient saddles. The configurations s_1 and s_2 are the direct saddles between a and c, but there does not exist any gradient saddle between a and c. **(C)** A set diagram of the sets of saddles, direct saddles and gradient saddles of a given landscape.

(1) $\max_i f(w_i) = \mathsf{S}(x,y)$

(2) If $w_k > w_{k+1} = \cdots = w_{l-1} < w_l$ then there is a minimal shelf L such that $w_j \in L$ for $k+1 \leq j \leq l-1$.

(3) If $w_k < w_{k+1} = \cdots = w_{l-1} > w_l$ then each w_j with $k+1 \leq j \leq l-1$ is a direct saddle separating the nearest valley points that the path P passed before and after w_j .

Theorem 6.1. *If $x,y \in V$ are two configurations so that neither $x \rightsquigarrow y$ nor $y \rightsquigarrow x$ then there is a zig-zag path connecting x and y.*

Proof. By construction, $x \xleftrightarrow{\mathsf{S}(x,y)} y$, hence there is a path $\wp$ from x to y whose height does not exceed $\mathsf{S}(x,y)$. Consider the graph $G^* = G/\sim_f$ derived from G by contracting any $G^f(x) \in \Pi$ into a vertex of G^*. In the meanwhile, we obtain a path $\wp^*$ in G^* from $\wp$ accordingly.

To prove the theorem, all we need is to first construct a zig-zag path $P^* \in G^*$ from $\wp^*$ and then prove the existence of a zig-zag path $P \in G$ such that P^* is the resulted graph of P after the contraction. The latter is trivial since by construction, $G^f(x)$ is connected for any $x \in G$. Therefore the proof reduces to the construction of $P^* \in G^*$ from $\wp^*$. This construction is described as follows and illustrated in Figure 6.5.

Let $\{v_i\}_{i=1}^t$ denote the valley points in $\wp^*$. From each valley point v_i, a gradient walk is simulated to reach some local minimum ℓ_i. Without loss of generality, we set $v_0 = \ell_0 = x$, $v_{t+1} = \ell_{t+1} = y$ and assume that all ℓ_i are different configurations. In this context, we observe that there exists a pair of hill-climbing walks from "adjacent" local minima ℓ_i and ℓ_{i+1} to some peak point of $\wp^*$, denoted by p_i. By definition, $f(p_i) \geq \mathsf{DS}[\ell_i,\ell_{i+1}]$. Depending on whether they are equivalent or not, there are two cases. In case of $f(p_i) = \mathsf{DS}[\ell_i,\ell_{i+1}]$, then we just substitute the pair of sections

$([v_i, p_i], [p_i, v_{i+1}])$ in $\wp^*$ into the pair of hill-climbing walks from ℓ_i and ℓ_{i+1} to p_i, respectively. Otherwise, by definition, there must exist a configuration d_i such that $f(d_i) = \mathsf{DS}[\ell_i, \ell_{i+1}] < f(p_i)$. In this case, we substitute the pair of sections $([v_i, p_i], [p_i, v_{i+1}])$ in $\wp^*$ into the pair of hill-climbing walks from ℓ_i and ℓ_{i+1} to d_i, respectively. □

For each saddle point s, the *basin* below s [8] is the set $\mathscr{B}(s) := \mathscr{B}_{f(s)}(s)$ of configurations that can be reached from s by a path along which the energy of the configurations on the path never exceeds $f(s)$. An obvious connection between basins below saddle points and adaptive basins is the following:

$$\mathscr{B}(s) \subseteq \bigcup_{x \in \mathscr{B}(s) \cap M(G,f)} \mathscr{A}(x) \tag{6.6}$$

The analogous result for gradient walks holds only in non-degenerate landscapes.

6.5 Barrier Trees

It is not hard to verify that for any two saddles s' and s'' either $\mathscr{B}(s') \subseteq \mathscr{B}(s'')$, $\mathscr{B}(s'') \subseteq \mathscr{B}(s')$, or $\mathscr{B}(s'') \cap \mathscr{B}(s') = \emptyset$ is satisfied, i.e., the basins below saddles of a landscape give rise to a Hasse diagram with respect to set inclusion order. Given that the landscape is connected, this Hasse diagram can be naturally interpreted as a tree. This *barrier tree* [8, 38], denoted by $T_{\mathscr{B}} := (V_{\mathscr{B}}, E_{\mathscr{B}})$ is defined as follows. Its vertex set $V_{\mathscr{B}}$ comprises the set of local minima $M(G, f)$ and the equivalence classes of saddles with respect to the equivalent relation $\sim_{\mathsf{S}}$, i.e. $s_1 \sim_{\mathsf{S}} s_2 \Leftrightarrow \mathscr{B}(s_1) = \mathscr{B}(s_2)$. As for the edge set $E_{\mathscr{B}}$, two vertices a and $b \in V_{\mathscr{B}}$ are adjacent if $\mathscr{B}(a) \subsetneq \mathscr{B}(b)$ or $\mathscr{B}(b) \subsetneq \mathscr{B}(a)$. In this case we set $\mathscr{B}(m) := \{m\}$ if $m \in M(G, f)$. By construction, local minima and the equivalent classes of saddles are the leaves and the interior vertices in the tree, respectively.

Methods to elucidate the basin structure of landscapes by means of barrier trees have been developed independently in different contexts. For instance, potential energy surfaces for protein folding [13, 37], molecular clusters [4] and the kinetics of RNA landscapes [8].

The barrier tree can by computed by the program `barriers` [6, 8] using a flooding algorithm [30]. The program `barriers` takes an energy sorted list of the K configurations as input. This list may contain either all configurations or only the configurations below some threshold energy. The only part of `barriers` that relies on the geometric properties of the configuration space is the routine that generates all neighbors of each configuration in the list. Therefore, `barriers` has a time complexity of $O(\Delta \times K)$, where Δ denotes the maximum number of neighbors for a configuration in the landscape. To be precise, the program `barriers` processes each configuration on the list consecutively. It identifies a local minimum when it reads one. To identify the saddles, `barriers` labels each configuration x by the lowest local minimum ℓ such that $\ell \in \mathscr{B}_{f(x)}(x)$. Therefore, the configuration s

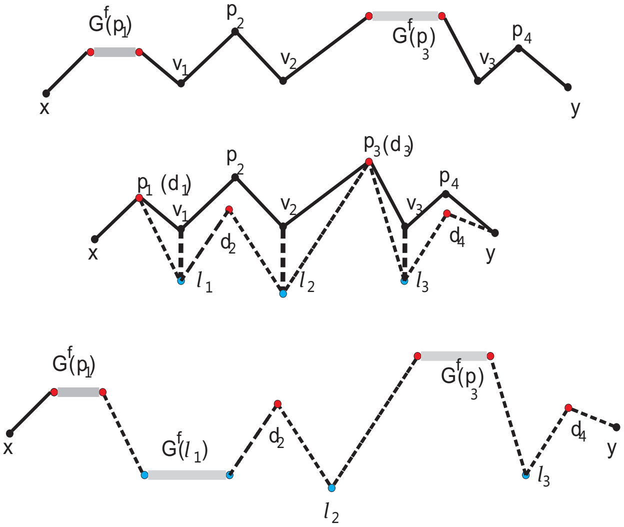


Fig. 6.5 Construction of the path $(\wp \to \wp^* \to P^* \to P)$ in the proof of Theorem 6.1. Bold lines in grey denote the path in $G^f(z)$, $z \in \{p_1, \ell_1, p_3\}$.

processed earliest in the list K which has two neighbors labeled by different local minima ℓ_1 and ℓ_2 is exactly the saddle separating ℓ_1 and ℓ_2.

In the same process, sets of configurations of equal energy are decomposed into connected components. The program then checks for each component whether it is a component of saddle points. For each local minimum ℓ appearing in the barrier tree, the program also records the number of configurations in adaptive basin $\mathscr{A}(\ell)$ as well as the associated restricted partition functions.

6.6 Funnels

The presence of a large number of non-global local minima poses a difficulty for optimization, i.e., identifying movement towards global optima based on purely local information about the landscape. Several measures quantify this difficulty [15] also termed the *ruggedness* of a landscape. Despite being rugged, natural folding landscapes of biopolymers (cf. the example in Section 6.2) allow for fast folding, i.e. a Markov chain quickly hits the global minimum after a relatively short time. The picture of a *funnel* [3] has been used for an arrangement of local minima and saddles that guide dynamics towards the optimum. In the following, we present a rigorous definition of a funnel as the set of configurations that reach the global minimum by iterating exits from gradient basins over the lowest gradient saddle.

Intuitively, we picture a "funneled" dynamics as always taking the lowest transition to another gradient basin and thereby eventually reaching the basin of a ground state. Let $x, y \in M(G, f)$, $r \in \mathscr{G}(x)$ and $s \in \mathscr{G}(y)$. If $e = \{r, s\} \in E$, e is called a

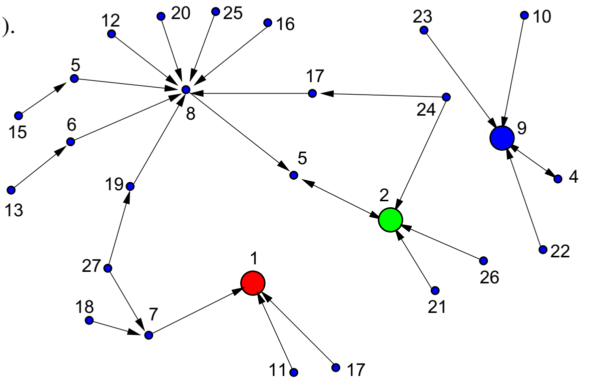


Fig. 6.6 Funnel digraph for the folding landscape of the RNA sequence xbix (CUGCGGCUUUGGCUCUAGCC). The funnel of the landscape contains the local minima $F \cap M = \{1, 7, 11, 17, 18, 27\}$, node 1 being the unique ground state. In the funnel partition of the landscape, the set containing node 2 is the largest. This is consistent with the observation that a large part of the folding trajectories reach the node 2 whose energy lies 0.8 kcal above the energy of the ground state [38]. Figure reproduced from ref. [17].

transition between x and y. The height of a transition e is $h(e) = \max\{f(r), f(s)\}$. We call $T_x \subseteq E$ the set of all transitions from x, i.e.,

$$T_x = \{\{r,s\} \in E : r \in \mathscr{G}(x), s \notin \mathscr{G}(x)\} \tag{6.7}$$

We define the *funnel digraph* with node set $M(G,f)$ and arc set $B \subseteq (M(G,f))^2$ as follows. An arc from x to y exists, $(x,y) \in B$, if there is a transition $e \in T_x$ from x to y with minimal height, i.e., $h(e) \leq h(e')$ for all $e' \in T_x$. Figure 6.6 shows the funnel digraph of an RNA folding landscape.

Now if $e \in T_x$ of minimal height, and e is a transition from x to y, then e contains a direct saddle s between x and y. In fact, we have

$$f(s) = \min\{\mathsf{S}(x,z) : z \in M(G,f)\} \tag{6.8}$$

so s is a lowest saddle for x.

The *funnel* of the landscape is a set $F \subseteq V$ containing those local minima, from which a directed path to a global minimum exists in $(M(G,f),B)$. Equivalently, the funnel may be defined recursively [17] as follows:

(i) F contains all global minima.
(ii) A local minimum $x \in M$ belongs to the funnel F if the funnel digraph contains an edge $\{x,y\} \in B$ with $y \in F$.

Thus $F \cap M$ is the maximal subset of M containing all global minima and fulfilling for all $x, y \in M$:

$$x \notin F \wedge (x,y) \in B \Rightarrow y \notin F \tag{6.9}$$

This determines which local minima belong to F. The funnel is completed by including in F all nodes in the gradient basins of these minima.

In practice, the funnel digraph may be obtained by Algorithm 3. Analogous to `barriers`, it scans the landscape from low to high energy. Each node x is assigned the local minimum $c(x)$ reached from x by a gradient walk, where $c(x) = x$ in the case that x is a local minimum itself. A loop over $y \in N^-$, the neighbors of x prior in the list, checks for each $\{x,y\}$ if it is a transition and potentially updates $h(c(x))$ and $h(c(y))$, the heights of the lowest exits from the gradient basins. Furthermore, if the current height $f(x)$ is at the height of a lowest exit from $c(x)$ an arc from $c(x)$ to $c(y)$ is included in B; analogously an arc $(c(y), c(x))$ is included if $f(x)$ is the lowest exit height from $c(y)$.

Algorithm 3. Computes the arc set B of the funnel digraph

```
Require: A landscape $(G = (V,E), f)$ with neighborhood function $N$.
Require: An energy sorted list $L : \{1,2,\ldots,|V|\} \to V$.
  $B \leftarrow \emptyset$
  for all $i \in \{1,\ldots,|V|\}$ do
    $x \leftarrow L(i)$
    $N^- = N(x) \cap \{L(i)|i < j\}$
    $N^> = \{y \in N(X) : f(x) > f(y)\}$
    if $N^> = \emptyset$ then
      $c(x) \leftarrow x$ // $x$ is a local minimum
      $h(x) \leftarrow +\infty$
    else
      $j^* = \min\{j < i : L(j) \in N^-\}$ // index of gradient neighbor of $x$
      $c(x) \leftarrow c(L(j^*))$
    end if
    for all $y \in N^-$ do
      if $c(y) \neq c(x)$ then
        if $f(x) < h(c(y))$ then
          $h(c(y)) \leftarrow f(x)$
        end if
        if $f(x) < h(c(x)$ then
          $h(c(x)) \leftarrow f(x)$
        end if
        if $f(x) = h(c(y))$ then
          $B \leftarrow B \cup \{(c(y), c(x))\}$
        end if
        if $f(x) = h(c(x))$ then
          $B \leftarrow B \cup \{(c(x), c(y))\}$
        end if
      end if
    end for
  end for
```

After finding the funnel F of the landscape, one may be interested in the landscape outside the funnel. Thus the funnel may be removed and the residual landscape analyzed the same way. Iterating this procedure leads to the *funnel partitioning* of a landscape, being a family $F_1, F_2, \ldots, F_k$. Here $F_1 = F$ is the funnel of the landscape itself and, for all $2 \leq i \leq k$, F_i is the funnel of the landscape on the subgraph induced by $V \setminus \bigcup_{j=1}^{i-1} F_j$.

The identification of funnels relies on identification of the lowest saddles. For applied studies of real landscape instances, exact computation requires enumeration of all configurations. It is thus restricted to small instances [35]. In larger landscapes, saddle heights and transition rates between basins may be approximated by efficient sampling methods [20].

6.7 Landscapes as Combinatorial Vector Fields

6.7.1 Combinatorial Vector Fields on Graphs

Here we consider only the special case of combinatorial vector fields on simple undirected graphs. For the general case we refer to [11].

A *combinatorial vector field* (CVF) on G is a map $\eta : V \to E \cup \{\varnothing\}$ such that, for all $e = \{x,y\} \in E$, $\eta^{-1}(e) \in \{\varnothing, \{x\}, \{y\}\}$.

It is easy to show that CVFs on G are in one-to-one correspondence with the relations $P \subset V \times V$ that satisfies

(1) $(x,y) \in P$ implies $\{x,y\} \in E$ (consistency with G)
(2) $(x,y) \in P$ and $(x,z) \in P$ implies $y = z$ (uniqueness)
(3) $(x,y) \in P$ implies $(y,x) \notin P$ (antisymmetry)

The correspondence is established by $(x,y) \in P$ if and only if $\eta(x) = \{x,y\}$ [31]. We can therefore interpret a CVF as a subset P of directed edges so that each vertex has at most one successor. Note that in contrast to the outgoing arc, the number of incoming arcs is not restricted.

A vertex $x \in V$ is a *rest point* of η if $\eta(x) = \varnothing$, i.e., if x has no successor. The η-trajectory of x is the sequence of v_i, $i \geq 0$, of vertices such $x = v_0$ and $(v_i, v_{i+1}) \in P$. Thus a trajectory either ends in a rest point or it has infinite lengths. In the latter case it contains a finite directed cycle (limit cycle) that is visited infinitely often. The *ω-limit* $\omega_\eta(x)$ of a vertex x is either the (unique) rest point y at which the trajectory starting at x comes to an end, or the limit cycle in which it becomes trapped. Clearly for all $x \in V$, $\omega_\eta(x) \neq \emptyset$ and a vertex y is a rest point if and only if $\omega_\eta(y) = \{y\}$.

The *chain recurrent set* $\mathscr{R}_\eta$ of a combinatorial vector field η on G is defined as

$$\mathscr{R}_\eta = \bigcup_{x \in V} \omega_\eta(x) \tag{6.10}$$

i.e., it consists of the rest points and limit cycles.

Let η be a combinatorial vector field on G. A function $f : V \cup E \to \mathbb{R}$ is a *Lyapunov function* for η if

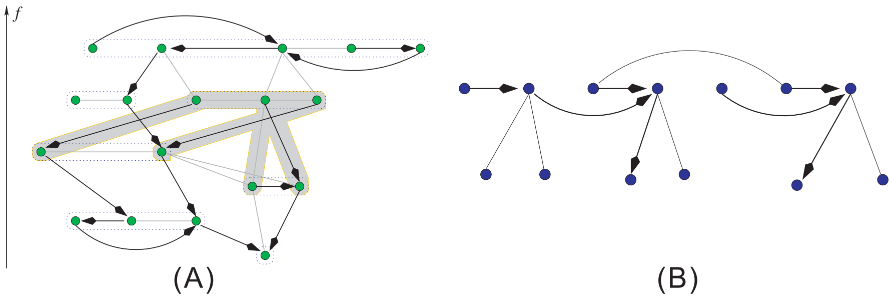


Fig. 6.7 **(A)** Example of a small highly degenerate landscape. Vertices of G are arranged according to the fitness values $f(x)$. Connected components $G^f(x)$ are indicated by dotted boxes. For one of them, the corresponding shelf-graph $\overrightarrow{G^f(x)}$ is highlighted in gray. A combinatorial vector field η (consistent with f) can be visualized as arrows corresponding to the set P_η of oriented edges. **(B)** Combinatorial vector fields on a single shelf satisfied condition (A1).

1. $f(v) \geq f(e) > f(v')$ if $\eta(v) = e$ and $e = \{v, v'\}$ and $v \notin \mathscr{R}_\eta$.
2. $f(v) = f(\eta(v)) = f(v')$ if $v \neq v'$ and v' and v are contained in a cycle.

The basic idea of [31] is now to study adaptive walks in terms of combinatorial vector fields on G for which the prescribed energy function f is a Lyapunov function. It is convenient to introduce the more general compatibility condition

(A1) $\eta(x) = \{x, y\}$ implies $f(x) \geq f(y)$

and to call a function of satisfying (A1) a *weak Lyapunov function* for η. A variety of additional axioms for the consistency of a CVF η on G with a given function f on G are introduced and discussed in detail [31]. We omit these technical details here.

From a technical point of view, the crucial result is that the set of all combinatorial vector fields on G has a product structure: it can be written as the set product of the sets of combinatorial vector fields on each shelf (see Figure 6.7). Since consistency of a CVF with a given (weak) Lyapunov function also boils down to conditions that only refer to the individual shelves separately (see [31] for the technical details). It follows that the set $\mathbb{CVF}(G, f)$ of combinatorial vector fields consistent with f is the direct product of the sets $\mathbb{CVF}(G, f_{|H})$ of combinatorial vector fields consistent with the restriction $f_{|H}$ of f to the individual shelves. The importance of this result is the observation that it is sufficient to understand the admissible combinatorial vector fields on the shelves. In particular, it implies that combinatorial vector fields on locally non-degenerate landscapes are entirely characterized by their behavior on the trivial shelves $\overrightarrow{N^>(x)}$.

6.7.2 Reachability in Combinatorial Vector Fields

The ensemble of CVFs that have f as weak Lyaponov function together describe in a meaningful way all possible adaptive paths on (G, f): every adaptive path alternates between strictly downhill steps that take it from one shelf to the next and neutral paths along which it traverses a shelf from its entry point to an exit point.

Given an ensemble of CVFs, it is natural to define reachability in a different way. We say that y is $\mathbb{CVF}$-reachable from x, $x \rightsquigarrow_{\mathbb{CVF}} y$, if and only if there is a CVF $\eta \in \mathbb{CVF}(G, f)$ so that y is contained in a trajectory starting at x in η. By construction $\mathbb{CVF}$-reachability is a reflexive and transitive, but in general not symmetric, relation. Just like the adaptive walk reachability discussed earlier, it translates to a corresponding reachability topology, a notion of connectedness, and $\looparrowleft\_\looparrowright$ relation, and corresponding concepts of saddle points.

Lemma 6.1. *Let* $\mathbb{CVF}$ *consist of all CVFs for which* f *is a weak Lyapunov function. Then (adaptive walk) reachability and* $\mathbb{CVF}$*-reachability are equivalent.*

Proof. Given an adaptive walk $\wp$ connecting x to y construct a CVF η by defining all vertices not in $\wp$ as rest points of η. Clearly, $x \rightsquigarrow_{\mathscr{C}} y$ and f is a weak Lyapunov function for η. Conversely, if y lies on a trajectory t starting at x in η and f is weak Lyapunov function for η, f is non-increasing along t and hence t is an adaptive walk.

In large shelves, quite complex vector field structures can be consistent with condition (A1) because degenerate fitness functions impose fewer constraints on the combinatorial vector fields. In particular they admit complex recurrent sets within individual sets. These can be restricted by a variety of conditions leading to less inclusive ensembles of CVFs and hence more restricted notions of reachability. We will not delve into these details here and refer to [31] for a more thorough and formal discussion. We note, finally, that the plethora of related saddle-point-like concepts explored in [8] differ primarily in the definition of connected level sets. It has remained unexplored, however, if and how they can be related to variations in the reachability topology.

6.7.3 Path Probabilities

An edge $\{x, y\}$ in the search graph G can be weighted by a measure ω_{xy} of its steepness in the landscape. In energy landscapes, vertices are naturally weighted by their Boltzmann factors $\exp(-\beta f(x))$, where β is the inverse temperature. A natural choice for the edge weights is thus $\omega_{xy} = \exp(\beta |f(x) - f(y)|)$. The "inverse temperature" β tunes our emphasis on steepness and allows to interpolate between adaptive walks and gradient walks. For $\beta = 0$, all valid transitions receive the same weight 1. On the other hand, the steepest edges dominate in each set $N^{>}(x)$ for $\beta \to \infty$. The edge weights carry over to a CVF $\omega(\eta) = \prod_{(x,y)\in\eta} \omega_{xy}$ so that CVFs with more steeper transitions have larger weight. This is formalized by introducing the discrete probability measure $p(\eta) := \omega(\eta)/Z$ where the normalization constant Z plays the role of a partition function on $\mathbb{CVF}(G, f)$.

The product structure of $\mathbb{CVF}(G,f)$ implies that $\omega(\eta)$ is the product of weights for η restricted to individual shelves. This implies, for instance, that the probability $p(x,y)$ of sampling a CVF that contains the prescribed arc (x,y) can be computed by considering the shelf A that contains (x,y) only. Using this observation one can compute recursively the probability $\mathbb{P}\{x \rightsquigarrow y\}$ to draw a combinatorial vector field that contains *an arbitrary* trajectory from x to y, i.e., the probability that y is reachable from x in $\mathbb{CVF}(G,f)$ as detailed in [31]:

$$\begin{aligned}
\mathbb{P}\{x \rightsquigarrow y\} &= \sum_{z\in V(G^f(y))} \tilde{P}(x \rightsquigarrow z) T^{\text{flat}}_{z\rightsquigarrow y} \\
\tilde{P}(x \rightsquigarrow y) &= \sum_{u\in N^>(V(G^f(x)))} T_{x\rightsquigarrow u} \tilde{P}(u \rightsquigarrow y) \\
T_{x\rightsquigarrow y} &= \sum_{w\in D_x} T^{\text{flat}}_{x\rightsquigarrow w} T^{>}_{w\rightarrow y}
\end{aligned} \tag{6.11}$$

Here $\tilde{P}(x \rightsquigarrow y)$ denotes the probability of a path that starts in x and terminates in y such that the final step is a downward step and $T_{x\rightsquigarrow y}$ is the conditional probability that a path exits the shelf A_x through the vertex $z \in V(A_x)^>$ given that it starts in $x \in A^{\text{flat}}_x$. Analogously, $T^{\text{flat}}_{x\rightsquigarrow w}$ describes the walk from x to a drainage point w on the plateau of A_x, and $T^{>}_{w\rightarrow z} = \omega_{wz}/Z_w$ with $Z_w = \sum_{z'\in N^>(w)} \omega_{wz'}$ is the probability to reach z given the path passes through the w.

6.8 Coarse Graining, Macrostates, and Folding Kinetics

The simplest type of kinetics on a landscape (G,f) is a Markov process on the vertex V with transition rates $r_{xy} > 0$ (for the transition from y to x) along the edges of G that depend on the energy difference $f(x) - f(y)$. Most commonly the Metropolis rule $r_{xy} \propto \exp(\beta \min\{0, f(y) - f(x)\})$ [23] or the Kawasaki rule [16] $r_{xy} \propto \exp\left(\frac{1}{2}\beta(f(y) - f(x))\right)$ are used to define the rate constants r_{xy}. Both rules have the Boltzmann distribution $\exp(-\beta f(x))/Z$ as stationary distribution.

A partition $\mathscr{P}$ of the vertex set V induces a coarse graining of the state space G given by the quotient graph $G/\mathscr{P}$. It is not immediately clear, however, how a coarse grained energy function on $G/\mathscr{P}$ should be defined. In [38], a coarse graining of the transition rates has been introduced as follows:

$$r_{XY} = \sum_{x\in X}\sum_{y\in Y} r_{xy}\mathbb{P}[y|Y] \tag{6.12}$$

where $\mathbb{P}[y|Y]$ is the probability of the process to be in state y given that it is in class Y. Assuming local equilibrium on Y, this conditional probability can be approximated by the Boltzmann distribution, yielding the macrostate transition rates

$$r_{XY} = \frac{1}{Z_Y}\sum_{x\in X}\sum_{y\in Y} r_{xy}\exp(-\beta f(y)) \tag{6.13}$$

where $Z_Y = \sum_{y \in Y} \exp(-\beta f(y))$.

In thermodynamic equilibrium, the probability to be in macrostate X is simply the sum of the probabilities of the microstates $x \in X$, i.e., $[X] = Z_X / Z$. Thus, a natural definition of the landscape F on $G/\mathscr{P}$ is the Boltzmann average

$$F(X) = -\beta \ln \sum_{x \in X} \exp(-\beta f(x)) \tag{6.14}$$

In this way, the thermodynamic properties are preserved.

This quotient landscape, however, does not preserve information on the saddle points in (G, f) and hence fails to incorporate the energy barriers and the kinetic properties associated with them. An alternative is to use either the barrier tree or the graph consisting of the local minima and the direct saddle points between them. In the context of the barrier trees the transition rates r_{XY} are approximated by the Arrhenius law [2]

$$r^{\text{Arrh}}_{XY} \propto \exp(-\beta(\mathsf{S}(x,y) - f(y))) \tag{6.15}$$

in terms of the saddle height $\mathsf{S}(x,y)$ between local minima x and y that determine the macrostates X and Y. A somewhat better approximation is achieved by defining r_{XY} via the direct saddles between x and y. A closely related concept is the *inherent structure network* [5] which uses gradient saddles only.

As we have seen in the previous sections, states with large energies typically can reach more than one local minimum by means of adaptive (downhill) walks. Thus it does not appear to be correct to associate a point only with a single local minimum. The issue becomes even more obvious, of course, in landscapes where gradient walks are not unambiguously defined. In [31] it has been proposed, therefore, to allow partial membership. More precisely, the idea is to use $\sum_{y' \in G^f(y)} \mathbb{P}\{x \rightsquigarrow y\}$ to determine the fraction of x that belongs the local minimum represented by $y \in M(G, f)$. This yields e.g. a (typically small) correction to the partition function of the macro states.

6.9 Encodings

Many optimization problems have a natural landscape representation. For instance, the landscape of the traveling salesman problem typically has the permutations of the n cities as the node set and a neighborhood relation based on exchanging two cities or reverting a part of the tour (cf. Section 6.2). This natural setting, however, is not unique, and different representations, even with considerably larger node set for the same problem instance, sometimes turn out as beneficial for optimization [18]. We use the term encoding for a reformulation of an optimization problem by a different landscape in the following sense. Let (G, f) with $G = (V, E)$ be a landscape, $H = (Y, K)$ a graph and $\alpha : Y \to V$ an arbitrary map. Then (H, α) is called an *encoding* of (G, f). We say that the encoding is *complete* if $\alpha(Y) = X$. The encoding is *unbiased* if $|\alpha^{-1}(x)|$ is independent of $x \in X$.

In the context of fitness landscapes for biological evolution, α has an interpretation as a *genotype phenotype map* [34]. The genotype space $H = (Y,K)$ is obtained as the set of genetic sequences Y structured by K, the pairs of sequences that differ by a single mutation. The viability of an individual with a genetic sequence $y \in Y$ depends on its phenotype $\alpha(y) \in V$, i.e., G is the phenotype space to whose elements fitness is assigned by the function f. Variation and selection work in distinct spaces H and G, which are linked by the mapping α.

From an encoding (H,α), we obtain a landscape as $(H, f\circ\alpha)$. If α is unbiased, we have $|\alpha^{-1}(x)| = |Y|/|X|$ for all $x \in V$. Every unbiased encoding is complete.

If the encoding is not unbiased, we would like to quantify if α preferentially maps to low-energy solutions. The cumulative densities $Q_V, Q_Y : \mathbb{R} \to [0,1]$ are defined as

$$Q_V(\eta) = |\{x \in V : f(x) \leq \eta\}| / |X| \tag{6.16}$$

$$Q_Y(\eta) = |\{y \in Y : f\circ\alpha(y) \leq \eta\}| / |Y| \,. \tag{6.17}$$

We say that there is *enrichment* at level $\eta \in \mathbb{R}$ if $Q_Y(\eta) > Q_V(\eta)$. Then, under a random uniform selection of nodes $x \in V$ and $y \in Y$, the probability of encountering a node with energy below η is larger in the encoding than in the original landscape.

Example 6.1. For the traveling salesman problem (cf. Section 6.2) an encoding by prepartitioning the set of cities is introduced in [18] as follows. For a problem instance with n cities, let $Y = [n]^n$ and choose K such that $x,y \in Y$ are adjacent if they differ at exactly one position. A prepartition $y \in Y$ assigns each city a relative position on the tour. A tour $s \in V$ is compatible with $y \in Y$ if, for all $i,j \in [n]$, $y_i < y_j$ implies that city i appears earlier than city j in the tour. For each $y \in Y$, the image $\alpha(y)$ is obtained by greedily constructing a tour under the constraint that it is compatible with y. Two prepartitions $y,z \in Y$ are adjacent, $\{y,z\} \in K$, if there is exactly one index i such that $y_i \neq z_i$. Thus $H = (Y,K)$ is the Hamming graph $H(n,n)$. The encoding (Y,K,α) is complete because, for each permutation $s \in S_n$, the inverse permutation $y = s^{-1}$ lies in Y and fulfills $\alpha(y) = s$. The coding is not unbiased for $n > 2$, because $|Y|/|V| = (n^n)/(n!)$ is not a natural number. As demonstrated in [18], a tour $x \in V$ of short length $f(x)$ tends to have a larger pre-image under α than a long tour. The enrichment is an effect of the greedy choice.

Example 6.2. Ruml. et al. [28] define and numerically analyze several encodings for the *number partitioning problem* [12].

Let us now turn to the local effect of encodings, where we consider the *closed neighborhood* $\bar{N}_V(x) = \{x\} \cup \{z \in V : \{x,z\} \in E\}$ of a node x of the original landscape in relation to a closed neighborhood $\bar{N}_Y(y) = \{y\} \cup \{y \in Y : \{y,z\} \in K\}$ for $y \in Y$ of the encoding. The encoding function $\alpha : Y \to X$ is *continuous* if $\alpha(\bar{N}_Y(y)) \subseteq \bar{N}_V(\alpha(y))$, for all $y \in Y$ and *anti-continuous* if $\bar{N}_V(\alpha(y)) \subseteq \alpha(\bar{N}_Y(y))$.

The definition of continuous maps corresponds to the natural definition of continuity between finite pre-topological spaces. For details we refer to [32]. The inverse condition, here termed anti-continuity, apparently has not been investigated in any detail. Note that (anti-)continuity refers to the map between the configuration spaces only. It is independent of the fitness function.

The reachability topologies on Y and V, on the other hand, contain a lot of information on the fitness function. Let $N^r_Y(y)$ be the reachable neighbors of y as defined by adaptive walks (or the CVFs for which f is a weak Lyapunov function). If $\alpha : Y \to V$ is continuous then $\alpha(N^r_Y(y)) \subseteq N^r_V(\alpha(y))$. Repeating this argument for the $y' \in N^r_Y(y)$ yields $\alpha(N^r_Y(N^r_Y(y))) \subseteq N^r_V(N^r_V(\alpha(y)))$ and, after finitely many iterations, $\alpha(\mathscr{C}_Y(y)) \subseteq \mathscr{C}_V(\alpha(y))$ and thus $\alpha(\mathscr{C}_Y(U)) \subseteq \mathscr{C}_V(\alpha(U))$ for all $U \subseteq Y$. Hence $\alpha : Y \to V$ is also a continuous map w.r.t. to the reachability topologies on Y and V. It remains to explore whether this is also true for alternative definitions of reachability arising from CVFs that have stronger requirements than f being a weak Lyapunov function.

An encoding that is both continuous and anti-continuous in particular preserves local structures. If the encoding is complete and anti-continuous, then the encoded landscape has necessarily larger or equal-sized neighborhoods: $|N_Y(y)| \geq |N(\alpha(y))|$. For these encodings, we obtain statements about the existence of paths and local minima, and bounds on saddle heights.

Lemma 6.2. *Let $f : X \to \mathbb{R}$ be an arbitrary landscape, and let $\alpha : Y \to X$ be an anti-continuous encoding. Then: If $y \in Y$ is a local minimum of the encoded landscape $(H, f \circ \alpha)$, then $\alpha(y)$ is a local minimum on G.*

Proof. We observe that $f(\bar{N}_V(\alpha(y))) \subseteq f(\alpha(\bar{N}_Y(y)))$: y is a local minimum in Y if and only if all function values $z \in f(\alpha(\bar{N}_Y(y)))$ satisfy $z \leq f(\alpha(y))$. Clearly, this implies the same inequality also for the subset $f(\bar{N}_V(\alpha(y)))$.

Lemma 6.3. *Let $P = (x_1, x_2, \dots, x_l)$ be a path in G, and let $\alpha : X \to Y$ be complete and anti-continuous. Then there is a path $\tau = (y_1, \dots, y_l)$ in H such that $\alpha(y_i) = x_i$ for all $i \in [l]$.*

Proof. It suffices to consider a single step in the path, say (x_1, x_2). Since α is complete, there is a $y_i \in Y$ with $\alpha(y_1) = x_1$. Since α is anti-continuous, $\alpha(N(y_1))$ contains all neighbors of x_1, i.e., $\bar{N}_Y(y_1)$ contains in particular a vertex y_2 that satisfies $\alpha(y_2) = x_2$. We simply have to repeat this argument for $x_2, x_3, \dots$ to see that a path in H with the desired property indeed exists.

Since saddles are defined by paths of minimal height, the fact that we find preimages of paths under α directly implies the following.

Corollary 6.1. *Let (H, α) be a complete and anti-continuous encoding of (G, f) and $x, y \in Y$. Then the saddle height between x and y on the encoded landscape is upper bounded by the saddle height between the mapped states on the original landscape:*

$$\mathsf{S}_{(G,f)}(\alpha(x), \alpha(y)) \geq \mathsf{S}_{(H, f\circ\alpha)}(x, y) \tag{6.18}$$

The landscape obtained from an anti-continuous and complete encoding is in a sense smoother and easier to search than the original one, because $(H, f \circ \alpha)$ has fewer local minima and lower saddles connecting the remaining basins. The cost for this improvement is a large search space and an increased size of the local neighborhood.

On the other hand, anti-continuity is not the only property of an encoding that may enhance search on a landscape. Example 6.1 above and other encodings studied in [18] are not anti-continuous. However, adaptive walks on the encoded landscape outperform adaptive walks on the original landscape at long times.

6.10 Concluding Remarks

The theory of combinatorial landscapes has emerged as an attempt to devise suitable mathematical structures for describing the "static" properties of landscapes as well as their influence on the kinetic dynamics of adaptation. The static point of view focuses on geometric properties such as basins, barriers, plateaus, CVFs, etc. As described in Section 6.8, there exists a close relation between the static properties and kinetic properties of combinatorial landscapes. This chapter is our attempt to clarify various geometric concepts that have been introduced in this field and discuss their relations with the kinetic properties of the landscapes.

Acknowledgements. P.F.S. and K.K. acknowledge financial support from VolkswagenStiftung.

References

[1] Binder, K., Young, A.P.: Spin glasses: Experimental facts, theoretical concepts, and open questions. Rev. Mod. Phys. 58, 801–976 (1986)

[2] Chan, H., Dill, K.: Protein folding in the landscape perspective: Chevron plots and non-Arrhenius kinetics. Proteins: Structure, Function, and Genetics 30, 2–33 (1998)

[3] Dill, K.A., Chan, H.S.: From levinthal to pathways to funnels. Nat. Struct. Biol. 4, 10–19 (1997)

[4] Doye, J.P., Miller, M.A., Welsh, D.J.: Evolution of the potential energy surface with size for Lennard-Jones clusters. J. Chem. Phys. 111, 8417–8429 (1999)

[5] Doye, J.P.K.: Network topology of a potential energy landscape: A static scale-free network. Phys. Rev. Lett. 88, 238701 (2002)

[6] Flamm, C., Fontana, W., Hofacker, I., Schuster, P.: RNA folding kinetics at elementary step resolution. RNA 6, 325–338 (2000)

[7] Flamm, C., Hofacker, I.L.: Beyond energy minimization: Approaches to the kinetic folding of RNA. Chemical Monthly 139, 447–457 (2008)

[8] Flamm, C., Hofacker, I.L., Stadler, P.F., Wolfinger, M.T.: Barrier trees of degenerate landscapes. Z. Phys. Chem. 216, 155–173 (2002)

[9] Flamm, C., Hofacker, I.L., Stadler, B.M.R., Stadler, P.F.: Saddles and barrier in landscapes of generalized search operators. In: Stephens, C.R., Toussaint, M., Whitley, L.D., Stadler, P.F. (eds.) FOGA 2007. LNCS, vol. 4436, pp. 194–212. Springer, Heidelberg (2007)

[10] Flamm, C., Stadler, P., Maurer-Stroh, S., Zehl, M., Hofacker, I.L.: Design of multistable RNA molecules. RNA 7, 254–265 (2001)

[11] Forman, R.: Combinatorial vector fields and dynamical systems. Math. Z. 228, 629–681 (1998)

[12] Garey, M., Johnson, D.: Computers and Intractability. A Guide to the Theory of NP Completeness. Freeman, San Francisco (1979)

[13] Garstecki, P., Hoang, T.X., Cieplak, M.: Energy landscapes, supergraphs, and "folding funnels" in spin systems. Phys. Rev. E 60, 3219–3226 (1999)

[14] Hofacker, I.L.: Vienna RNA secondary structure server. Nucl. Acids Res. 31(13), 3429–3431 (2003)

[15] Kallel, L., Naudts, B., Reeves, C.R.: Properties of fitness functions and search landscapes. In: Kallel, L., Naudts, B., Rogers, A. (eds.) Theoretical Aspects of Evolutionary Computing, pp. 175–206. Springer (2001)

[16] Kawasaki, K.: Diffusion constants near the critical point for time-dependent ising models. Phys. Rev. 145, 224–230 (1966)

[17] Klemm, K., Flamm, C., Stadler, P.F.: Funnels in energy landscapes. Eur. Phys. J. B 63, 387–391 (2008)

[18] Klemm, K., Mehta, A., Stadler, P.F.: Landscape encodings enhance optimization. PLoS One 7, e34780 (2012)

[19] Lorenz, R., Flamm, C., Hofacker, I.L.: 2D projections of RNA folding landscapes. In: Grosse, I., Neumann, S., Posch, S., Schreiber, F., Stadler, P.F. (eds.) GCB. LNI, vol. 157, pp. 11–20. G (2009)

[20] Mann, M., Klemm, K.: Efficient exploration of discrete energy landscapes. Phys. Rev. E 83(1), 011113 (2011)

[21] Mathews, D., Sabina, J., Zuker, M., Turner, D.H.: Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. J. Mol. Biol. 288, 911–940 (1999)

[22] Maňuch, J., Thachuk, C., Stacho, L., Condon, A.: NP-completeness of the energy barrier problem without pseudoknots and temporary arcs. Natural Computing 10(1), 391–405 (2011)

[23] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H.: Equation of state calculations by fast computing machines. J. Chem. Phys. 21, 1087–1092 (1953)

[24] Mezey, P.G.: Potential Energy Hypersurfaces. Elsevier, Amsterdam (1987)

[25] Morgan, S.R., Higgs, P.G.: Barrier heights between ground states in a model of RNA secondary structure. J. Phys. A: Math. Gen. 31, 3153–3170 (1998)

[26] Rammal, R., Toulouse, G., Virasoro, M.A.: Ultrametricity for physicists. Rev. Mod. Phys. 58, 765–788 (1986)

[27] Reidys, C.M., Stadler, P.F.: Combinatorial landscapes. SIAM Review 44, 3–54 (2002)

[28] Ruml, W., Ngo, J., Marks, J., Shieber, S.: Easily searched encodings for number partitioning. J. Opt. Th. Appl. 89, 251–291 (1996)

[29] Schuster, P., Fontana, W., Stadler, P.F., Hofacker, I.L.: From sequences to shapes and back: A case study in RNA secondary structures. Proc. Roy. Soc. Lond. B 255, 279–284 (1994)

[30] Sibani, P., van der. Pas, R., Schön, J.C.: The lid method for exhaustive exploration of metastable states of complex systems. Computer Physics Communications 116, 17–27 (1999)

[31] Stadler, B.M.R., Stadler, P.F.: Combinatorial vector fields and the valley structure of fitness landscapes. J. Math. Biol. 61(6), 877–898 (2010)

[32] Stadler, B.M.R., Stadler, P.F., Shpak, M., Wagner, G.P.: Recombination spaces, metrics, and pretopologies. Z. Phys. Chem. 216, 217–234 (2002)

[33] Stadler, P.F., Flamm, C.: Barrier trees on poset-valued landscapes. Genetic Prog. Evolv. Mach. 4, 7–20 (2003)

[34] Stadler, P.F., Stadler, B.M.R.: Genotype phenotype maps. Biological Theory 3, 268–279 (2002); Konrad Lorenz Institute Workshop on Biological Information organized by Werner Callebaut in 2002

[35] Tomassini, M., Vérel, S., Ochoa, G.: Complex-network analysis of combinatorial spaces: The NK landscape case. Phys. Rev. E 78, 066114 (2008)

[36] Van Nimwegen, E., Crutchfield, J.P.: Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths? Bull. Math. Biol. 62, 799–848 (2000)

[37] Wales, D.J.: Decoding the energy landscape: extracting structure, dynamics and thermodynamics. Phil. Trans. R. Soc. A 370, 2877–2899 (1969)

[38] Wolfinger, M.T., Svrcek-Seiler, W.A., Flamm, C., Hofacker, I.L., Stadler, P.F.: Exact folding dynamics of RNA secondary structures. J. Phys. A: Math. Gen. 37, 4731–4741 (2004)

[39] Wright, S.: The roles of mutation, inbreeding, crossbreeeding and selection in evolution. In: Jones, D.F. (ed.) Proceedings of the Sixth International Congress on Genetics, vol. 1, vol. 1, pp. 356–366. Brooklyn Botanic Gardens, New York (1932)

Chapter 7
Polytopes, Graphs and Fitness Landscapes

Kristina Crona

Abstract. Darwinian evolution can be illustrated as an uphill walk in a landscape, where the surface consists of genotypes, the height coordinates represent fitness, and each step corresponds to a point mutation. Epistasis, roughly defined as the dependence between the fitness effects of mutations, is a key concept in the theory of adaptation. Important recent approaches depend on graphs and polytopes. Fitness graphs are useful for describing coarse properties of a landscape, such as mutational trajectories and the number of peaks. The graphs have been used for relating global and local properties of fitness landscapes. The geometric theory of gene interaction, or the shape theory, is the most fine-scaled approach to epistasis. Shapes, defined as triangulations of polytopes for any number of loci, replace the well established concepts of positive and negative epistasis for two mutations. From the shape one can identify the fittest populations, i.e., populations where allele shuffling (recombination) will not increase the mean fitness. Shapes and graphs provide complementary information. The approaches make no structural assumptions about the underlying fitness landscapes, which make them well suited for empirical work.

7.1 Introduction

The fitness landscape was originally intended as a simple metaphor for an intuitive understanding of adaptation [38]. Adaptation can be pictured as an uphill walk in the fitness landscape, where height represents fitness and where each step is between similar genotypes. The concept of a fitness landscape has been formalized in somewhat different ways [4] and the current theory is extensive. Kaufman's NK model [20], block models [24, 28], as well as random (rugged or uncorrelated) fitness landscapes [12, 19, 21, 30, 33] have been especially influential in biology. Early work in the field was primarily motivated by theoretical considerations, such as the relation between global and local properties of fitness landscapes. However, it may not be

Kristina Crona
University of California at Merced, 5200 Lake Road, Merced, CA 95343, U.S.A.
e-mail: kcrona@ucmerced.edu

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_7, © Springer-Verlag Berlin Heidelberg 2014

clear if the classical models apply in a particular empirical context. The underlying assumptions, such as a block structure of the fitness landscape, may or may not hold.

Some recent approaches do not make any structural assumptions about the fitness landscapes. We will consider the geometric theory of gene interactions and fitness graphs. We define fitness as the logarithm of the expected reproductive success. There are different definitions of fitness in the literature [25]. Epistasis means that fitness is not linear. For instance, the combination of two beneficial mutations may result in a double mutant with much higher fitness, as compared to a linear expectation from the fitness of the wild-type, and the two single mutants. Such positive epistasis is common for drug resistance mutations, for example antibiotic resistance mutations [17]. It is not difficult to analyze the two-loci case, but it is less obvious how to quantify, classify and interpret epistasis for several loci.

The most fine-scaled approach to gene interactions is the recently developed geometric theory [3]. The theory extends the usual concept of epistasis for two mutations to any number of loci in the strict sense that all gene interactions are reflected. The shapes, as defined in the geometric theory, has the role of positive and negative epistasis for two mutations.

In contrast to the sensitive shapes, a fitness graph is determined by the fitness ranks of the genotypes only. Qualitative information such as if "good+good=better" or "good+good= not good" for two single mutations are reflected by the fitness graphs. From the graphs one can immediately understand the coarse properties of the landscapes, including the number of peaks. We argue that both the geometric theory and fitness graphs are well suited for empirical work. Moreover, to some extent shapes and fitness graphs provide complementary information. Shapes are relevant for recombination and fitness graphs for mutational trajectories.

In many real populations at most two alternative alleles occur at each locus, or a biallelic assumption is a reasonable simplification. Throughout the chapter, we will consider biallelic L-loci populations. Let $\Sigma = \{0,1\}$ and let Σ^L denote bit strings of length L. Σ^L represents the genotype space. In particular,

$$\Sigma^2 = \{00,01,10,11\} \text{ and } \Sigma^3 = \{000,001,010,011,100,101,110,111\}.$$

The *zero-string* denotes the string with zero in all L positions, and the *1-string* denotes the string with 1 in all L positions. We define a *fitness landscape* as a function $w : \Sigma^L \mapsto \mathbb{R}$, which assigns a fitness value to each genotype. The fitness of the genotype g is denoted w_g. The metric we consider is the Hamming distance, meaning that the distance between two genotypes equals the number of positions where the genotypes differ. In particular, two genotypes are adjacent, or *mutational neighbors*, if they differ at exactly one position.

A walk in the fitness landscape corresponds to a Darwinian process in a precise way. Consider a monomorphic population, i.e., a population where all individuals have the same genotype, after a recent change in the environment. Such a genotype is a called the *wild-type*. Assume that in the new environment the wild-type no longer has optimal fitness. Under the assumption of the strong-selection weak-mutation regime (SSWM), a beneficial mutation will go to fixation in the population before the next mutation occurs. It follows that the population is monomorphic for most of the time. The adaptation process can be described as a sequence of genotypes, all of which became fixed in the population at some point in time.

For instance, let 00 denote the wild-type, assume that the single mutants 10 an 01 have higher fitness than the wild-type, and that the double mutant 11 has the highest fitness of the four genotypes. The two possible adaptation scenarios for a population are

$$00 \mapsto 10 \mapsto 11 \text{ and } 00 \mapsto 01 \mapsto 11.$$

Each scenario corresponds to an uphill walk, which ends at the genotype 11. The example illustrates that we can think of a Darwinian process as a walk in the fitness landscape, where each step represents a beneficial mutation going to fixation in the population. Adaptation is not deterministic, but fitness has to increase by each step. The described model of adaptation has been widely used and relies on approaches developed in [15, 16, 27].

The chapter is structured as follows. The topic for Section 7.2 to 7.5 is fitness graphs, where most results depend on [6]. The topic for Section 7.6 to 7.10 is the geometric theory of gene interactions, where most results depend on [3], and triangulations of polytopes [9]. Section 7.11 compares fitness graphs and shapes, as defined in the geometric theory. Section 7.12 is a discussion. For more background, including proofs, we refer to [3, 6, 9].

7.2 Fitness Graphs and Sign Epistasis

With reference to the landscape metaphor, an *adaptive step* in the fitness landscape corresponds to a change in exactly one position of a string so that the fitness increases strictly. An *adaptive walk* is a sequence of adaptive steps. A *peak* in the fitness landscape has the property that there are no adaptive steps away from it, i.e., a genotype is at a peak if all mutational neighbors have lower fitness as compared to the genotype. The following concepts are central as well, in particular they are useful for relating the number of peaks to local observations.

For $L \geq 2$, given a string and two positions, exactly four strings can be obtained which coincide with the original string except (at most) at the two positions. Denote such a set of four strings

$$ab, Ab, aB, AB,$$

according to the two positions of interest, and assume that w_{ab} is minimal. *Sign epistasis* means that

$$w_{AB} < w_{Ab} \text{ or } w_{AB} < w_{aB}.$$

Reciprocal sign epistasis interactions means that

$$w_{AB} < w_{Ab} \text{ and } w_{AB} < w_{aB}.$$

Figure 7.1 shows the four possibilities under our assumption that w_{ab} is minimal.

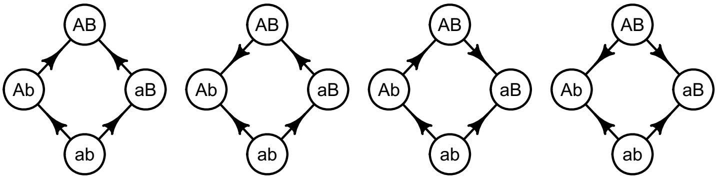


Fig. 7.1 The arrows point toward the more fit genotype. The graphs represent no sign epistasis, two cases with sign epistasis but not reciprocal sign epistasis, and one case with reciprocal sign epistasis.

Sign epistasis is by no means rare for microbes according to several studies [2, 11, 13, 17, 35, 36, 37]. In particular, sign epistasis occurs for antibiotic resistance mutations, as well as for HIV and malaria. In fact, existing studies suggest that *absence* of sign epistasis is exceptional for systems associated with drug resistance for $L \geq 4$.

Sign epistasis is of clinical importance for several reasons. A recent approach for preventing and managing resistance problems takes advantage of both sign epistasis and variable selective environments [17]. Another aspect of managing drug resistance is to find constraints for orders in which mutations accumulate from genotype data [2, 11]. A constraint could be that a particular mutation is selected for (meaning that that the mutation is beneficial) only if a different mutation has already occurred. The existence of constraints implies sign epistasis. Indeed, if a particular mutation is beneficial regardless of background, then it can occur before or after other mutations. Moreover, sign epistasis is relevant for predictions of how populations will adapt [37].

Fitness graphs are useful for the empirical problems mentioned, as well as for more theoretical problems, including the relation between global and local properties of fitness landscapes (see Section 7.3). If one can order a set of genotypes by decreasing fitness, one has determined the *fitness ranks*. More fine scaled information, such as relative fitness values, may not be known. A fitness graph compares the fitness ranks of mutational neighbors. For simplicity, whenever we use fitness graphs we assume that $w_s \neq w_{s'}$ for any two strings s and s' which differ in one position only.

Roughly, consider the zero-string as the starting point (possibly the wild-type), and each non-zero position of a string as an event, i.e., that a mutation has occurred. Under these assumptions the fitness graph coincides with the Hasse-diagram of the power set of events, except that each edge in the Hasse-diagram is replaced with an arrow toward the string with greater fitness.

For a formal definition, a fitness graph is a directed graph where each node corresponds to a string of Σ^L. The fitness graphs has $L+1$ levels. Each string such that $\sum s_i = l$ corresponds to a node on level l in the fitness graph. In particular, the node representing the zero-string is at the bottom, the nodes representing strings with exactly one non-zero position, including $10\cdots0$, are one level above, the nodes representing strings with exactly two non-zero positions, including $110\cdots0$, are on the next level, and the 1-string is at the top. Moreover, the nodes are ordered from left to right according to the lexicographic order where $1 > 0$ of the corresponding strings (see e.g. Figure 7.5). A directed edge connects each pair of nodes such that the corresponding strings differ in exactly one position. The edge is directed toward the node representing the more fit of the two genotypes.

Remark 1: *Unless otherwise stated, the words "level", "up", "down" "above" and "below" refer to fitness graphs. In particular, notice that a higher level does not imply greater fitness.*

For $L \geq 2$, given a string and two positions, consider the four strings which coincide with the original string except in (at most) the two positions. We call the strings a *type 2* system if there is reciprocal sign epistasis, a *type 1* system if there is sign epistasis, but not reciprocal sign epistasis, and a *type 0* system if there is no sign epistasis.

For interpretations of general fitness graphs, it may be helpful to first analyze the two-loci case shape in some detail. There exist exactly 14 fitness graphs for biallelic two-loci systems (see Figure 7.2), where 4 are type 0 systems, 8 type 1 systems, and 2 type 2 systems. One verifies the following result.

Remark 2: *For two-loci, type 0, 1, and 2 systems have the following properties:*

(1) A type 0 system can be rotated so that all arrows point up.
(2) A type 1 system differs from a cycle by exactly one arrow.
(3) A type 2 system have two nodes such that all edges are directed toward them, and two nodes such that no edges are directed toward them.

The observations from the two-loci case should make it easy to identify type 0, 1 and 2 systems for general fitness graphs. Figures 7.3 and 7.4 show fitness graph for 3-loci systems. Figure 7.3a has type 0 systems only, Figure 7.3b type 0 and 2

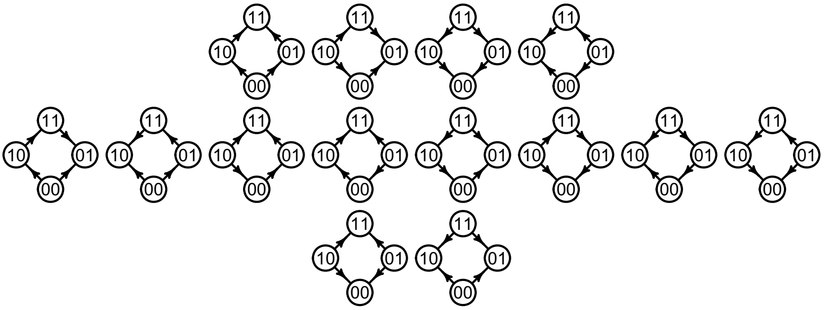
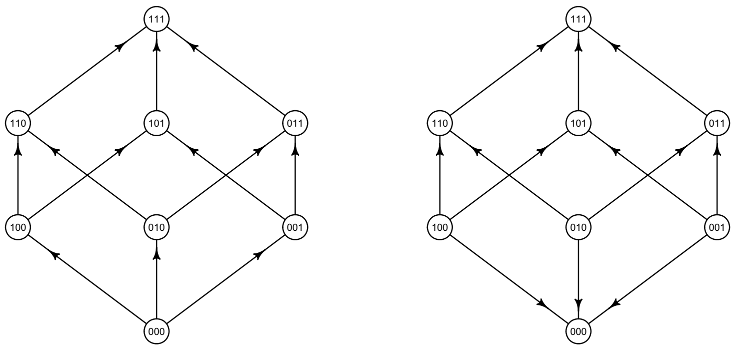
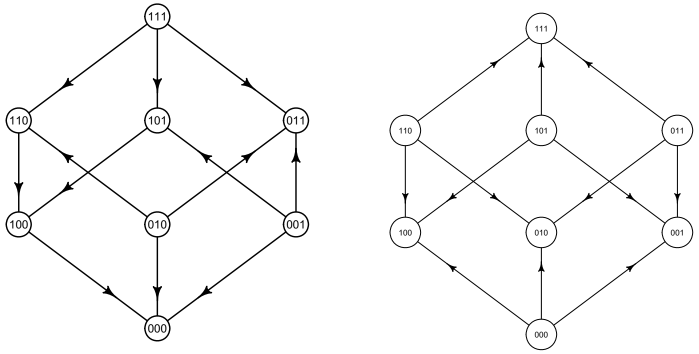


Fig. 7.2 For a fitness graph, the arrows point toward the genotype of greater fitness. There exist exactly 14 fitness graphs for biallelic two-loci systems, where the type 0 systems are on the first row, the type 1 systems on the second row, and the type 2 systems on the third row.



(a) (b)

Fig. 7.3 A fitness graph shows sign epistasis and the peaks. The graph in Figure 7.3a has type 0 systems only. The graph in Figure 7.3b has type 0 and type 2 systems, but no type 1 systems.



(a) (b)

Fig. 7.4 The graph in Figure 7.4a has type 0, type 1 and type 2 systems. The graph in Figure 7.4b has type 2 systems only, and the corresponding fitness landscape has four peaks.

systems, Figure 7.4a type 0, 1 and 2 systems, and Figure 7.4b type 2 systems only. Figure 7.5 shows a fitness graph for a 4-loci population, where there are several type 2 systems, including 0001, 0101, 0011, 0111.

7.3 Fitness Graphs and Theoretical Results

Fitness graphs have mostly been used in empirical work [10, 13, 17, 35]. However, we will indicate how they can be used in theoretical arguments, and mention some results where the proofs depend on fitness graphs.

It is known that one can have 2^{L-1} peaks in a fitness landscape [18] and this number is an upper bound. The proof is elementary, and we will not give the details. However, we will construct fitness landscapes with the maximal number of peaks using fitness graphs.

Example 1: *For any L, consider the fitness graph where the edges are directed up from level 0 to 1, down from level 1 to 2, up from level 2 to 3, and so on. The fitness graph in Figure 7.4b is an example. Notice that the graph corresponds to fitness landscapes with 4 peaks, i.e., the maximal number of peaks for* $L = 3$. *In general, all nodes at level* $1, 3, 5 \ldots$ *are at peaks, and such fitness graphs correspond to fitness landscapes with exactly* 2^{L-1} *peaks.*

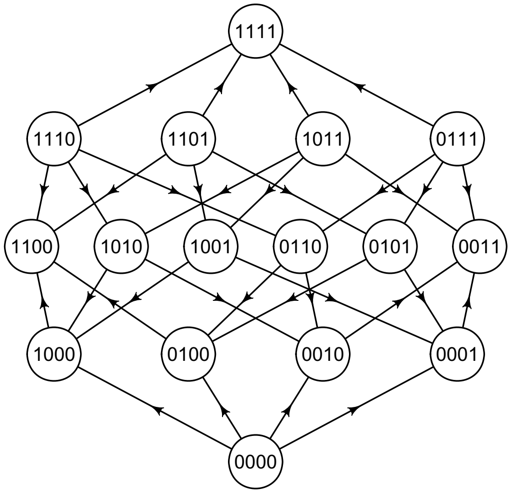


Fig. 7.5 The fitness landscape has peaks at 1100, 0011 and 1111, whereas all triple mutants (mutants on the third level) have low fitness

Recent work relates global and local properties of fitness landscapes [6, 31, 32]. This topic is of interest, since most empirical studies of fitness landscapes concern local properties, including sign epistasis. It has been shown that multipeaked fitness landscapes have type 2 systems [32]. The converse is not true. However, a sufficient condition for multiple peaks can be phrased in terms of type 1 and 2 systems. More precisely, the following result was proved using fitness graphs.

Result 1: Crona et al., 2013 [6]

If a fitness landscape has type 2 systems and no type 1 systems, then it has multiple peaks.

It follows that the landscapes corresponding to Figures 7.3b and 7.4b have multiple peaks. Fitness graphs are efficient for analyzing mutational trajectories. We will state a result regarding accessible mutational trajectories from [36]. A brief proof of the result using fitness graphs was given in [6], but the original proof does not use fitness graphs.

We call the global maximum of the landscape “the fitness peak”. Moreover, define a *general step* similar to “adaptive step”, except that the fitness may decrease. A *general walk*, as opposed to an “adaptive walk” is a sequence of general steps. If a general walk between two nodes has minimal length, we call it a *shortest walk*.

Result 2: Weinreich et al., 2005 [36]

(1) *The following conditions are equivalent for a fitness landscape.*

- (i) *Each general step toward the fitness peak, i.e., a step that decreases the graph theoretical distance to the peak, is an adaptive step.*
- (ii) *Each shortest general walk to the fitness peak is an adaptive walk.*
- (iii) *The fitness landscape has no type 1 or 2 systems.*

(2) *If the equivalent conditions (i), (ii) and (iii) in (1.) are satisfied, then each adaptive walk to the fitness peak is a shortest general walk.*

A fitness landscape satisfying the equivalent conditions (i)–(iii) above is referred to as *a fitness landscape lacking genetic constraints on accessible mutational trajectories* in [36]. For $L = 3$, the fitness graph in Figure 7.3a corresponds to this category of landscapes. Fitness landscapes lacking genetic constraints on accessible mutational trajectories can be represented by fitness graphs where all arrows are up. For brevity, we will refer to “all arrows up landscapes”.

It is important to notice that the concept of an all arrows up landscape is biologically meaningful. Even if a landscape is single peaked, type 1 systems may cause the adaptation process to be slower since not all shortest general walks to the peak are adaptive walks. However, for all arrows up landscapes, there are no local obstacles for the adaptation process.

7.4 Fitness Graphs and Recombination

Recombination can generate new genotypes in a population. Under some circumstances, recombination will speed up adaptation. An early hypothesis about the possible advantage of recombination concerned double mutants of high fitness, where the corresponding single mutations are deleterious. It was suggested that recombination could generate such double mutants. In terms of fitness graphs this case can be described as a type 2 system, where the wild-type is at a fitness peak. However, the hypothesis was immediately criticized, and described as a “widespread fallacy” by Muller [8]. The two single mutations being deleterious, it seems unlikely that the the corresponding genotypes would appear and recombine to the double mutant. The (current) consensus is that under most circumstances recombination will not be of any use in the situation described, i.e., for a two-loci type-2 system, where the wild-type is at a peak (see also [23]). However, using fitness graphs we will argue that recombination could be an advantage in somewhat related cases where $L \geq 3$.

The topic of recombination is involved with subtle differences between effects on the population level and the gene level. For instance, it is theoretically possible that recombination is beneficial for a population and at the same time recombination suppressors could be selected for (see e.g. [29] for comments and references). We do not intend to develop new theory, or describe existing knowledge of recombination in any detail. For an overview of the field, we refer to Otto and Lenormand [29]. Our goal is to point out mechanisms specific for $L \geq 3$ loci which should be considered for an analysis of the effect of recombination. This is justified since the field is dominated by work in the two-loci case, or mechanisms which can be reduced to the two-loci case.

It has been suggested that recombination has an especially strong impact in *structured populations*, see e.g. [26]. A populations is structured, as opposed to well mixed, if the genotype frequencies varies between geographic locations. In particular, if a population is subdivided into local subpopulations with some migration between them, then recombination could be advantageous.

We will sketch a model within this framework, which we call a *puddles and flood population*. We mainly have microbes in mind, for example bacteria. Assume that the local subpopulations live in puddles, and the subpopulations are homogeneous for most of the time. Occasionally, there is a flood where the contents of the local puddles get thoroughly mixed. After a flood, life proceeds as usual in the puddles for an extended period, until the next flood. Under these assumptions, genotypically different subpopulations are likely to mix, so that recombination can generate new genotypes.

Example 2: *Consider the fitness graph in Figure 7.5, and assume that 0000 is the wild-type. Both 1100 and 0011 are at peaks, whereas the triple mutants are less fit as compared to adjacent double mutants. For a puddles and flood population, recombination of double mutants may result in 1111. In this case recombination could speed up adaptation*

Notice that in the absence of recombination, one could obtain 1111 from 1100, only if there is a double mutation, since the triple mutants are not fit.

Example 3: *Consider the fitness graph in Figure 7.4b. Assume that 000 is the wild-type. From the fitness graph, the single mutants 100, 010, 001 are at peaks. Under the assumption that 111 has maximal fitness, recombination could speed up adaptation. However, two recombination events are necessary. For instance, recombination of 100 and 010 could result in 110. Then recombination of 110 and 001 could result in 111.*

Notice that there is an important difference between Examples 2 and 3. For instance, consider the outcome for a puddles and flood population where no more than two puddles mix at the time. Then one could obtain 1111 by recombination in Example 2. Indeed, if an 1100 population and an 0011 population mix, recombination could produce the 1111 genotype.

In contrast, consider Example 3 under the same restriction (no more than two puddles mix). If say an 100 and 010 population mix, one could obtain 110 by recombination. However, 110 is selected against so that 111 is unlikely to appear under most circumstances. (On the other hand, if *several* puddles mix, one could get a mixture of the single mutants 100, 010 and 001, and recombination could result in 111).

Consider all arrows up fitness landscapes where the 1-string has maximal fitness. Then one could obtain the 1-string from a sequence of single mutations. However, for a puddles and flood population, recombination could speed up adaptation. This is because the process of accumulating L single mutations could be time consuming.

Example 4: *For an all arrows up L-loci fitness landscape where considerably more than L puddles tend to mix during a flood period, one could obtain the 1-string already after one flood period.*

The examples described are theoretical constructions. It is not obvious if Examples 2 and 3, or similar cases, occur frequently enough in nature for having much of an impact. A first question to ask for a population, is how frequently it happens that "good+good=not good" for single mutations. This type of problems is the topic for the next section.

7.5 Fitness Graphs and Other Qualitative Measures

In order to determine if one has a reasonable chance to find fitness graphs of the types described in the previous sections, the following qualitative concept [6] may be useful.

We define B and B_p as follows. *The set B_p consists of all double mutants such that both corresponding single mutations are beneficial. The set $B \subseteq B_p$ consists of all double mutants in B_p which are more fit than at least one of the corresponding single mutants.* The *qualitative measure of additivity* for a fitness landscape is the ratio $\frac{|B|}{|B_p|}$. Notice that $\frac{|B|}{|B_p|} = 1$ for all arrows up landscapes.

Fitness landscapes are defined as *additive* if fitness effects of mutations sum. For example, if

$$w_{00} = 1, w_{10} = 1.2, w_{01} = 1.3,$$

then additive fitness implies that $w_{11} = 1.5$ (since $0.5 = 0.2 + 0.3$, so that the fitness effects of two mutations sum). Notice that fitness is additive exactly if

$$w_{11} - w_{10} - w_{01} + w_{00} = 0\,.$$

By definition, fitness is additive exactly if there is no epistasis. One may consider all arrows up landscapes as the qualitative correspondence to additive fitness landscapes.

Antibiotic resistance landscapes for a particular 4-loci system and 9 selective environments were studied in [17]. More precisely, all combination of the TEM-1 mutations L21F, R164S, T265M and E240K were considered. The length of TEM-1 is 287, i.e., TEM-1 can be represented as a sequence of 287 letters in the 20-letter alphabet corresponding to the amino acids. The notation L21F means that the amino acid Leucine (L) at position 21 has been replaced by the amino acid Phenylalanine (F). The mutations R164S, T265M and E240K are defined similarly, using the standard notation for amino acids. The mean value of $\frac{|B|}{|B_p|}$ for the 9 selective environments was 0.57.

In contexts where the relative fitness values of genotypes are not known, qualitative concepts can still be used. It is valuable to understand qualitative information for several reasons. Fitness ranks tend to be easier to determine as compared to relative fitness, and from records of mutations one can sometimes draw conclusions about fitness ranks without making measurements. We argue that much can be learned about fitness landscape from existing records of mutations, in particular from drug resistance mutations. However, one needs to be able to interpret qualitative information, a theme developed in [7] with applications to antibiotic resistance, see also [6].

The qualitative measure of additivity is coarse. If relative fitness values can be determined, one may want to consider quantitative fitness differences as well. For a quantitative measure of additivity, we refer to the concept “roughness” [1, 5]. Briefly, additive fitness landscapes have roughness 0, and any deviation from additivity implies that the roughness is greater than 0.

7.6 Shapes

Throughout the remainder of the chapter, we consider biallelic L-loci populations, where we assume that all 2^L genotypes occur in the populations (for a comment regarding this simplification, see Section 7.10), and a fitness landscape

$$w : \Sigma^L \mapsto \mathbb{R}, \Sigma = \{0,1\}.$$

Moreover, for $L = 2$ and 3 we use the following orders of the genotypes (from left to right):

$$00, 01, 10, 11 \text{ and } 000, 001, 010, 011, 100, 101, 110, 111.$$

Most empirical studies of epistasis for several loci focus on the average curvature (Figure 7.6) or pairwise gene interactions using ANOVA methods [3]. For beneficial mutations, *antagonistic epistasis* means that the combined effect of mutations are less than the sum of individual effects, whereas *synergistic epistasis* means that the combined effect of mutations exceeds the sum of individual effects. It has been claimed that antagonistic epistasis dominates for beneficial mutations in nature, see e.g. [22]. Antagonistic and synergistic epistasis are defined analogously for deleterious mutations (Figure 7.6). A motivation for the interest in average effects of mutations is the connection to recombination. According to standard models, antagonistic epistasis for beneficial mutations sometimes implies an advantage for recombination [29].

Conventional summary statistics for epistasis have their limitations. The average curvature may obscure a diversity of interaction types, and pairwise tests fail to discover curvature at genetic distances greater than two. The most fine-scaled approach

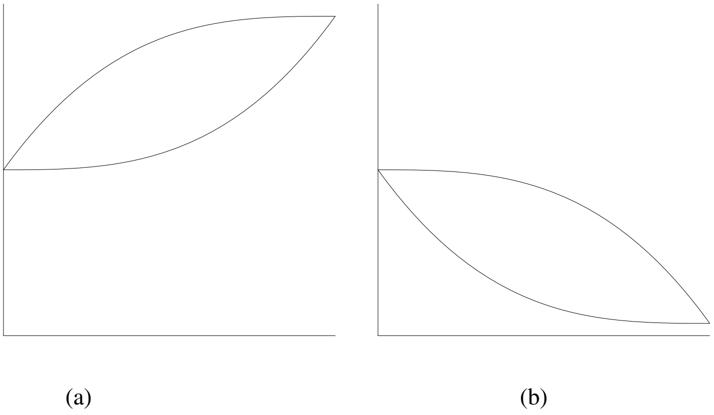


Fig. 7.6 The number of mutations increases along the horizontal axis, and the fitness increases along the vertical axis. For Figure 7.6a, the mutations are beneficial. The upper curve corresponds to antagonistic epistasis, and the lower curve to synergistic epistasis. For Figure 7.6b, the mutations are deleterious. The upper curve corresponds to synergistic epistasis and the lower curve to antagonistic epistasis.

to gene interactions is the geometric theory, introduced in [3]. The theory reveals all the gene interactions, and it depends on triangulations of polytopes. For mathematical background we refer to [9], see also [39] for the general theory about polytopes. The geometric approach has revealed previously unappreciated gene interactions for HIV, *Escherichia-coli* and in some other cases [3, 4], and the approach is relevant for recombination.

We will start with an informal introduction to the geometric theory, where the main purpose is to provide an intuitive understanding and some geometric interpretations. More formal descriptions are given in the next sections.

Roughly, a triangulation of a polygon is a subdivision of the polygon into triangles. A triangulation of the L-cube is a subdivision of the cube into simplices (triangles if $L = 2$, tetrahedra if $L = 3$, pentachora for $L = 4$, and so on). We will use some concepts which are defined in terms of populations. If one groups individuals into classes of identical genotypes, a population can be described as the frequencies of the genotypes. The *fitness of a population* is defined as the average fitness of all individuals.

First consider the case $L = 2$. Let

$$\Delta = \{(p_{00}, p_{01}, p_{10}, p_{11}) \in [0,1]^4 \; : \; p_{00} + p_{01} + p_{10} + p_{11} = 1\}$$

denote the population simplex. A population is given as a point in Δ. The *genotope* for $L = 2$ is the square with vertices $00, 01, 10, 11$. We denote this genotope $[0,1]^2$, and interpret a point $v = (v_1, v_2) \in [0,1]^2$ as the allele frequencies of the population, where v_1 denotes the frequency of 1's at the first locus, and v_2 the frequency of 1's at the second locus.

For a simple example, consider a population where half of the individuals have genotype 00, and the other half 11. Then the allele frequency vector is $(0.5,0.5)$. For a population where half of the individuals have the genotype 01 and the other half 10, the allele frequency vector is $(0.5,0.5)$ as well. However, the average fitness may differ between the two populations.

We will analyze examples in more detail.

Example 5: *Consider* $v=(0.4,0.8)\in[0,1]^2$ *and the populations* $p^1=(0.2,0.4,0,0.4)\in\Delta$ *and* $p^2=(0,0.6,0.2,0.2)\in\Delta$. *One verifies that both populations have the allele frequencies described by v; indeed, adding the contributions of 1's for* p^1 *and the first locus gives* $0+0.4=0.4$, *and for the second locus* $0.4+0.4=0.8$. *The contributions for* p^2 *gives* $0.2+0.2=0.4$ *for the first locus, and* $0.6+0.2=0.8$ *for the second.*

Let ρ denote a corresponding map from the population simplex Δ to the genotope $[0,1]^2$, where

$$\rho(p_{00},p_{01},p_{10},p_{11})=(p_{10}+p_{11},p_{01}+p_{11})\,.$$

Then ρ maps a point of the population simplex to the allele frequencies, where $p_{10}+p_{11}$ equals the frequency of 1's at the first locus, and $p_{01}+p_{11}$ equals the frequency of 1's at the second locus. Notice that $\rho(p^1)=\rho(p^2)=v$ in the previous example.

Given a fitness landscape and a vector $v\in[0,1]^2$, a *fittest population* $p\in\Delta$ has maximal fitness among populations such that the allele frequencies are described by v. Moreover, p is unique for $L=2$, except in the case when fitness is additive (see also the comment after Case 2 below). For a fittest population, one cannot increase the fitness by shuffling around alleles. The biological significance is immediate, since such allele shuffling relates to recombination.

For a geometric interpretation, the fitness landscapes w (usually) induces a triangulation of the genotope $[0,1]^2$. This triangulation is the *shape* of the fitness landscape. The critical property of the triangulation is that for $v\in[0,1]^2$, the genotypes that occur in the fittest population are the vertices of the triangle which contains v. The corresponding result holds for any L. We will first describe the triangulations, and then give a geometric interpretation of shapes. As remarked, fitness is additive exactly if

$$w_{11}-w_{10}-w_{01}+w_{00}=0\,.$$

For simplicity, we call the case where 11 has higher fitness as compared to a linear expectations positive epistasis, and similarly for negative epistasis.

Case 1: (positive epistasis) If

$$w_{11}-w_{10}-w_{01}+w_{00}>0,$$

then the triangulation induced by the fitness landscape has $00-11$ diagonal, meaning that the triangles are $\{00,01,11\}$ and $\{00,10,11\}$ (Figure 7.7).

Case 2: (negative epistasis) If

$$w_{11} - w_{10} - w_{01} + w_{00} < 0,$$

then the induced triangulation of the genotope has $10 - 01$ diagonal meaning that the triangles are $\{00, 01, 10\}$ and $\{01, 10, 11\}$ (Figure 7.7).

For positive epistasis, p^1 in Example 5 is a fittest population. Indeed, positive epistasis implies that whenever one replaces 10 and 01 genotypes by 00 and 11 genotypes, the result is increased average fitness of the population. However, for p^1 the proportions of 00 and 11 genotypes are maximal (in the sense that one cannot replace 10 and 01 genotypes by 00 and 11 genotypes), so that p^1 has maximal fitness given the allele frequency vector $v = (0.4, 0.8)$. For positive epistasis, notice that $v = (0.4, 0.8)$ is a point in the triangle $\{00, 01, 11\}$ and that the genotypes of p^1 are 00,01, 11. Moreover, 10 and 01 are not on the same triangle, which indicates that one can increase fitness by replacing them with other genotypes. These observations illustrate how shapes and fittest populations relate for $L = 2$.

For a geometric interpretation of shapes, consider the genotope $[0, 1]^2$ and the four points above the vertices of $[0, 1]^2$, such that the height coordinates correspond to fitness. The four points are vertices of a tetrahedron (Figure 7.7). The upper sides of the tetrahedron (marked with different patterns) project onto two triangles of $[0, 1]^2$. The projections describe the triangulation induced by w. The left picture corresponds to positive epistasis, and the right to negative epistasis. This construction should make sense, since the triangulation obtained as projections of the upper faces of the tetrahedron has the critical property for all fittest populations. More precisely, for any $v \in [0, 1]^2$, a fittest population consists of vertices of the triangle which contain v. The fitness landscape almost always induces a triangulation of the genotope $[0, 1]^2$ as described. Such a triangulation is a generic shape. The exceptional (non-generic) case is when fitness is additive.

In general, consider a biallelic L-loci system and the fitness landscape w. The *genotope* is the L-cube $[0, 1]^L$, where the vertices represent the genotypes. As in the two-loci case, let Δ denote the population simplex and let ρ denote the corresponding map from Δ to $[0, 1]^L$. For a fixed $v \in [0, 1]^L$, consider the linear programming problem

$$\max\{p \cdot w : \rho(p) = v\}.$$

A solution gives the maximal population fitness, i.e., the maximum of $p \cdot w$, given the allele frequency vector v (since $\rho(p) = v$). Consequently, finding the fittest population translates to solving this linear programming problem.

If we let v vary, we get the following parametric linear programming problem

$$\tilde{w}(v) = \max\{p \cdot w : \rho(p) = (v) \text{ for all } v \in [0, 1]^L\}.$$

The domains of linearity of $\tilde{w}$ do almost always constitute a triangulation of the genotope [9, Chapter 2]. The shape of the fitness landscape is the triangulation of $[0, 1]^L$ induced by the fitness landscape w. The geometric interpretation is analogous to the two-loci case, so that the triangulation is obtained as the projections of the

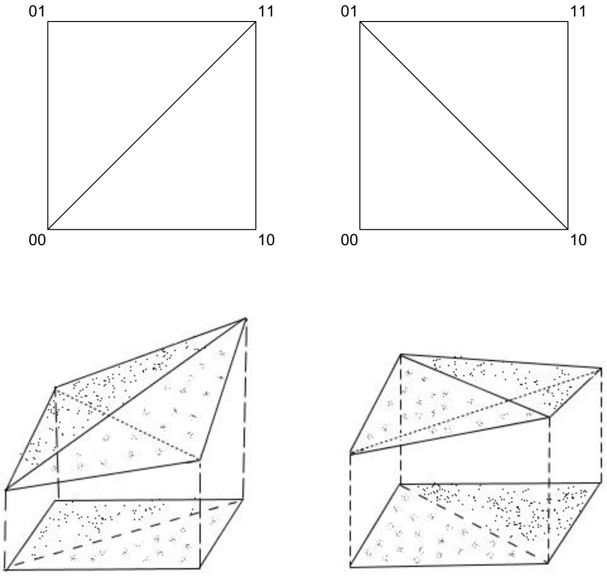


Fig. 7.7 The upper pictures show the triangulations of the genotopes (the squares) in Case 1 (positive epistasis) and 2 (negative epistasis). The lower left picture shows the tetrahedron above the genotope in Case 1, where the height coordinates correspond to the fitness of the four genotypes under consideration. The projections of the upper sides of the tetrahedron describe the triangulations. The lower right picture shows how the triangulation is induced in Case 2.

upper faces of the polytope constructed from the fitness landscape. Moreover, in the generic case, i.e., if the shape of the fitness landscape is a triangulation, the fittest population is unique for a given allele frequency vector v. More precisely, the genotypes that occur in the fittest population are the vertices of the simplex which contains v.

For the two-loci case, the geometric theory does not contribute anything new, since there exist only two triangulations corresponding to the two types of epistasis in the usual sense. However, for $L = 3$, there are 74 generic shapes corresponding to triangulations of the cube (see Section 7.9).

Not all triangulations can be obtained from a fitness landscape. A triangulation is *regular* if it is induced by some fitness landscape. Figure 7.8 shows a non-regular triangulation. This is the smallest non-regular triangulation.

In the literature, a regular triangulation is described as a triangulation which is induced by a cost vector. Then the linear programming problem concerns minimizing the cost, and the triangulations are obtained as projections of all lower faces of

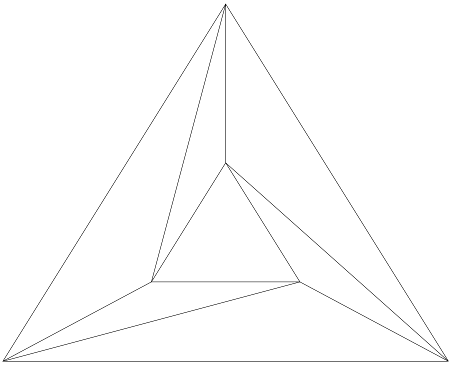


Fig. 7.8 A non-regular triangulation. This triangulation cannot be induced from a fitness landscape.

the polytope constructed from the cost vector. Since our topic is fitness landscapes, we think in terms of maximal fitness rather than minimal costs.

7.7 Shapes and Flips

For $L > 2$, there are many possible shapes. It may seem that shapes are difficult to apply in empirical biology, due to ambiguity from measurement errors. However, the geometric theory comes with a structure. Shapes may be similar or completely different, and the relation between shapes can be described in a systematic way. We start with intuitive descriptions. Briefly, a *flip*, sometimes referred to as a geometric bistellar flip, is a minimal change between triangulations. Figures 7.9, 7.10 and 7.11

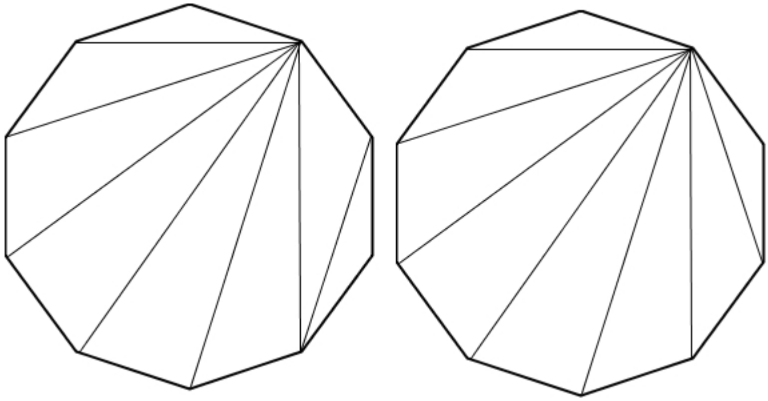


Fig. 7.9 The left triangulation can be transformed into the right triangulation by a flip

show flips. For the two-loci case, the two triangulations corresponding to positive and negative epistasis differ by a flip.

For an overview of how all triangulations of a polytope are related, one can consider the *flip graph*. The nodes of the graph are the triangulations, and edges connect triangulations which differ by a flip. Figure 7.12 shows the flip graph of a hexagon. The graph theoretical distance between triangulations can be considered a measure of how closely related the triangulations are. Some caution is necessary if one is primarily interested in regular triangulation, since a regular triangulation may be transformed into a non-regular triangulation by a flip.

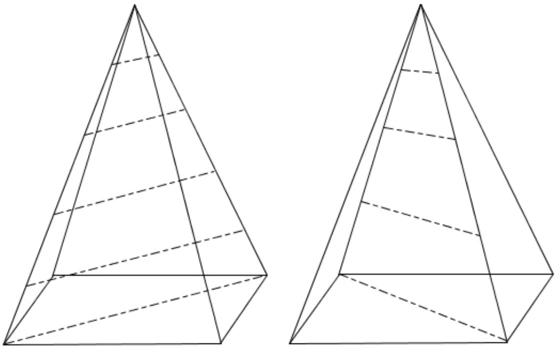


Fig. 7.10 The dashed lines indicate the triangulations. The triangulations differ by a flip.

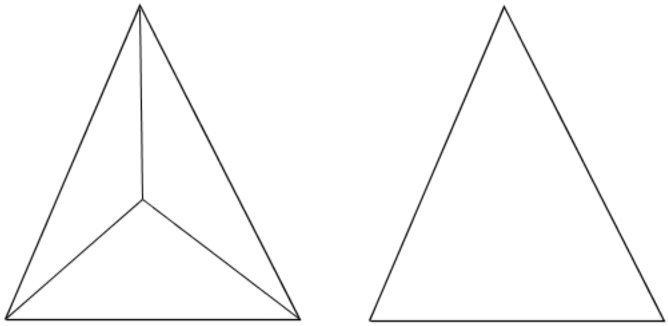


Fig. 7.11 The triangulations differ by a flip, and the number of triangles are different for the two triangulations

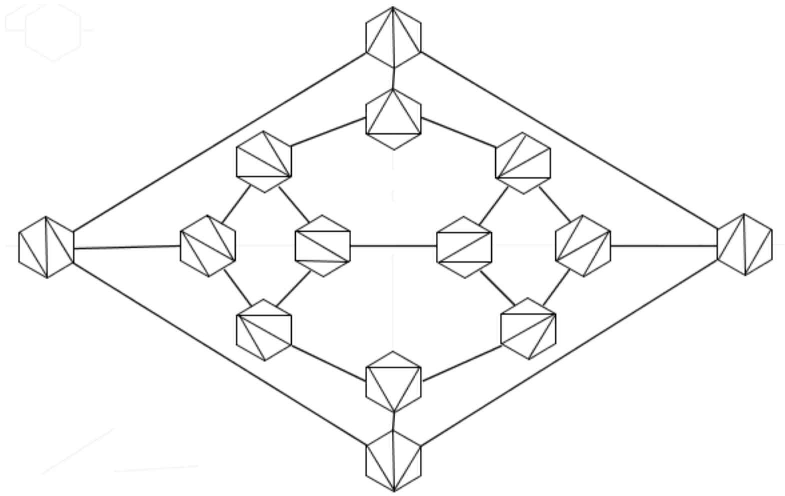


Fig. 7.12 The flip graph of a hexagon

7.8 Shapes and Polyhedral Subdivisions

Given a genotype space Σ^L, consider all possible shapes induced by fitness landscapes $w : \Sigma^L \mapsto \mathbb{R}$. We will describe how the shapes are related. Most results depend on triangulations of polytopes. In particular, we will discuss the secondary polytope [14], an important construction in discrete mathematics. The secondary polytope is useful for a global understanding of shapes. We will not provide proofs, but rather describe results and how they apply to epistasis. For a thorough treatment, including proofs, see [9] and for the biological perspective [3]. Remark 3 in the end of Section 7.9 explains how our applications relate to the general theory about polytopes. Some definitions below may seem technical, but the figures and intuitive descriptions from the previous sections should help.

Throughout the section, let $\mathbf{A} \in \mathbb{R}^d$ denote a finite point set. A *polytope* is the convex hull of a point set, and conv$(\mathbf{A})$ denotes the convex hull of $\mathbf{A}$.

Polytopes include points, line segments, triangles and tetrahedra, as well as L-cubes and polygons. We will use some concepts expressed in terms of point sets, although we have polytopes in mind. In particular, a triangulation of a polytope is a triangulation of the set of its vertices, and similarly for the other concepts.

A k-simplex is the convex hull of $k+1$ affinely independent points. In particular, points, segments, triangles and tetrahedra are simplices.

A j-*face* of a k-simplex is the convex hull of a subset of j vertices.

We will give a formal definition of triangulations and some related concepts. A *polyhedral subdivision* of a point set $\mathbf{A}$ is a collection of polytopes $\mathscr{C}$, such that

(i) If $C \in \mathscr{C}$, then each face of C belongs to $\mathscr{C}$ as well (closure property),
(ii) the union $\cup_{C \in \mathscr{C}} C = \mathrm{conv}(\mathbf{A})$ (union property),
(iii) for $C \neq C'$ where $C, C' \in \mathscr{C}$, the intersection $C \cap C'$ does not contain any interior points of C or C' (intersection property).

A *triangulation* of $\mathbf{A}$ is a polyhedral subdivision such that all polytopes are simplices. A *refinement* $\mathscr{C}'$ of a polyhedral subdivision $\mathscr{C}$ is a polyhedral subdivision $\mathscr{C}'$ where for each $C' \in \mathscr{C}'$, there exists a $C \in \mathscr{C}$, such that $C' \subset C$. A polyhedral subdivision is an *almost triangulation* if it is not a triangulation, but all its proper refinements are triangulations. Two triangulations of the same point set are connected by a *flip* if they are the only two triangulations refining an almost triangulation. All these concepts are illustrated in Figures 7.9 and 7.13. Specifically, Figure 7.13 shows a polyhedral subdivision which is also an almost triangulation. Moreover, the two possible refinements are the triangulations in Figure 7.9. As mentioned, the triangulations in Figure 7.9 differ by a flip, so that the formal definition agrees with the descriptions in the previous section.

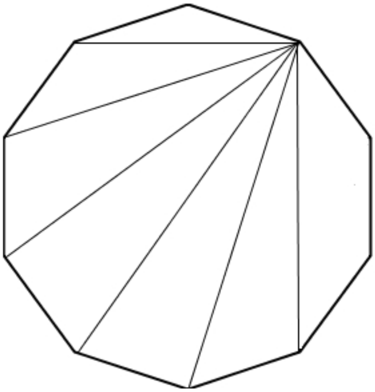


Fig. 7.13 The polyhedral subdivision is an almost triangulation. The two possible refinements result in the two triangulations from Figure 7.9.

From the previous section, a triangulation induced by the fitness landscape is the shape of the landscape. More generally, we can describe all shapes using the concepts defined in this section. Consider again the parametric linear programming problem where v varies:

$$\tilde{w}(v) = \max\{p \cdot w : \rho(p) = (v) \text{ for all } v \in [0,1]^L\}.$$

The domains of linearity of $\tilde{w}$ constitute a polyhedral subdivision [9, Chapter 2] of the genotope. The *shape* of a fitness landscape is the polyhedral division induced by the landscape. This subdivision is not always a triangulation. Recall from the two-loci case that positive epistasis corresponds to $u > 0$ and negative epistasis to $u < 0$, for

$$u = w_{00} - w_{01} - w_{10} + w_{11}.$$

However, one does not obtain a triangulation if $u = 0$. As remarked, the fitness landscape is generic if it induces a triangulation, and the corresponding shapes are the generic shapes.

In order to further describe the relations between shapes, we will consider minimal dependence sets of points, as in the following example.

Example 6: *Consider the vertices of the genotope* $[0,1]^2$. *The relation*

$$1 \cdot (0,0) - 1 \cdot (0,1) - 1 \cdot (1,0) + 1 \cdot (1,1) = 0,$$

is an affine dependence relation, since the sum of the coefficients $1 - 1 - 1 + 1$ *is zero. This set of four points is a minimal dependence set, in the sense that every proper subset of the four points is independent. The form*

$$w_{00} - w_{01} - w_{10} + w_{11},$$

corresponds to the dependence relation. Notice that the form is unique up to scaling, i.e., multiplication by a constant.

We define a *circuit* as a minimal affine dependence set. The corresponding forms are called circuits as well, and they are unique up to scaling. Flips and circuits are closely related. The circuit u corresponds to the flip between the two triangulations of the genotope in the two-loci case. More precisely, the triangulation corresponding to positive epistasis is described by $u > 0$, and the flip corresponds to replacing $u > 0$ by $u < 0$. In general, a flip corresponds to changing sign of a circuit, and some examples for $L = 3$ are given in the next section. The next concept will be used for defining the secondary polytope, and for describing flips and circuits in more detail.

For a triangulation of a polytope, we define the *GKZ vector* as follows: Each component of the GKZ vector corresponds to a vertex of the polytope. The component is the sum of the normalized volumes of all simplices containing the vertex.

In this context, it is sufficient to use that the normalized volume of the L-cube $[0,1]^L$ is $n!$, so that the genotope $[0,1]^2$ in the two loci case has area 2, and the genotope $[0,1]^3$ in the 3-loci case has volume 6. In general, for an n-dimensional lattice polytope, the normalized volume is defined as the Euclidean volume of the polytope multiplied by $n!$

Example 7: *The GKZ-vector for the triangulation of* $[0,1]^2$ *in the two-loci case associated with positive epistasis is* $(2,1,1,2)$*, using the order as given in the beginning of Section 7.6. Indeed, both triangles have area 1 (using normalized volumes). The vertices* 00 *and* 11 *belong to two different triangles each, whereas* 10 *and* 01 *are "sliced off", so that each of them belong to one triangle only. Similarly, the GKZ-vector for the triangulation associated with negative epistasis is* $(1,2,2,1)$.

The purpose with the next example is to relate circuits and GKZ vectors.

Example 8: *From the previous example, the GKZ-vector for the triangulation associated with positive epistasis is* $(2,1,1,2)$*, whereas the the GKZ vector for the triangulation associated to negative epistasis is* $(1,2,2,1)$*. We relate to the circuit* $u = w_{00} - w_{01} - w_{10} + w_{11}$ *the vector* $(1,-1,-1,1)$*. The flip between the two triangulations corresponds to u, and*

$$(2,1,1,2)-(1,2,2,1) = (1,-1,-1,1)$$

so that the GKZ vectors differ by the vector corresponding to the circuit associated with the flip.

See Remark 3 for some comments about the relation between GKZ vectors and flips, including references. In the next section, we will consider the relations between flips and GKZ vectors for $L = 3$.

For a given polytope the *secondary polytope* is defined as the the convex hull of the GKZ vectors. The geometric classification of fitness landscapes depends on the secondary polytope. For a genotope, the vertices of the secondary polytope correspond to generic shapes, and its edges to flips between the generic shapes. The higher dimensional faces of the secondary polytope correspond to non-generic shapes. Consequently, the secondary polytope represent all the shapes and their relations.

Example 9: *The secondary polytope for the two-loci case is a line segment, where the vertices corresponds to the two triangulations, and the line segment to the flat shape.*

7.9 The 74 Generic Shapes of the Cube

The relations between shapes, circuits and GKZ vector for the 3-cube, is analogous to the two-loci case, as indicated in the previous section. Recall that the square has 2 generic shapes, corresponding to $u > 0$ and $u < 0$, for

$$u = w_{00} - w_{01} - w_{10} + w_{11}.$$

The cube has 74 generic shapes, where a shape is determined by the following 20 circuits:

$$
\begin{aligned}
a &:= w_{000} - w_{010} - w_{100} + w_{110}\\
b &:= w_{001} - w_{011} - w_{101} + w_{111}\\
c &:= w_{000} - w_{001} - w_{100} + w_{101}\\
d &:= w_{010} - w_{011} - w_{110} + w_{111}\\
e &:= w_{000} - w_{001} - w_{010} + w_{011}\\
f &:= w_{100} - w_{101} - w_{110} + w_{111}\\
g &:= w_{000} - w_{011} - w_{100} + w_{111}\\
h &:= w_{001} - w_{010} - w_{101} + w_{110}\\
i &:= w_{000} - w_{010} - w_{101} + w_{111}\\
j &:= w_{001} - w_{011} - w_{100} + w_{110}\\
k &:= w_{000} - w_{001} - w_{110} + w_{111}\\
l &:= w_{010} - w_{011} - w_{100} + w_{101}\\
m &:= w_{001} + w_{010} + w_{100} - w_{111} - 2w_{000}\\
n &:= w_{011} + w_{101} + w_{110} - w_{000} - 2w_{111}\\
o &:= w_{010} + w_{100} + w_{111} - w_{001} - 2w_{110}\\
p &:= w_{000} + w_{011} + w_{101} - w_{110} - 2w_{001}\\
q &:= w_{001} + w_{100} + w_{111} - w_{010} - 2w_{101}\\
r &:= w_{000} + w_{011} + w_{110} - w_{101} - 2w_{010}\\
s &:= w_{000} + w_{101} + w_{110} - w_{011} - 2w_{100}\\
t &:= w_{001} + w_{010} + w_{111} - w_{100} - 2w_{011}
\end{aligned}
$$

We will use the letters $a - t$ in the list, as well as u, throughout the section.

In order to emphasize the connection to gene interactions, especially the algebraic aspects of epistasis, we will consider the *interaction space*. For any L, let $\mathscr{L}$ be the subspace of $\mathbb{R}^{\Sigma^L}$ consisting of additive fitness landscapes. The interaction space is the vector space dual to the quotient of $\mathbb{R}^{\Sigma^L}$ modulo $\mathscr{L}$, or

$$\left(\mathbb{R}^{\Sigma^L}/\mathscr{L}\right)^*.$$

The interaction space is spanned by the set of all circuits, where the circuits are unique up to scaling. In the two-loci case, the interaction space is spanned by u. For $L = 3$, the interaction space is spanned by the 20 circuits $a - t$.

The *circuit sign pattern* of a fitness landscape consists of the sign (positive, negative or zero) of each circuit. In the two-loci case there is only one circuit and the sign pattern is either $u > 0$, $u < 0$ or $u = 0$. A central result for the geometric classification is that the circuit sign pattern determines the shape of the fitness landscape, but in general the converse does not hold [3]. In particular, the signs of the 20 circuits $a - t$ determine the shape of the fitness landscape for $L = 3$. In total, there are 74 generic shapes. The fact that there are 20 circuits and only 74 generic shapes reflects dependence relations.

Table 7.1 lists the shapes, where the vertices of the cube are ordered as follows

$$000, 001, 010, 011, 100, 101, 110, 111.$$

Table 7.1 Shape numbers, GKZ vectors, inequalities and adjacent shapes

#	GKZ	inequalities	#	GKZ	circuits
1	15515115	tqom3,4,5,6	38	31355313	$\bar{l}\bar{g}$cd39,44,51,59
2	51151551	srpn7,8,9,10	39	31533513	l$\bar{i}$ef38,44,53,60
3	14436114	$\bar{t}$bd$\bar{e}$1,11,13,17	40	33155133	$\bar{j}\bar{g}$ab42,45,54,61
4	14614314	$\bar{q}$bf$\bar{c}$1,12,14,18	41	33511533	$\overline{hi}$ab43,46,55,62
5	16414134	$\bar{o}$df$\bar{a}$1,15,16,19	42	35133153	j$\bar{k}$ef40,45,57,63
6	34414116	$\overline{meca}$1,28,29,31	43	35311353	h$\bar{k}$cd41,46,58,64
7	41163441	$\bar{s}$ac$\bar{f}$2,20,22,26	44	51333315	gi$\bar{b}\bar{a}$38,39,65,68
8	41341641	$\bar{r}$ae$\bar{d}$2,21,23,27	45	53133135	gk$\bar{d}\bar{c}$40,42,66,69
9	43141461	$\bar{p}$ce$\bar{b}$2,24,25,30	46	53311335	ik$\bar{f}\bar{e}$41,43,67,70
10	61141443	$\bar{n}\overline{fdb}$2,32,33,34	47	13356222	$\overline{db}$f$\bar{e}$11,13,35,71
11	13446213	$\overline{bl}$d$\bar{e}$3,12,47,51	48	13623522	$\overline{fb}$d$\bar{c}$12,14,36,72
12	13624413	$\bar{b}$lf$\bar{c}$4,11,48,53	49	16323252	$\overline{fd}$b$\bar{a}$15,16,37,73
13	14346123	$\overline{dj}$b$\bar{e}$3,15,47,54	50	22265331	$\overline{ca}$e$\bar{f}$20,22,35,71
14	14613423	$\overline{fh}$b$\bar{c}$4,16,48,55	51	22356213	e$\bar{b}\bar{c}$d11,17,38,71
15	16324143	$\bar{d}$jf$\bar{a}$5,13,49,57	52	22532631	$\overline{ea}$c$\bar{d}$21,23,36,72
16	16413243	$\bar{f}$hd$\bar{a}$5,14,49,58	53	22623513	c$\bar{b}\bar{e}$f12,18,39,72
17	23346114	e$\bar{g}$bd3,28,51,54	54	23256123	e$\bar{d}\bar{a}$b13,17,40,71
18	23613414	c$\bar{i}$bf4,29,53,55	55	23612523	c$\bar{f}\bar{a}$b14,18,41,72
19	26313144	a$\bar{k}$df5,31,57,58	56	25232361	$\overline{ec}$a$\bar{b}$24,25,37,73
20	31264431	$\bar{a}\bar{l}$c$\bar{f}$7,21,50,59	57	26223153	a$\bar{d}\bar{e}$f15,19,43,73
21	31442631	$\bar{a}$le$\bar{d}$8,20,52,60	58	26312253	a$\bar{f}\bar{c}$d16,19,43,73
22	32164341	$\bar{c}\bar{j}$a$\bar{f}$7,24,50,61	59	31265322	f$\bar{a}\bar{d}$c20,26,38,71
23	32431641	$\bar{e}\bar{h}$a$\bar{d}$8,25,52,62	60	31532622	d$\bar{a}\bar{f}$e21,27,39,72
24	34142361	$\bar{c}$je$\bar{b}$9,22,56,63	61	32165232	f$\bar{c}\bar{b}$a22,26,40,71
25	34231461	$\bar{e}$hc$\bar{b}$9,23,56,64	62	32521632	d$\bar{e}\bar{b}$a23,27,41,72
26	41164332	f$\bar{g}$ac7,32,59,61	63	35132262	b$\bar{c}\bar{f}$e24,30,42,73
27	41431632	d$\bar{i}$ae8,33,60,62	64	35221362	b$\bar{e}\bar{d}$c25,30,32,73
28	43324116	eg$\overline{ca}$ 6,17,65,66	65	52323216	ceb$\bar{a}$28,29,44,74
29	43413216	ci$\overline{ea}$6,18,65,67	66	53223126	aed$\bar{c}$28,31,45,74
30	44131362	b$\bar{k}$ce9,34,63,64	67	53312226	acf$\bar{e}$29,31,46,74
31	44313126	ak$\overline{ec}$6,19,66,67	68	61232325	dfa$\bar{b}$32,33,44,74
32	61142334	fg$\overline{db}$10,26,68,69	69	62132235	bfc$\bar{d}$32,34,45,74
33	61231434	di$\overline{fb}$10,27,68,70	70	62221335	bde$\bar{f}$33,34,46,74
34	62131344	bk$\overline{fd}$10,30,69,70	71	22266222	ef$\overline{db}\overline{ca}$47,50,51,54,59,61
35	13355331	$\overline{ljf}\bar{e}$36,37,47,50	72	22622622	cd$\overline{fb}\overline{ea}$48,52,53,55,60,62
36	13533531	l$\overline{hd}\bar{c}$35,37,48,52	73	26222262	ab$\overline{fd}\overline{ec}$49,56,57,58,63,64
37	15333351	jh$\bar{b}\bar{a}$35,36,49,56	74	62222226	acebdf65,66,67,68,69,70

For each shape, the table gives the GKZ vector, the defining inequalities, and the adjacent shapes. In particular, for Shape 74 the notation

$$\text{acebdf}65,66,67,68,69,70,$$

in the second column means that Shape 74 is defined by

$$a,c,e,b,d,f > 0.$$

and that the adjacent shapes are 65,66,67,68,69,70. Similarly, for Shape 65 the notation

$$\text{ceb}\bar{\text{a}}28,29,44,74,$$

means that the shape is defined by

$$c > 0, e > 0, b > 0, a < 0,$$

where $\bar{\text{a}}$ indicates that $a < 0$, and the adjacent shapes are 28, 29, 44, 74.

Each inequality of Shape 74 can be described in terms of epistasis (in the usual sense), since each inequality keeps one locus fixed. In contrast, the inequalities of Shape 1 considers three-way interactions. The fact that $m > 0$, where

$$m = w_{001} + w_{010} + w_{100} - w_{111} - 2w_{000}$$

shows that the genotype 111 has lower fitness as compared to a linear expectation from the values

$$w_{001}, w_{010}, w_{100}, w_{000}.$$

This observation shows already that the geometric theory is more fine-scaled as compared to conventional approaches.

The 74 shapes fall into six categories, called the *interaction types*. Specifically, the types consist of Shape 1-2, 3-10, 11-34, 35-46, 47-70 and 71-74. For pictures of the six interaction types, see [9, Chapter 1]. Shapes of the same type differ only in the labeling of the vertices. In particular, the shapes of the same interaction type in the table have GKZ vectors that differ only by a permutation of the components.

As in the two-loci case, the circuits correspond to flips. The letters representing circuits are ordered according to the shapes resulting from the corresponding flips. Consider again Shape 74. In addition to the information described, the notation

$$acebdf65,66,67,68,69,70.$$

indicates how flips and shapes relate in a precise way. The flip corresponding to a results in Shape 65 (the first letter is paired with the first number), the flip corresponding to c results in Shape 66, and so forth.

For an explicit description, consider Shape 74 and the flip corresponding to a. Since Shape 74 is defined by

$$a > 0, c > 0, e > 0, b > 0, d > 0, f > 0,$$

the result of the flip is the shape defined by

$$a < 0, c > 0, e > 0, b > 0, d > 0, f > 0,$$

which reduces to

$$a < 0, c > 0, e > 0, b > 0,$$

since the four inequalities imply that $d > 0$ and $f > 0$. Shape 65 is described by exactly these four inequalities in the table.

The table lists GKZ vectors as well. Flips and GKZ-vectors are related, as in the two-loci case. For instance, the GKZ vector is 62222226 for Shape 74, and 52323216 for Shape 65. The circuit a corresponds to the vector

$$(1,0,-1,0,-1,0,1,0),$$

and

$$(6,2,2,2,2,2,2,6) - (5,2,3,2,3,2,1,6) = (1,0,-1,0,-1,0,1,0).$$

(see Remark 3). For a systematic interpretation of the 20 circuits $a - t$ listed, one may consider the Fourier transform for the group $(\mathbb{Z}_2)^n$ [3]. Geometric interpretations of the circuits are given in the same paper.

Remark 3: *We have indicated results about polytopes and triangulations of relevance in evolutionary biology (see also the next section about shapes and empirical data). We refer to De Loera et al. [9] for general background about triangulations of polytopes, where the Chapters 4 and 5 are especially important. "Additive fitness landscapes", as defined here, translates to "linear evaluations" in the general theory, and "interaction spaces" to "linear dependences". The fact that the interaction space is spanned by the circuits is an aspect of Gale duality [9, Chapter 4]. The relation between GKZ vectors and flips were illustrated above for the Shapes 65 and 74, and in Example 8 in the previous section. In terms of the general theory, the interaction space equals the linear space parallel to the secondary polytope [9, Chapter 5], and a detailed description of the relation between GKZ vectors and flips is given in the same chapter.*

7.10 Shapes and Empirical Data

The described relations between circuits, flips, GKZ-vectors and the secondary polytopes hold under very general assumptions. We restricted our discussion to biallelic L-loci populations in order to keep the presentation simple. For the geometric theory of gene interactions, the genotope is defined for any set of genotypes found in a population and the shape is defined accordingly [3]. In fact, the authors stress that the genotope is never an L-cube for binary data and many loci (≥ 20), which is important for complexity reasons. Empirical examples of general type [3, 4] can be analyzed similarly to the restricted case we considered here. Empirical data can be understood as to belong to an empirical fitness landscape. For a shape analysis of empirical data, one needs several fitness measurements of each genotype due to statistical variation. One may not find a unique shape, but rather a set of similar shapes which are compatible with the data.

A shape analysis of HIV fitness data is given in [3]. The biallelic three-loci system considered there is associated with HIV drug resistance. From bootstrapping, the three dominant shapes are 2, 7 and 10. Notice that these shapes are adjacent, and have similar GKZ vectors. Moreover, the five most dominant shapes 2, 7, 10, 26 and 32 appear as a face of the secondary polytope of the cube, and have similar GKZ vectors as well.

Software for analyzing shapes is available, for example Polymake[1].

7.11 Shapes and Fitness Graphs

A fitness graph is determined by fitness ranks only. The information from shapes is incomparably more fine-scaled. It is of interest to compare the two perspectives.

For the two-loci case, assume that the 11 genotype has maximal fitness. Then positive epistasis is compatible with three fitness graphs (no arrows down, exactly one arrow down, or two arrows down). On the other hand, consider the fitness graphs with all arrows up. Such a graph is compatible with positive, negative or no epistasis. This example shows that fitness graphs provide information that cannot be obtained from the geometric classification, and vice versa, and the same observation holds for any L. Since L=2 is rather special, we will demonstrate that fitness graphs and shapes provide complementary information also for $L = 3$, where the examples are from [6]. For more comparisons of fitness graphs and shapes, also in the context of empirical data, see the same paper.

The following all arrows up landscapes is of Shape 74,

$$w_{000} = 1, w_{100} = w_{010} = w_{001} = 2, w_{110} = w_{101} = w_{011} = 4, w_{111} = 7.$$

The landscape

$$w_{000} = 2, w_{100} = w_{010} = w_{001} = 1, w_{110} = w_{101} = w_{011} = 4, w_{111} = 8,$$

is of shape 74 as well. The corresponding fitness graph has exactly 3 arrows down, and both 000 and 111 are at peaks. It is easily seen that there exist fitness landscapes of shape 74 with other fitness graphs, in addition to the two mentioned.

For each interaction type for $L = 3$, Table 2 gives the shape number and an example of an all arrows up landscape of this shape.

As we have seen, fitness graphs and shapes provide complementary information. There is usually an overlap in the information as well. For instance, if all arrows point away from a particular genotype, and if the genotype is ”sliced off” for the shape, then one has two indications that the genotype has relatively low fitness.

Fitness graphs provide information about the adaptive potential if one restricts to single mutations. The graphs reveal coarse properties, such as sign epistasis,

[1] `http://www.polymake.org/doku.php`

Table 7.2 Interaction types, shape numbers and fitness landscapes

	w_{000}	w_{100}	w_{010}	w_{001}	w_{110}	w_{101}	w_{011}	w_{111}
Type 1, no 2:	1	2	2	2	4	4	4	5
Type 2, no 10:	1	2	2	2	6	6	6	9
Type 3, no 34:	1	2	2	2	10	6	5	12
Type 4, no 46:	1	2	5	5	8	8	8	13
Type 5, no 70:	1	2	5	5	9	9	10	15
Type 6, no 74:	1	2	2	2	4	4	4	7

mutational trajectories, and the number of peaks. It is clear that a complete analysis of recombination requires more fine-scaled information as compared to what fitness graphs provide. The geometric theory, on the other hand, reveals all gene interactions. Finding the shape of a landscape is equivalent to finding all the fittest populations, which explains why shapes are relevant for recombination.

From a more philosophical perspective, the interest in fitness graphs and shapes depends on the belief that average effects of mutations are insufficient for analyzing evolutionary dynamics.

7.12 Discussion

We have considered fitness graph and the geometric theory of gene interactions. Fitness graphs and shapes provide complementary information, and there tend to be some overlap in the observations. Fitness graphs are useful for analyzing peaks and other coarse properties of fitness landscapes. The graphs have been used in empirical work, and for relating global and local properties of fitness landscapes.

The geometric theory extends the usual concept epistasis to any number of loci, where shapes, as defined in the geometric theory, correspond to positive and negative epistasis for two mutations. The geometric classification is meaningful because it comes with a structure. A particular shape can be put in a context, and compared to other shapes. In summary, for biallelic populations where all 2^L genotypes are represented, the genotope is an L-cube. The shape of a fitness landscape is a polyhedral subdivision of the genotope induced by the landscape. The generic shapes are the triangulations of the genotope. The relation between all the generic shapes can be described in terms of flips, or minimal changes between shapes. The flip graph provides an overview of the generic shapes and how they can be transformed into each other by flips. The secondary polytope encodes all shapes and their relations, where the generic shapes correspond to vertices, and the non-generic shapes to the higher dimensional faces. For an algebraic perspective, the interaction space is spanned by a set of linear forms, or circuits. The shapes are determined by the sign pattern of the circuits, and changing sign of a circuit corresponds to a flip.

The geometric theory has provided new insights about gene interactions in empirical studies. The theory may be considered a fundamentally new approach to recombination. There is clearly a potential for new applications of shapes to evolutionary biology, even if the geometric theory is complete.

The approaches discussed here are similar in one respect. They make no assumptions, or minimal assumptions, about the underlying fitness landscapes. The accuracy of an analysis of empirical data using fitness graphs or the geometric theory does not depend on any *a priori* assumptions about the fitness landscape. Fitness graphs and shapes are well suited for empirical studies for that reason.

Acknowledgements. I am grateful to Francisco Santos for valuable comments.

References

[1] Aita, T., Iwakura, M., Husimi, Y.: A cross-section of the fitness landscape of dihydrofolate reductase. Protein Eng. 14, 633–638 (2001)

[2] Beerenwinkel, N., Eriksson, N., Sturmfels, B.: Conjunctive Bayesian networks. Bernoulli 13, 893–909 (2007)

[3] Beerenwinkel, N., Pachter, L., Sturmfels, B.: Epistasis and shapes of fitness landscapes. Statistica Sinica 17, 1317–1342 (2007)

[4] Beerenwinkel, N., Pachter, L., Sturmfels, B., Elena, S.F., Lenski, R.E.: Analysis of epistatic interactions and fitness landscapes using a new geometric approach. BMC Evolutionary Biology 7, 60–61 (2007)

[5] Carnerio, M., Hartl, D.L.: Colloquium papers: Adaptive landscapes and protein evolution. Proc. Natl. Acad. Sci USA 107(Suppl. 1), 1747–1751 (2010)

[6] Crona, K., Greene, D., Barlow, M.: The peaks and geometry of fitness landscapes. J. Theor. Biol. 317, 1–13 (2013)

[7] Crona, K., Patterson, D., Stack, K., Greene, D., Goulart, C.P., Mentar, M., Jacobs, S.J., Kallmann, M., Barlow, M.: Antibiotic resistance landscapes: a quantification of theory-data incompatibility for fitness landscapes (2013), http://arxiv.org/abs/1303.3842

[8] Crow, J.F.: H. J. Muller and the "competition hoax". Genetics 173, 511–514 (2006)

[9] De Loera, J.A., Rambau, J., Santos, F.: Triangulations: Applications, Structures and Algorithms. Algorithms and Computation in Mathematics, vol. 25. Springer, Heidelberg (2010)

[10] De Visser, J.A.G.M., Park, S.C., Krug, J.: Exploring the effect of sex on empirical fitness landscapes. The American Naturalist 174(suppl. 1), S15–S30 (2009)

[11] Desper, R., Jiang, F., Kallioniemi, O.P., Moch, H., Papadimitriou, C.H., Schäffer, A.A.: Inferring tree models for oncogenesis from comparative genome hybridization data. Comput. Biol. 6, 37–51 (1999)

[12] Flyvbjerg, H., Lautrup, B.: Evolution in a rugged fitness landscape. Phys. Rev. A 46, 6714–6723 (1992)

[13] Franke, J., Klözer, A., de Visser, J.A.G.M., Krug, J.: Evolutionary accessibility of mutational pathways. PLoS Comput. Biol. 7(8), e1002134 (2011), doi:10.1371/journal.pcbi.1002134

[14] Gelfand, I.M., Kapranov, M., Zelevinsky, A.: Discriminants, Resultants and Multidimensional Determinants. Birkhäuser, Basel (1994)

[15] Gillespie, J.H.: A simple stochastic gene substitution model. Theor. Pop. Biol. 23, 202–215 (1983)

[16] Gillespie, J.H.: The molecular clock be an episodic clock. Proc. Natl. Acad. Sci. USA 81, 8009–8013 (1984)

[17] Goulart, C.P., Mentar, M., Crona, K., Jacobs, S.J., Kallmann, M., Hall, B.G., Greene, D., Barlow, M.: Designing antibiotic cycling strategies by determining and understanding local adaptive landscapes. PLoS One 8(2), e56040 (2013), doi:10.1371/journal.pone.0056040

[18] Haldane, J.B.S.: A mathematical theory of natural and artificial selection. Part VIII. Metastable populations. Proc. Cambridge Philos. Soc. 27, 137–142 (1931)

[19] Kauffman, S.A., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. J. Theor. Biol. 128, 11–45 (1987)

[20] Kauffman, S.A., Weinberger, E.D.: The NK model of rugged fitness landscape and its application to maturation of the immune response. J. Theor. Biol. 141, 211–245 (1989)

[21] Kingman, J.F.C.: A simple model for the balance between selection and mutation. J. Appl. Prob. 15, 1–12 (1978)

[22] Kryazhimskiy, S., Draghi, J.A., Plotkin, J.B.: In evolution, the sum is less than its part. Science 332, 1160–1161 (2011)

[23] Lenski, R.E., Ofria, C., Pennock, R.T., Adami, C.: The evolutionary origin of complex features. Nature 423, 139–144 (2003)

[24] Macken, C.A., Perelson, A.S.: Protein evolution on partially correlated landscapes. Proc. Natl. Acad. Sci. USA 92, 9657–9661 (1995)

[25] Mani, R., Onge, R.P.S., Hartman, J.L., Giaever, G., Roth, F.P.: Defining genetic interaction. Proc. Natl. Acad. Sci. USA 105, 3461–3466 (2008)

[26] Martin, G., Otto, S.P., Lenormand, T.: Selection for recombination in structured populations. Genetics 172, 593–609 (2006)

[27] Maynard Smith, J.: Natural selection and the concept of protein space. Nature 225, 563–564 (1970)

[28] Orr, H.A.: The population genetics of adaptation on correlated fitness landscapes: the block model. Evolution 60, 1113–1124 (2006)

[29] Otto, S.P., Lenormand, T.: Resolving the paradox of sex and recombination. Nature Reviews Genetics 3, 252–261 (2002)

[30] Park, S.C., Krug, J.: Evolution in random fitness landscapes: The infinite sites model. J. Stat. Mech., P4014 (2008)

[31] Poelwijk, F.J., Kiviet, D.J., Weinreich, D.M., Tans, S.J.: Empirical fitness landscapes reveal accessible evolutionary paths. Nature 445, 383–386 (2007)

[32] Poelwijk, F.J., Sorin, T.N., Kiviet, D.J., Tans, S.J.: Reciprocal sign epistasis is a necessary condition for multi-peaked fitness landscapes. J. Theor. Biol. 272, 141–144 (2011)

[33] Rokyta, D.R., Beisel, C.J., Joyce, P.: Properties of adaptive walks on uncorrelated landscapes under strong selection and weak mutation. J. Theor. Biol. 243, 114–120 (2006)

[34] Segal, M.R., Barbour, J.D., Grant, R.M.: Relating HIV-1 sequence variation to replication capacity via trees and forests. Stat. Appl. Genet. Mol. Biol. 3, 2-1-20 (2004)

[35] Szendro, I.G., Schenk, M.F., Franke, J., Krug, J., de Visser, J.A.G.M.: Quantitative analyses of empirical fitness landscapes. J. Stat. Mech., P1005 (2013)

[36] Weinreich, D.M., Watson, R.A., Chao, L.: Sign epistasis and genetic constraint on evolutionary trajectories. Evolution 59, 1165–1174 (2005)

[37] Weinreich, D.M., Delaney, N.F., Depristo, M.A., Hartl, D.L.: Darwinian evolution can follow only very few mutational paths to fitter proteins. Science 312, 111–114 (2006)

[38] Wright, S.: Evolution in Mendelian populations. Genetics 16, 97–159 (1931)

[39] Ziegler, G.: Lectures on Polytopes. Graduate Texts in Mathematics, vol. 152. Springer, Berlin (1995)

Chapter 8
Estimating the Degree of Neutrality and Ruggedness of Fitness Landscapes

Yoshiaki Katada

Abstract. In recent years, not only ruggedness but also neutrality has been recognized as an important feature of a fitness landscape for genetic search. As it has been reported that the evolutionary dynamics on a fitness landscape with neutrality is clearly different from the canonical explanations, ruggedness alone might be inadequate describing it. Another measure, i.e., neutrality is required. This study discusses the use of *standard genetic distance*, which originates from population genetics, for measuring neutrality of fitness landscapes. Firstly, several computer simulations are conducted with a test landscape with neutrality as well as ruggedness in order to clarify the characteristics of standard genetic distance on it. Second, computer simulations are conducted with an evolutionary robotics problem which would be expected to include neutrality in its landscape in order to investigate the validity of the proposed approach on a real-world problem. The results suggest that genetic distance is a reliable method for estimating the degree of neutrality of real-world problems.

8.1 Concept of Fitness Landscapes

We may consider evolution to be similar to mountain climbing. In this sense, a fitness landscape is what mountain geography or shape represents. The concept of fitness landscapes was first introduced by a population geneticist, S. Wright [43]. A fitness landscape is a tuple (S, f) where S is the set of admissible solutions, $f : S \rightarrow R$ is the fitness function and R is the set of fitness values. In a multidimensional space, we can recognize neither the shape of a fitness landscape nor that of a fitness function. In three dimensional space, we can visually recognize a fitness landscape as a mountain with a horizontal plane representing two genes on locus and the genes'

Yoshiaki Katada
Department of Electrical and Electronic Engineering,
Setsunan University, 17-8 Ikeda-nakamachi, Neyagawa, Osaka 572-8508, Japan
e-mail: katada@ele.setsunan.ac.jp

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_8, © Springer-Verlag Berlin Heidelberg 2014

fitness normal to the plane. One point on a fitness landscape represents a particular genotype and its fitness value. The application of a genetic operator means a movement of the point on the fitness landscape and the application of selection means the fixation or elimination of those points. Thus, an evolutionary process is considered to be the movements of the points (a population) on a fitness landscape. In other words, a sequence of pairs consisting of a genotype and its fitness value obtained during evolutionary process corresponds to footprints or a trail of evolution on the landscape, and the application of genetic operators corresponds to one step on the landscape (Figure 8.1).

Based on the above definition, however, a fitness landscape is identified with a fitness function. More generally, a fitness landscape is a triplet $(S, N(s), f)$ where $N(\cdot)$ is neighborhood function and $N(s)$ is the neighborhood of $s \in S$ [6, 39, 40]. The reason why $N(s)$ is included in the tuple might be explained as follows: Understanding the whole fitness landscape cannot be achieved without the enumeration of admissible solutions in the search space. Thus, some sampling methods are required to outline the fitness landscape. Random sampling has been rarely used. *Random walk* or *hill climbing* algorithms have been generally used as such sampling methods to obtain sequential solutions (neighboring solutions) according to some transition rules[1]. Therefore, the fitness landscape (outlined by using those sampled solutions) depends on the neighborhood function. This is analogous to the metaphor of Simon's ant on the beach [30]: From the perspective of a small animal, the landscape looks completely different. The step size and the way to leave its footprints are different from those of a big animal (Figure 8.2).

The fitness landscape directly connects to the difficulty of a problem to be solved. Thus, the objectives to investigate the geography of a fitness landscape are as follows:

(1) to quantify the difficulty of a problem to be solved by an evolutionary algorithm in advance, and then to select an useful evolutionary algorithm, including parameter tuning for it, and
(2) to analyze the performance of the search algorithm after the run.

As I have mentioned above, understanding the whole fitness landscape cannot be achieved without the enumeration of admissible solutions in the search space.

Fig. 8.1 Large footprints on a fitness landscape

[1] If an evolutionary algorithm is used as such a sampling method, genetic operators, ϕ, correspond to the transition rule. Thus, ϕ specifies the neighborhood $N_\phi(\cdot)$.

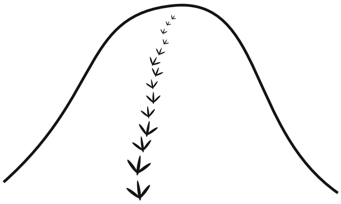


Fig. 8.2 Small footprints on a fitness landscape

Additionally, even for the same fitness function, each evolutionary algorithm leaves different footprints on the fitness landscape. Therefore, the difficulty of a problem has been discussed based on the geography of a fitness landscape.

8.2 Featuring Fitness Landscapes

In early works of the theoretical Genetic Algorithms (GA) community, problem difficulties for a GA have been discussed using the abstract terms concerning the geography of a fitness landscape: *isolation*, *deception* and *multimodality*. However, there are still contentious issues with how these affect the performance of the GA to solve optimization problems. Several counterexamples on the latter two factors were found, showing that they are neither necessary nor sufficient to make a problem difficult [7, 8, 9].

Another attempt to characterize difficulty has been done by measuring the feature of a fitness landscape, i.e. *ruggedness*. In the GA community, the majority of fitness landscape descriptions have been based on ruggedness [8, 6, 18, 21, 36, 42]. Generally, the degree of ruggedness can be estimated from the average of fitness correlations between parents and offspring or the fitness distance autocorrelation function obtained by using a random walk.

In recent years, the existence of problem domains has been reported where evolutionary dynamics is clearly different from the canonical explanations based on the schema theory and the building block hypothesis [22, 26]. Those kinds of problems, such as the evolution of neural network controllers in robotics [5, 33, 34, 35] and on-chip electronic circuit evolution [37, 38, 39], mainly show equilibrium periods, i.e. *neutral evolution* in their evolutionary dynamics. This characteristic is caused by highly redundant mappings from genotype to phenotype or phenotype to fitness [12]. With these kinds of problems, redundancy is inevitable although it is customary among GA practitioners to deliberately avoid redundancy in the genetic coding of artificial evolution problems. Therefore, ruggedness alone is not enough to measure the search difficulty if equilibrium periods seem dominant in the process of evolution. Neutrality is required.

8.3 Measuring Neutrality of Fitness Landscapes

To the best of our knowledge, statistical measurements with respect to neutrality in evolutionary computation are found only in the references [32, 39, 41]. Vassilev *et al.* [39] studied the structure of on-chip electronic circuit evolution landscapes. They proposed *information analysis* of fitness landscapes, which is defined over a time series obtained by a walk on a landscape. Their entropic measure of the time series makes it possible to confirm the existence of neutrality of a landscape and which feature, neutrality or ruggedness, is dominant in the landscape.

Smith *et al.* [32] proposed a method for measuring neutrality of a landscape as one of the *fitness evolvability portraits*. This is defined as the probability that an offspring fitness is equal to the parent fitness. In their measurement, a certain distinctive difference, ε, between two fitness values considered to be neutral must be set by GA practitioners; that is, $|f_x - f_y| \leq \varepsilon$, where f_x is a parent fitness and f_y is the offspring fitness. However, Vassilev *et al.* [39] have reported that the measure of Smith *et al.* is very sensitive to ε. Especially in the case that the fitness is evaluated as a real value or in a noisy environment, a great influence of the value on the measure of neutrality would be predicted. For these problems, Smith proposed the use of the *neutral fitness band* [34] or the significance level for the Student t-test [31] as ε. However, no significant difference between two fitness landscapes was detected although introducing such statistical neutrality [31]. This implies a difficulty in using fitness data for measuring neutrality.

Verel *et al.* [41] defined a neutral neighbor of s as $N_n(s) = \{x \in N(s) | f(x) = f(s)\}$ (according to the notation in Section 8.1) and calculated the neutral degree of a solution as the number of its neutral neighbors. They claimed that a fitness landscape is neutral if there are many solutions with high neutral degree. However, they may have the same problem as that of Smith *et al*'s method as mentioned above.

8.4 Objectives in This Work

Population geneticists have been trying to explain the change of gene frequency in a population. That is, they have used genotype data for their explanations. Recently, we have been motivated by this to investigate the characteristics of Nei's standard genetic distance [23], which is one of statistical methods for estimating gene differences between populations in population genetics, in artificial evolution [17]. The results show some consistencies with the neutral theory [19] and the nearly neutral theory [28, 29] in population genetics.

The remainder of this chapter introduces how genetic distance applies to estimate the degree of neutrality of fitness landscapes. In this study, we use a binary representation for the genotype. Thus, the greatest advantage of our approach is that the difficulty using fitness data for measuring neutrality, which was mentioned above, need not to be taken into account because our approach uses genotype data for measuring neutrality. The chapter is organized as follows. The next section describes Nei's standard genetic distance. Section 8.6 applies the genetic distance to tunable neutral landscapes and shows the characteristics of the genetic distance. Section 8.7 and 8.8 shows some guidelines for estimating the degree of neutrality of a fitness landscape on a real-world problem as well as investigates the validity of the proposed approach on a robot control problem. Conclusions are given in the last section.

8.5 Tools for Measurements of Features of Fitness Landscapes and Their Characteristics

In this section, the tools for measuring ruggedness and neutrality are described. Ruggedness is estimated based on Smith *et al*'s measurement [32] and neutrality is based on our measurement, standard genetic distance [11, 23].

8.5.1 Measure of Ruggedness

In real-world problems, ruggedness of a fitness landscape is predicted by fitness correlation [32, 42]. In this work, the measurement of Smith *et al.* [32] was employed for the measure of ruggedness because fitness correlation can be expressed as a scalar value using their measurement. The reason why a scalar value is needed is described in Section 8.6.4 and Section 8.7.

Smith *et al.* [32] reported that fitness correlation is expressed by the gradient of the expected offspring fitness versus parent fitness graph. The expected offspring fitness for their parent fitness k is given by

$$\bar{f}(k) = \frac{\sum_{g \in G^k} f(g)}{|G^k|} \tag{8.1}$$

where G^k is the set of offspring genotype from parents with the fitness k over generations, g is an offspring genotype and $f(.)$ is the fitness function. It has also been reported in [11, 32] that the gradient of the expected offspring fitness versus parent fitness, r, increases with the decrease of ruggedness; that is, $r \simeq 1.0$ without any epistatic linkages between genes while $r \simeq 0.0$ with maximum epistatic linkages and that this gradient is independent of neutrality. These characteristics are illustrated in Section 8.6.

8.5.2 Standard Genetic Distance

Genetic distance is a term from population genetics used for estimating gene differences per locus between populations. Although there are several definitions for genetic distance, Nei's standard genetic distance [23] is adopted in our method [11].

Nei's standard genetic distance is defined for a binary coded GA as follows: Consider two populations, X and Y. Let $x_{il} = n_{il}/M$ and $y_{il} = n_{il}/M$ be the frequencies of the l-th alleles in X and Y, respectively ($i = 1, \cdots, N$; N is the length of the genotype, $l \in \{1,2\}$ in a binary coded GA, n_{il} is the number of the l-th allele, M is the population size). The probability of identity of two randomly chosen genes is $j_{xi} = x_{i1}^2 + x_{i2}^2$ in the population X, while it is $j_{yi} = y_{i1}^2 + y_{i2}^2$ in the population Y. The probability of identity of a gene from X and a gene from Y is $j_{xyi} = x_{i1}y_{i1} + x_{i2}y_{i2}$. The normalized identity of genes between X and Y with respect to a locus is defined as

$$I_i = \frac{j_{xyi}}{\sqrt{j_{xi}}\sqrt{j_{yi}}}, \tag{8.2}$$

where $I_i = 1.0$ if the two populations have the same alleles in identical frequencies, and $I_i = 0.0$ if they have no common alleles. The normalized identity of genes between X and Y with respect to the average in all loci is defined as

$$I = \frac{J_{XY}}{\sqrt{J_X}\sqrt{J_Y}}, \tag{8.3}$$

where $J_X = \sum_{i=1}^{N} j_{xi}/N$, $J_Y = \sum_{i=1}^{N} j_{yi}/N$ and $J_{XY} = \sum_{i=1}^{N} j_{xyi}/N$. The genetic distance between X and Y is defined as

$$D = -\log_e I. \tag{8.4}$$

The above definition cannot be applied to the GAs directly, because it is assumed in population genetics that a new allele always appears on a locus when a mutation occurs, while "back mutations" [28] frequently occur in GAs due to the binary coding scheme. Therefore, the accumulated genetic distance over generations is calculated for GAs as

$$D(T) = \sum_{t=1}^{T-1} D_{t,t+1} \tag{8.5}$$

where T is the number of the last generation and $D_{t,t+1}$ is the genetic distance between the population in the t-th and the $(t+1)$-th generation. In the reminder of this chapter, for simplicity, we call $D(T)$ in Equation (8.5) the genetic distance.

8.6 Characteristics of the Tools in Tunably Neutral NK Landscapes

In this section, the characteristics of the tools described in the previous section are illustrated in tunably neutral landscapes. Based on this illustration, how to use these tools for estimating the degree of neutrality as well as ruggedness reveals.

8.6.1 Tunably Neutral NK Landscapes

Tunably neutral NK landscapes were employed as test functions in our computer simulations. These are the extended versions of Kauffman's NK fitness landscape model [18]. One is referred to as the NKq fitness landscapes proposed by Newman and Engelhardt [24] and the other is referred to as the NKq-p fitness landscapes, which we extended to increase the degree of neutrality in the NKq.

The NKq landscape has three parameters: N, the length of the genotype; $K(<N)$, the number of epistatic linkages between genes [18]; and $F(\geq 2)$, the integer that tunes the neutrality. The fitness value is calculated as follows: The fitness contribution of the i-th locus, f_i, is an integer generated randomly in the range $0 \leq f_i < F$, $i = 1, \cdots, N$. To calculate the fitness, f, of a genotype, the fitness contribution of each locus is averaged, and then divided by $F-1$, normalizing f to the range 0.0 to 1.0. More formally,

$$f = \frac{1}{N(F-1)} \sum_{i=1}^{N} f_i. \tag{8.6}$$

The neutrality of the landscape can be tuned by changing the value of F. The neutrality of the landscape is maximized when $F = 2$, and is effectively non-existent as $F \to \infty$ [24].

The NKq-p fitness landscape is an extended form of the NKq for $F = 2$ in order to increase the degree of neutrality in the NKq with its own maximum neutrality. For the NKq-p, the fitness contribution of the i-th locus, f_i in Equation (8.6), is set to 0 with the probability, P $(0 \leq P \leq 1)$, following the way to involve neutrality of the NKp fitness landscape [1] proposed by Barnett. The neutrality of the landscape can be tuned by changing the value of P as well as F. Thus, the neutrality of the landscape is effectively maximized when $F = 2$ as $P \to 1$. When $P = 1$, the landscape becomes completely flat because all the fitness values are set to 0.

8.6.2 Simulation Conditions

The genetic distance is calculated using genotype data as described in Section 8.5. In this study, such genotype data are obtained by applying the standard GA (SGA) [4] to the NKq landscapes; that is, the genotype data are generated during the run of the SGA. Computer simulations were conducted using populations of size M by varying the landscape parameters. The SGA used standard bit mutation as the genetic operation. The per-bit mutation rate was set to q. Crossover was not employed, following Nimwegen's suggestion [26]. Tournament selection was adopted. The tournament size was set to 2 because the SGA generally prefers low selection pressure. A generational model was used. Each run lasted 2,000 generations. We conducted 50 independent runs for each NKq fitness landscape[2] under the

[2] The NKq-p fitness landscape is employed in Section 8.6.4.

landscape parameters, $N = 20$, $K \in \{0,2,6,12,19\}$, $F \in \{2,3,4,6,\infty^3\}$. The results were averaged over the 50 runs.

8.6.3 Simulation Results

8.6.3.1 Measure of Ruggedness

The gradient of the expected offspring fitness versus parent fitness, r, described in Section 8.5.1 is calculated by using the method of least squares on the results of all the runs. Figure 8.3 show rs on the NKq for $F = 2$. It is found, as in [32], that r decreases with the increase of K. That is, the degree of ruggedness increases with the increase of K. This gradient is independent of F (neutrality). We omit these graphs because the same tendency is found for each F.

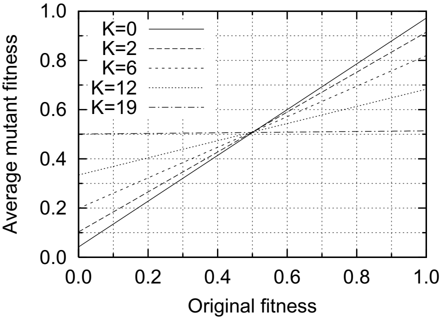


Fig. 8.3 Expected offspring fitness value over all parent fitness values on the NKq fitness landscape for $F = 2$

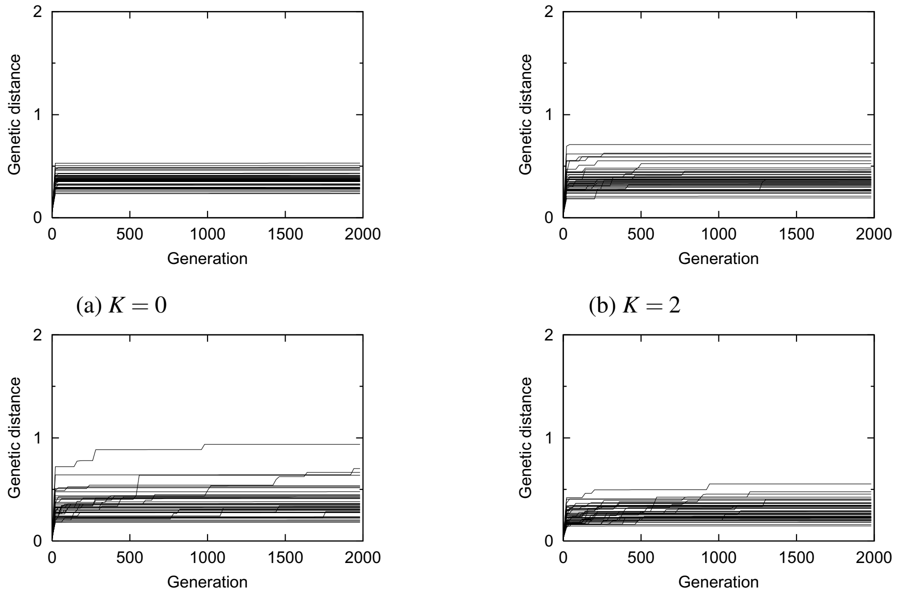
8.6.3.2 Characteristics of Genetic Distance

a) Existence and non-existence of neutrality

The first experiments were conducted to investigate the effect of the existence of neutrality on the transition of the genetic distance with a small mutation rate. Figure 8.4 shows the genetic distance for $K = \{0,2,6,19\}$ and $F = \infty$, where q was set to 0.008 ($\ll 1/N$). They leveled off in the very early generations. This means that the population converged to a certain point in the genotype space while the genetic distance between the generations ($D_{t,t+1}$ in Equation (8.5)) simultaneously became zero.

In contrast, the genetic distance for each generation for $K = \{0,2,6,12,19\}$ and $F \neq \infty$ (for instance, the results for $F = 2$ are shown in Figure 8.5) with $q = 0.008$ increased approximately linearly over generations in all runs. This differentiates

[3] For $F = \infty$, the NK fitness landscape [18] was employed instead of the NKq landscape as [32], which results in practically non-existence of neutrality.



(a) $K = 0$

(b) $K = 2$

(c) $K = 6$

(d) $K = 19$

Fig. 8.4 Genetic distance at each generation with $q = 0.008$ and $M = 50$ for $F = \infty$ in 50 runs

between the existence and the non-existence of neutrality of the fitness landscape. That is, the increase of the genetic distance over generations indicates the presence of neutrality of the fitness landscape.

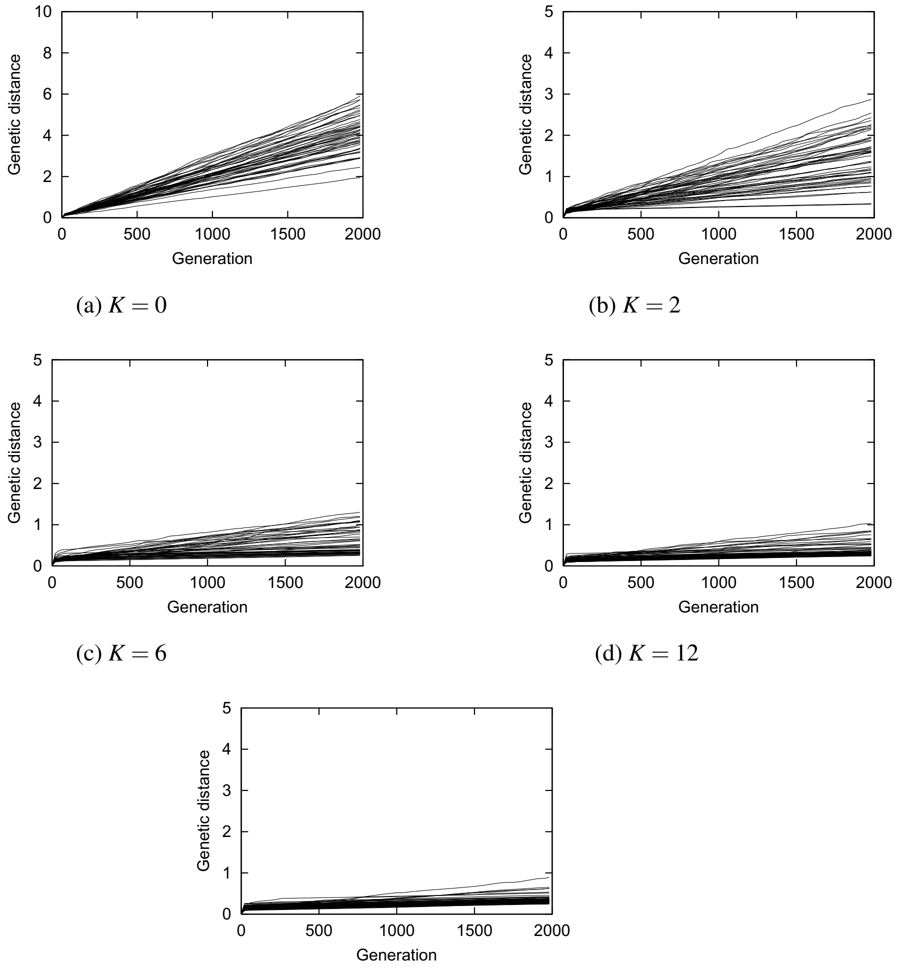
In the remainder of this chapter, the gradient of the genetic distance over generations for $F \neq \infty$ is calculated by using the method of least squares on the results of all the runs and referred to as α.

b) Neutrality

Figure 8.6 shows the gradient of the genetic distance over generations, α, for $q = 0.008$ and $M = \{50, 100, 200, 400\}$. Notice first that α increased with the decrease of F for all Ks. This means that the genetic distance increases with the increase of neutrality. Secondly, α decreased with the increase of K for all Fs. This means that α decreased with the increase of ruggedness; that is, not only neutrality but also ruggedness has an influence on α.

c) Varying the population size

The next experiment was conducted by varying the population size. Figure 8.7 shows α at each K for $q = 0.008$ and $F = \{2,3,4,6\}$. With the increase of the



(a) $K = 0$

(b) $K = 2$

(c) $K = 6$

(d) $K = 12$

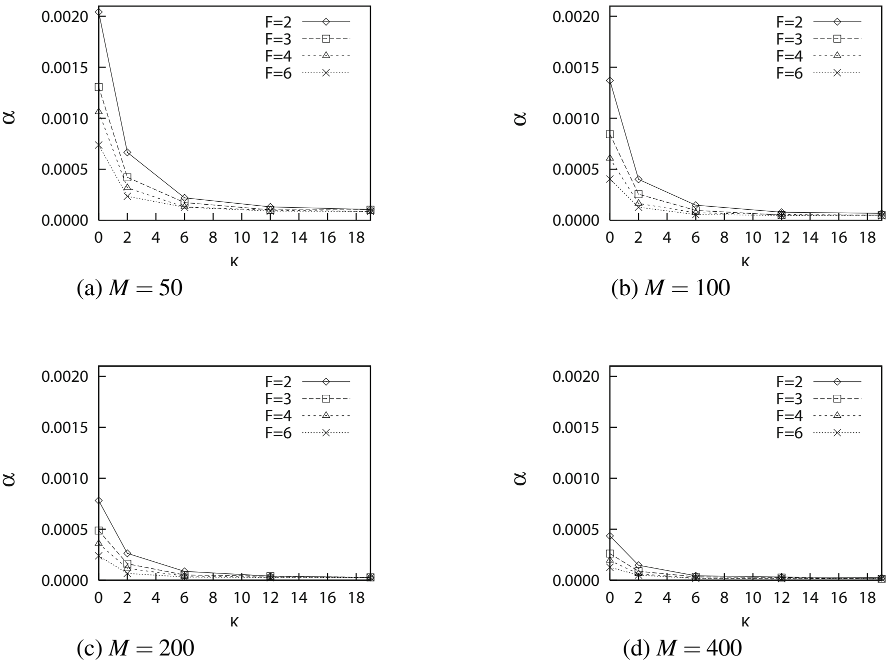
(e) $K = 19$

Fig. 8.5 Genetic distance at each generation with $q = 0.008$ and $M = 50$ for $F = 2$ in 50 runs

population size, α decreased for each K and F. Therefore, the larger the population size becomes, the slower the population moves.

d) Varying the mutation rate

In population genetics, it is assumed that the mutation rate per locus is sufficiently small as mentioned in Section 8.5.2. Thus, the experiments in the previous subsection were conducted using a small mutation rate. In the last series of experiments, the transition of the standard genetic distance were observed by varying the mutation rate from $q = 0.005$ to 0.010 and 0.1 with $M = 50$.



(a) $M = 50$ (b) $M = 100$

(c) $M = 200$ (d) $M = 400$

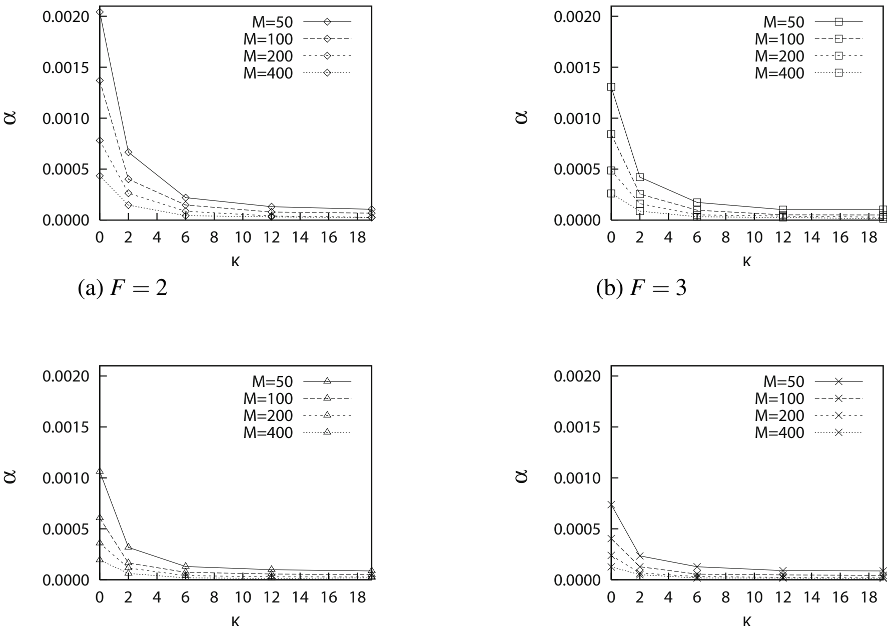
Fig. 8.6 Gradient of genetic distance α at each K with $q = 0.008$ and each M

Figure 8.8 shows the results with $q = \{0.005, 0.006, 0.007, 0.008, 0.009, 0.010\}$. In this range, α increased with the increase of the mutation rate for each K and F. For each q, similar behaviors were observed to the results with $q = 0.008$ in the previous subsections. In contrast, the results with $q = 0.1$ show the different behaviors (Figure 8.9). Surprisingly, α increased with the increase of K for all Fs. In addition to this, no significant differences were found between the graphs of different Fs. This implies that artificial evolution has changed into random search, caused by the mutation rate which is larger than the *error threshold* [1, 25, 26].

From the above, we confirmed that the standard genetic distance depends on the mutation rate, and can be used as long as the mutation rate is sufficiently small compared with the error threshold.

8.6.4 Preparing for Estimating the Degree of Neutrality

In the previous subsection, the gradients of the genetic distance over generations, α, for K and F were examined. In addition to this, we know from the results obtained in Section 8.6.3.1 that r decreases with increase in K; that is, the correlation decreases with increase in K. For real-world problems, the degree of ruggedness of a fitness



(a) $F = 2$

(b) $F = 3$

(c) $F = 4$

(d) $F = 6$

Fig. 8.7 Gradient of genetic distance α at each K with $q = 0.008$ and each F

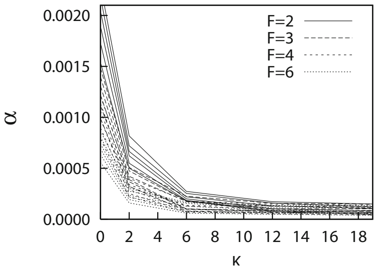


Fig. 8.8 Gradient of genetic distance α with $M = 50$: The solid lines, from left to right, correspond to α for $F = 2$ with $q = \{0.005, 0.006, 0.007, 0.008, 0.009, 0.010\}$. Similarly, the dashed lines correspond to α for $F \in \{3, 4, 6\}$.

landscape can be predicted by fitness correlation [32, 42], which was illustrated on r in Section 8.6.3.1. Therefore, Figure 8.6 will be redrawn by using the same data obtained in the previous subsection to illustrate the relationship between r and α. Figure 8.10 shows α at each correlation, r, corresponding to each K for all Fs

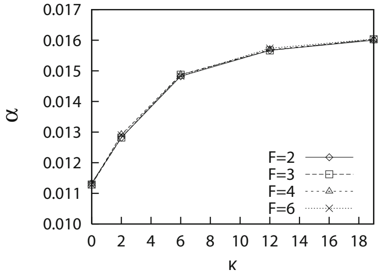


Fig. 8.9 Gradient of genetic distance α with $q = 0.1$ and $M = 50$

in Figure 8.6a. It is confirmed that α increased with increase in r for all Fs and increased with decrease in F for all rs. Of course, these are consistent with the results obtained in the previous subsection.

In addition, in order to fill the empty space in Figure 8.10, the NKq-p with $F = 2$ and $P \in \{0.9, 0.99\}$ was introduced, where the NKq-p was described in Section 8.6.1. Figure 8.11 shows α at each correlation, r, corresponding to each K for all Fs and Ps in the NKq-p. As a result, a set of points (r, α) forms a curve, which increases with the increase of the correlation when F and P are constant. A set of curves is also found with different levels of neutrality. This demonstrates that α would predict the increase of neutrality combined with the measure of ruggedness. The details of the procedure of estimating the degree of neutrality is described in Section 8.7.

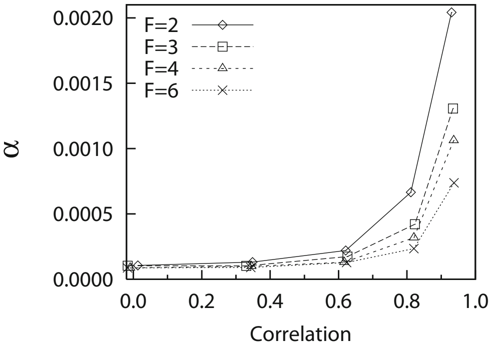


Fig. 8.10 Gradient of genetic distance α as a function of the correlation (r) for F with $q = 0.008$ and $M = 50$

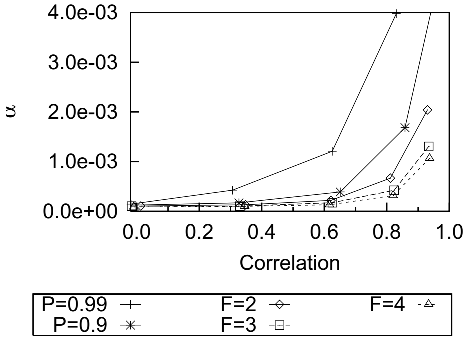


Fig. 8.11 Gradient of genetic distance α as a function of the correlation (r) for F and P with $q = 0.008$ and $M = 50$

8.7 Procedure for Estimating the Degree of Neutrality

In this study, r and α which are described in Section 8.5.1 and 8.5.2 are calculated by using the method of least squares on the result of each run. Thus, the procedure for estimating the degree of neutrality of a fitness landscape on a real-world problem can be summarized as follows:

When the mutation rate per locus is sufficiently small,

i) Confirm the existence of neutrality of a fitness landscape on a real-world problem by investigating transitions of the genetic distance over generations; and
ii) Calculate a point, (r, α), from each run on the real-world problem; and
iii) Calculate a set of r-α curves in test functions with different levels of ruggedness and neutrality; and
iv) By using a set of r-α curves as a baseline, judge where a point, (r, α), obtained from the real-world problem is located on the graph of (r, α). Then estimate indirectly the degree of neutrality of the real-world problem.

The reason why steps (iii) and (iv) are conducted is that the genetic distance is affected by both neutrality and ruggedness as mentioned in Section 8.6. Test functions, the NKq and NKq-p fitness landscapes, are adopted for obtaining a set of r-α curves. The landscape parameters of the NKp and the NKq should be decided at step (iii) for conducting step (iv). In addition to this, the parameters of the GAs (a mutation rate, a population size, a tournament size, etc.) for test functions should be the same as those for a real-world problem due to the characteristics of the genetic distance.

8.8 Estimating the Degree of Neutrality of Evolutionary Robotics Fitness Landscapes

In this section, several computer simulations were conducted with an evolutionary robotics problem, where it is predicted that the evolutionary robotics fitness landscape includes neutrality, in order to investigate the validity of the proposed method in Section 8.7.

8.8.1 The Task and the Fitness Function

The control task used in this experiment was motion pattern discrimination [13, 16], and is based on a task originally implemented by Beer [2]. The agent must discriminate between two types of vertically falling object based on the object's period of horizontal oscillation; it must catch (i.e., move close to) falling objects that have a long period whilst avoiding those with a short period (see Figure 8.12). An array of proximity sensors allow the agent to perceive the falling objects. If an object intersects a proximity sensor, the sensor outputs a value inversely proportional to the distance between the object and the agent. The agent can move horizontally along the bottom of the arena. In our experiment, the agent of diameter 30 had 7 proximity sensors of maximum range 220 uniformly distributed over a visual angle of 45 degrees. The horizontal velocity of the agent was proportional to the sum of the opposing horizontal forces produced by a pair of effectors. It has maximum velocity of 8. Each falling object was circular, with diameter 30, and dropped from the top of the arena with a vertical velocity of 4, a horizontal amplitude of 30 and an initial horizontal offset of ±50. An object's horizontal velocity was ±10 (12 steps in a period) for a long period and ±30 (4 steps in a period) for a short period.

The performance measure to be maximized was as follows:

$$Fitness = 1000 \sum_{i=1}^{NumTrials} \frac{H_i}{NumTrials} \tag{8.7}$$

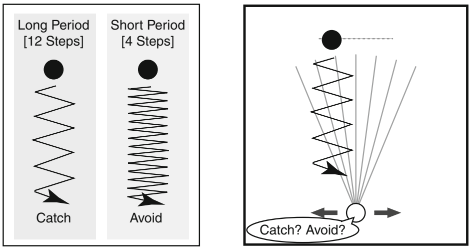


Fig. 8.12 Experimental setup for the discrimination of the motion patterns. Two kinds of period used in the discrimination experiments (left) and the agent in the arena with its array of the proximity sensors (right).

where $H_i = 1 - d_i$ for a long period and $H_i = d_i$ for a short period, $d_i = 1$ when $hd_i > 60$ and $d_i = hd_i/60$ when $hd_i \leq 60$, hd_i is the final horizontal distance between the center of the agent and the object, and $NumTrials$ is the number of trials for an individual (8 trials for each period).

8.8.2 Simulation Conditions

For this experiment, the agent controller was a *spike response model* network [20], which is a form of *Pulsed Neural Network* (PNN) with 7 sensory neurons, 2 fully interconnected motor neurons and N_h fully interconnected hidden neurons, where $N_h \in \{0, 1, 5, 10, 15\}$, in order to estimate and compare the features among the fitness landscapes with each N_h. The network's connection weights and the firing threshold for each neuron were genetically encoded and evolved. The total number of parameters is equal to $\{20, 33, 105, 240, 425\}$ corresponding to each N_h. The parameters were mapped linearly with the following ranges: connection weights $\omega \in [-1.0, 1.0]$, thresholds $\theta \in [0.0, 3.9]$. The parameters of the neurons and synapses were set as follows: $\tau_m = 4$, $\tau_s = 10$, $\Delta^{ax} = 2$ for all neurons and all synapses in the network following the recommendations given in [3].

Computer simulations were conducted using populations of size 50. Each individual was encoded as a binary string with 10 bits for each parameter. Therefore, the total length of the genotype was $L = \{200, 330, 1050, 2400, 4250\}$ for each N_h. The SGA was adopted to evolve PNN parameters. The genetic operation for the SGA was standard bit mutation. Based on the characteristics of genetic distance in Section 8.6 and the procedure in Section 8.7, two types of mutation rate were set as follows:

a) $q = 1/L_{N_h=15}$, which is constant for each landscape, corresponding to $1/L$ for the longest genotype.
b) $q = 1/L$, which is varied for each landscape, following the recommendation in the evolutionary computation community.

According to the procedures in Section 8.7, all the parameters of the SGA must be the same among fitness landscapes to compare the features. Thus, the other parameters were set as follows. Tournament selection was adopted. The tournament size was set to 2. Each run lasted 6,000 generations. We conducted 10 independent runs for each landscape. As the test functions, the NKq and the NKq-p landscapes were adopted. A set of landscape parameters of the NKq and NKq-p at step (iii) in Section 8.7 was decided in the preliminary computer simulations with the same SGA parameters as those of the PNNs. In this experiment, they were set at the same values as those of the NKq and the NKq-p in Section 8.6.4.

Let me explain again the reason to conduct step (iv) in these examples: In the case of a) there are some cases in which it is not possible to estimate and directly compare the degrees of neutrality between fitness landscapes by using only genetic distance (the details will be described in the next subsection), and in the case of b) there is no theoretical meaning to directly compare the degrees of neutrality between the fitness landscapes of the PNNs due to the different mutation rates set for each N_h.

8.8.3 Simulation Results

This subsection shows the results of the computer simulations conducted using the two types of mutation rate mentioned above and discusses them.

a) $q = 1/L_{N_h=15}$

Figure 8.13 shows the genetic distance at each generation for $N_h = 15$. Approximately linear increases were observed for all runs. For other N_hs, the same transitions were observed. From the results obtained in Section 8.6.3.2, this might indicate the presence of neutrality of the fitness landscape of the PNN. Thus, the gradient of the genetic distance over generations, α, is shown by using the method of least squares on the results of all the runs in the following parts.

Figures 8.14, 8.15 and 8.16 show the features of the fitness landscape for each N_h. Figure 8.14 plots the correlation, r, for each N_h. In this experiment, the correlation increased with increase in N_h except for $N_h = 15$. That is, ruggedness decreased with increase in N_h. In Figure 8.15, α increased with increase in N_h. When the fitness landscape for $N_h = 1$ was compared with that of $N_h = 0$, there was no significant differences in the correlation, but α increased much. For another instance, when the fitness landscape for $N_h = 15$ was compared with that of $N_h = 10$, the correlation decreased, but α increased. In these cases, the increase of neutrality can be estimated based on the results obtained in Section 8.6.3.2. Note that the increase of neutrality cannot be estimated either when the fitness landscape for $N_h = 5$ was compared with that of $N_h = 1$ or when the fitness landscape for $N_h = 10$ was compared with that of $N_h = 5$, because the increase of α coincided with the increase of the correlation.

By using the r-α curves obtained in the NKq and NKq-p landscapes as baselines, where F and P are constant for each curve, the degrees of neutrality of the landscape for all N_hs were estimated indirectly (Figure 8.16). The points, (r, α), for $N_h = 0$ and $N_h = 1$ are located between the r-α curves for $P = 0.99$ and $P = 0.9$. The point for $N_h = 1$ is nearer to the curve for $P = 0.99$ than the point for $N_h = 0$. The points for $N_h = 5$ and $N_h = 15$ are located on the curve for $P = 0.9$. The point for $N_h = 10$ is located between the the curves for $P = 0.9$ and $F = 2$. Therefore, the degrees of neutrality of the landscape for N_h are ranked from lowest to highest as follows: $N_h = 10 \rightarrow \{5, 15\} \rightarrow 0 \rightarrow 1$.

When the fitness landscape for $N_h = 1$ was compared with that of $N_h = 0$ and when the fitness landscape for $N_h = 15$ was compared with that of $N_h = 10$, increases in neutrality were observed. These are consistent with the results of the

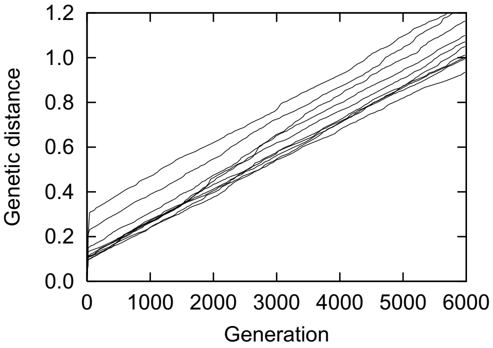


Fig. 8.13 Genetic distance at each generation for the SGA with $q = 1/L$ for $N_h = 15$ in 10 runs

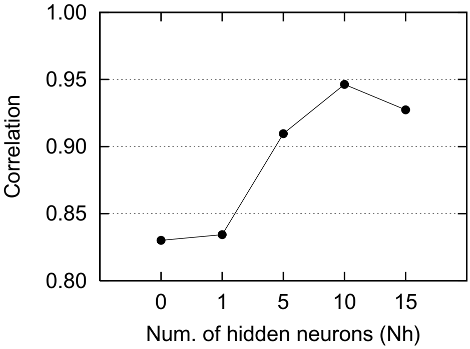


Fig. 8.14 Correlation (r) for each N_h at $q = 1/L_{N_h=15}$

direct comparison between the PNN landscapes mentioned above. Additionally, decreases in neutrality were also observed when the fitness landscape for $N_h = 5$ was compared with that of $N_h = 1$ and when the fitness landscape for $N_h = 10$ was compared with that of $N_h = 5$ although it was not possible to estimate these decreases by direct comparison between the PNN landscapes.

Figure 8.17 shows the maximum fitness at each generation for N_h. Except for $N_h = 1$, the fitness increased faster as the decrease in N_h, that is, the decrease in the genotypic search space. The poor performance in the SGA for $N_h = 1$ cannot be predicted by the comparison with $N_h = 0$ only on the correlation in Figure 8.14. However, it would be predicted by the large degree of neutrality for $N_h = 1$ observed in Figure 8.16.

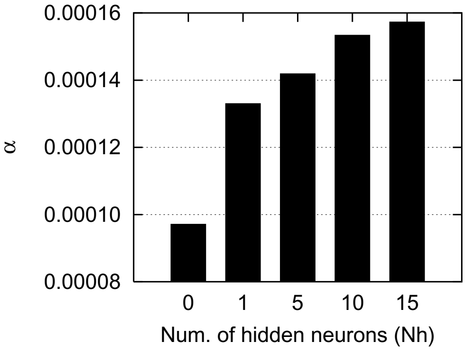


Fig. 8.15 Gradient of genetic distance α for each N_h at $q = 1/L_{N_h=15}$

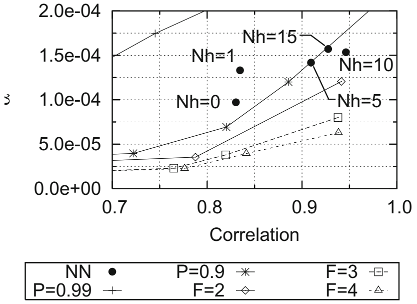


Fig. 8.16 Gradient of genetic distance α as a function of the correlation (r) for each N_h and the TNKq-p at $q = 1/L_{N_h=15}$

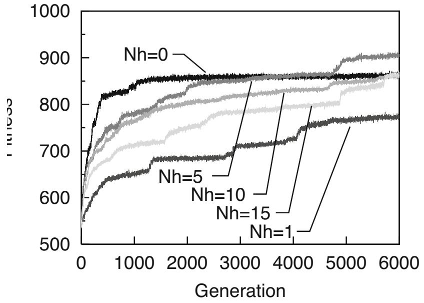


Fig. 8.17 Maximum fitness at $q = 1/L_{N_h=15}$

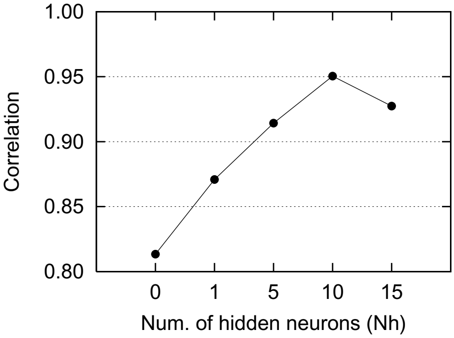
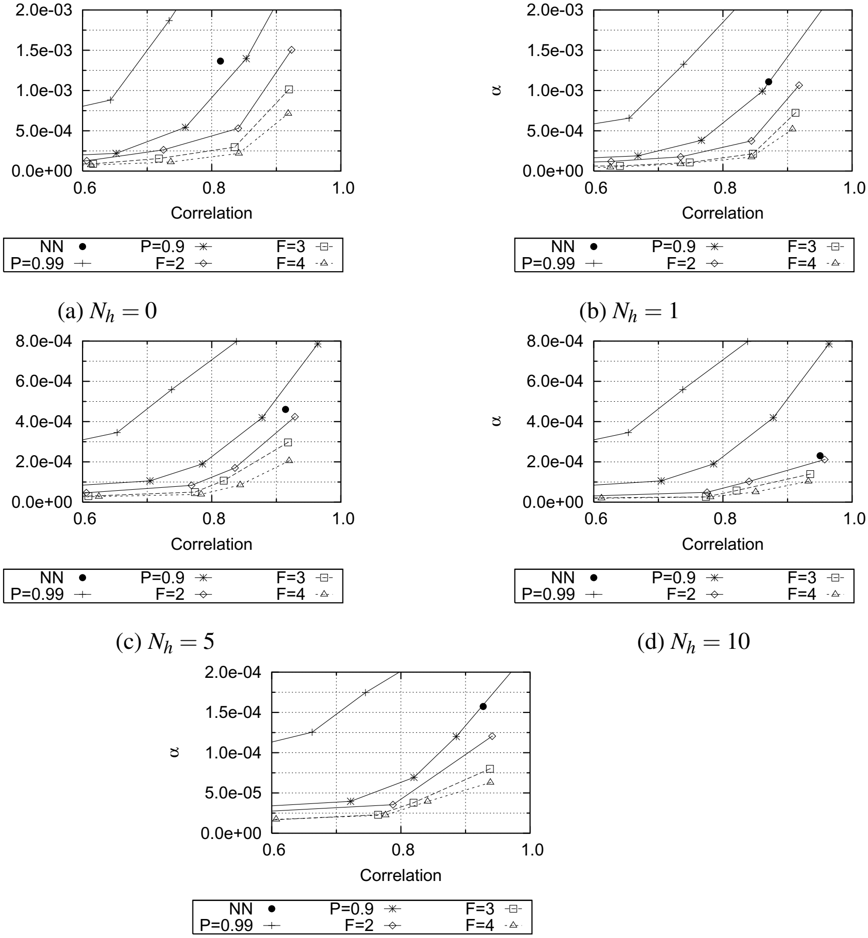


Fig. 8.18 Correlation (r) for each N_h at $q = 1/L$



(a) $N_h = 0$

(b) $N_h = 1$

(c) $N_h = 5$

(d) $N_h = 10$

(e) $N_h = 15$

Fig. 8.19 Gradient of genetic distance α as a function of the correlation (r) for each N_h and the NKq and NKq-p at $q = 1/L$

b) $q = 1/L$

Figure 8.18 plots the correlation, r, for each N_h. As observed in a), the correlation increased with increase in N_h except for $N_h = 15$. Because of different qs among the landscapes, the degrees of neutrality of the landscape for all N_hs can be estimated indirectly by using the r-α curves obtained in the NKq and NKq-p landscapes (Figure 8.19). Thus, the degrees of neutrality of the landscape for N_h are ranked from lowest to highest as follows: $N_h = 10 \rightarrow 5 \rightarrow \{1, 15\} \rightarrow 0$.

Figure 8.20 shows the maximum fitness at each generation for N_h. The fitness increased faster as the decrease in N_h, that is, the decrease in the genotypic search space. This can be explained as follows: In the process of evolution, no error threshold effects were observed [4]. This implies that the effective mutation rate at $q = 1/L$ would be below the error threshold under each condition [14, 15, 16]. In addition, the degrees of ruggedness were not so high for this problem (Figure 8.18). Therefore, the neutrality of the landscapes would not explicitly have any influence on the performances of the SGA but the effect of the size of the genotypic search space would be dominant.

From the viewpoint of the structure of evolutionary artificial neural networks, it was confirmed by the proposed approach that the number of hidden neurons have a great influence on the features of the fitness landscape of neural networks.

From these results, we can conclude that the proposed method can estimate and indirectly compare the degree of neutrality of the fitness landscapes by using the features of the fitness landscape of test functions as baselines.

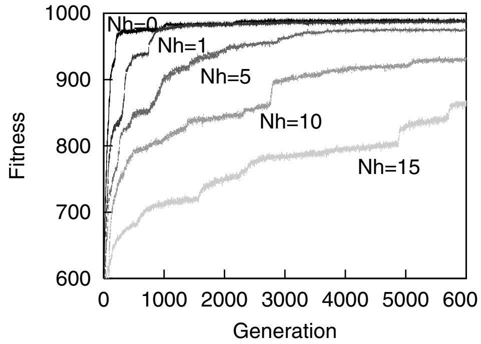


Fig. 8.20 Maximum fitness at $q = 1/L$

[4] Generally, an error threshold sets the upper limit for a mutation rate that will enable efficient search [1, 25, 26].

8.9 Summary and Conclusions

In this chapter, I proposed the use of Nei's standard genetic distance, which originates from population genetics, for estimating the degree of neutrality of fitness landscapes. The characteristics of genetic distance in tunably neutral NK landscapes were investigated and the guideline was provided to apply our approach to a real-world problem. Several computer simulations were conducted with an evolutionary robotics problem where neutrality would be included in the fitness landscape. The results obtained in this study suggest the validity of the proposed approach. By using this, we might expect to explain evolutionary dynamics on problems where they have not been explained by canonical fitness landscape descriptions. In a recent work [10], we applied our approach to estimate features of competitive co-evolutionary fitness landscapes on a predator-prey problem in computer simulations and investigated the Red Queen effect on the fitness landscape. Those results suggest that our method can track the progress of fitness landscapes on competitive co-evolutionary robotics.

In this study, I assume the use of a binary representation for the genotype. As another research direction, we would also like to extend our approach to other representations.

The results can be summarized from the research objectives investigating the geography of a fitness landscape presented in Section 8.1 as follows:

> "to analyze the performance of the search algorithm after the run."

- I explained the performance of the SGA using the estimates of features of the fitness landscape obtained in the experiment. However, the performance of the SGA was not explained enough because it was based in part on conjecture.
- Generally speaking, it is still unclear what relation exists between the estimates of features of the fitness landscape and the performance of an evolutionary algorithm applied to the problem.

> "to quantify the difficulty of a problem to be solved by an evolutionary algorithm in advance then to select an useful evolutionary algorithm, including parameter tuning for it."

- According to the results obtained in this study, the parameters of the extended GA [14, 27] were tuned for the same problem used in this study. The extended GA showed good performance on the problem [16].
- To date, a proper methodology has not been developed to select an useful evolutionary algorithm or tune the parameters of an evolutionary algorithm based on the estimates of features of the fitness landscape.

These suggest future research directions, including the discussion whether another feature of fitness landscapes is needed.

References

[1] Barnett, L.: Tangled webs: Evolutionary dynamics on fitness landscapes with neutrality. Master's thesis, School of Cognitive and Computing Sciences, Sussex University, Brighton, UK (1997)

[2] Beer, R.: Toward the evolution of dynamical neural networks for minimally cognitive behavior. In: Maes, P., Mataric, M., Meyer, J., Pollack, J., Wilson, S. (eds.) Proceedings of From Animals to Animats 4, pp. 421–429. MIT Press, Cambridge (1996)

[3] Floreano, D., Mattiussi, C.: Evolution of spiking neural controllers for autonomous vision-based robots. In: Gomi, T. (ed.) ER 2001. LNCS, vol. 2217, pp. 38–61. Springer, Heidelberg (2001)

[4] Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston (1989)

[5] Harvey, I.: Artificial evolution for real problems. In: Gomi, T. (ed.) Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER 1997). AAI Books, Tokyo (1997)

[6] Hordijk, W.: A measure of landscapes. Evolutionary Computation 4(4), 335–360 (1994)

[7] Horn, J., Goldberg, D.E.: Genetic algorithm difficulty and the modality of fitness landscapes. In: Whitley, L.D., Vose, M.D. (eds.) Foundations of Genetic Algorithms 3, pp. 243–269. Morgan Kaufmann, San Francisco (1995)

[8] Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Eshelman, L.J. (ed.) Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 184–192. Morgan Kaufmann, San Mateo (1995)

[9] Kallel, L.: Inside GA dynamics: Ground basis for comparison. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P., et al. (eds.) PPSN V 1998. LNCS, vol. 1498, pp. 57–66. Springer, Heidelberg (1998)

[10] Katada, Y., Handa, Y.: Tracking the red queen effect by estimating features of competitive co-evolutionary fitness landscapes. In: Fogel, G., et al. (eds.) Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC 2010), pp. 4417–4424. IEEE Press (2010)

[11] Katada, Y., Ohkura, K.: Estimating the degree of neutrality in fitness landscapes by the nei's standard genetic distance – an application to evolutionary robotics –. In: Yen, P.G., et al. (eds.) Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC 2006), pp. 1590–1597. IEEE Press (2006)

[12] Katada, Y., Ohkura, K.: Analysis on topologies of fitness landscapes with both neutrality and ruggedness based on neutral networks. In: Rothlauf, F. (ed.) Proceedings of the 2009 Genetic and Evolutionary Computation Conference (GECCO 2009), pp. 1855–1856 (2009)

[13] Katada, Y., Ohkura, K., Ueda, K.: Artificial evolution of pulsed neural networks on the motion pattern classification system. In: Nakauchi, Y. (ed.) Proceedings of 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp. 318–323 (2003)

[14] Katada, Y., Ohkura, K., Ueda, K.: Tuning genetic algorithms for problems including neutral networks -a more complex case: The terraced nk problem-. In: Wang, P.P. (ed.) Proceedings of the 7th Joint Conference on Information Sciences, pp. 1661–1664 (2003)

[15] Katada, Y., Ohkura, K., Ueda, K.: Tuning genetic algorithms for problems including neutral networks -the simplest case: The balance beam function-. In: Wang, P.P. (ed.) Proceedings of the 7th Joint Conference on Information Sciences, pp. 1657–1660 (2003)

[16] Katada, Y., Ohkura, K., Ueda, K.: An approach to evolutionary robotics using a genetic algorithm with a variable mutation rate strategy. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 952–961. Springer, Heidelberg (2004)

[17] Katada, Y., Ohkura, K., Ueda, K.: The nei's standard genetic distance in artificial evolution. In: Greenwood, G.W. (ed.) Proceedings of the 2004 IEEE Congress on Evolutionary Computation (CEC 2004), pp. 1233–1239. IEEE Press (2004)

[18] Kauffman, S.A.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York (1993)

[19] Kimura, M.: The Neutral Theory of Molecular Evolution. Cambridge University Press, New York (1983)

[20] Maass, W., Bishop, C.: Pulsed Neural Networks. MIT press (1998)

[21] Manderick, B., Weger, M.D., Spiessens, P.: The genetic algorithm and the structure of the fitness landscape. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 143–150. Morgan Kaufmann, San Mateo (1991)

[22] Mitchell, M., Forrest, S., Holland, J.H.: The royal road for genetic algorithms: Fitness landscapes and GA performance. In: Varela, F.J., Bourgine, P. (eds.) Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 1991, pp. 245–254. A Bradford Book, The MIT Press, Paris (1992)

[23] Nei, M.: Genetic distance between populations. The American Naturalist 106, 283–292 (1972)

[24] Newman, M., Engelhardt, R.: Effect of neutral selection on the evolution of molecular species. Proc. R. Soc. London B 265, 1333–1338 (1997)

[25] van Nimwegen, E., Crutchfield, J.P.: Optimizing epochal evolutionary search: Population-size dependent theory. SFI Working Paper 9810-090 (1998)

[26] van Nimwegen, E., Crutchfield, J.P., Mitchell, M.: Statistical dynamics of the royal road genetic algorithm. Theoretical Computer Science 229(1), 41–102 (1999)

[27] Ohkura, K., Ueda, K.: Adaptation in dynamic environment by using GA with neutral mutations. International Journal of Smart Engineering System Design 2, 17–31 (1999)

[28] Ohta, T.: The nearly neutral theory of molecular evolution. Annu. Rev. Ecol. Syst. 23, 263–286 (1992)

[29] Ohta, T.: Evolution by nearly-neutral mutations. Genetica 102/103, 83–90 (1998)

[30] Simon, H.: The Sciences of the Artificial. MIT Press, Cambridge (1969)

[31] Smith, T.: The evolvability of artificial neural networks for robot control. Ph.D. thesis, School of Biological Sciences, University of Sussex (2002)

[32] Smith, T., Husbands, P., Layzell, P., O'Shea, M.: Fitness landscapes and evolvability. Evolutionary Computation 10(1), 1–34 (2002)

[33] Smith, T., Husbands, P., O'Shea, M.: Neutral networks and evolvability with complex genotype-phenotype mapping. In: Kelemen, J., Sosík, P. (eds.) ECAL 2001. LNCS (LNAI), vol. 2159, pp. 272–281. Springer, Heidelberg (2001)

[34] Smith, T., Husbands, P., O'Shea, M.: Neutral networks in an evolutionary robotics search space. In: Kim, J.H., et al. (eds.) Proceedings of the 2001 IEEE Congress on Evolutionary Computation, CEC 2001, pp. 136–145. IEEE Press, Piscataway (2001)

[35] Smith, T., Philippides, A., Husbands, P., O'Shea, M.: Neutrality and ruggedness in robot landscapes. In: Fogel, D., et al. (eds.) Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), pp. 1348–1353. IEEE Press (2002)

[36] Stadler, P.F.: Landscapes and their correlation functions. J. Math. Chem. 20, 1–45 (1996)

[37] Thompson, A.: An evolved circuit, intrinsic in silicon, entwined with physics. In: Higuchi, T., Iwata, M., Weixin, L., et al. (eds.) ICES 1996. LNCS, vol. 1259, pp. 390–405. Springer, Heidelberg (1997)

[38] Vassilev, V.K., Miller, J.F.: The advantages of landscape neutrality in digital circuit evolution. In: Miller, J.F., Thompson, A., Thompson, P., Fogarty, T.C. (eds.) ICES 2000. LNCS, vol. 1801, pp. 252–263. Springer, Heidelberg (2000)

[39] Vassilev, V.K., Miller, J.F., Fogarty, T.C.: Information characteristics and the structure of landscapes. Evolutionary Computation 8(1), 31–60 (2000)

[40] Verel, S.: Fitness landscapes and graphs: Multimodularity, ruggedness and neutrality. In: Rothlauf, F. (ed.) Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference, Tutorial, pp. 3593–3655 (2009)

[41] Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of nk landscapes with neutrality. IEEE Transactions on Evolutionary Computation 15(6), 783–797 (2011)

[42] Weinberger, E.D.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cybernetics 63, 325–336 (1990)

[43] Wright, S.: The role of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones, D. (ed.) Proceedings of the Sixth International Congress on Genetics, vol. 1, pp. 356–366 (1932)

Chapter 9
Local Optima Networks: A New Model of Combinatorial Fitness Landscapes

Gabriela Ochoa, Sébastien Verel, Fabio Daolio, and Marco Tomassini

Abstract. This chapter overviews a recently introduced network-based model of combinatorial landscapes: Local Optima Networks (LON). The model compresses the information given by the whole search space into a smaller mathematical object that is a graph having as vertices the local optima and as edges the possible weighted transitions between them. Two definitions of edges have been proposed: basin-transition and escape-edges, which capture relevant topological features of the underlying search spaces. This network model brings a new set of metrics to characterize the structure of combinatorial landscapes, those associated with the science of complex networks. These metrics are described, and results are presented of local optima network extraction and analysis for two selected combinatorial landscapes: NK landscapes and the quadratic assignment problem. Network features are found to correlate with and even predict the performance of heuristic search algorithms operating on these problems.

9.1 Introduction

The fitness landscape metaphor appears most commonly when describing the dynamics of evolutionary algorithms, and its origins are attributed to the population

Gabriela Ochoa
Department of Computing Science and Mathematics, School of Natural Sciences,
University of Stirling, Stirling, FK9 4LA Scotland, UK
e-mail: gabriela.ochoa@cs.stir.ac.uk

Sébastien Verel
INRIA Lille - Nord Europe and University of Nice Sophia-Antipolis,
06903 Sophia Antipolis Cedex, France
e-mail: verel@3s.unice.fr

Fabio Daolio · Marco Tomassini
Department of Information Systems, University of Lausanne,
1015 Lausanne, Switzerland
e-mail: {fabio.daolio,marco.tomassini}@unil.ch

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_9, © Springer-Verlag Berlin Heidelberg 2014

geneticist Sewall Wright [50]. However, the metaphor can be used for computational search in general; the search space can be regarded as a spacial structure where each point (candidate solution) has a height (objective function value) forming a landscape surface. In this scenario, the search process would be an adaptive-walk over a landscape that can range from having many peaks of high fitness boarding deep cliffs to valleys of low fitness, to being smooth, with low hills and gentle valleys.

Identifying the landscape features affecting the effectiveness of heuristic search algorithms is relevant for both predicting their performance and improving their design. Some properties of landscapes that are known to have a strong influence on heuristic search are the number of local optima or peaks in the landscape, the distribution of the local optima in the search space, the correlation between fitness values of neighboring points in the landscape, the topology of the basins of attraction of the local optima, and the presence of neutrality (different search points having the same objective value). Statistical methods have been proposed to measure some of these properties, for example, fitness-distance correlation [19], distributions of solutions density [37], landscape correlation functions [47], and the negative slope coefficient [42]. These metrics work by sampling the landscape surface to provide an approximation of its shape. An alternative view, first introduced in chemical physics in the study of energy landscapes [39], is to construct a network formed by the landscape local optima (minima or maxima). In this view of energy surfaces, the network's vertices are energy minima and there is an edge between two minima if the system can jump from one to the other with an energy cost of the order of the thermal energies. Usually this "transition state" goes through a low energy barrier such as a saddle point in the surface. The resulting graph has been referred to as an *inherent network*. Recent work by Doye and coworkers and by Caflish and coworkers [13, 14, 34] has shown the benefits of this approach: it provides a synthetic view of the energy landscape and the network can be studied using appropriate statistical methods to characterize it in various ways [30]. For example, Doye et al. [13, 14] found that the inherent networks of the energy landscapes of small atomic clusters are often of the scale-free type with a power-law degree distribution function, featuring a kind of single or multiple "funnel" structure. The global energy minimum is the most highly connected node at the bottom of the funnel. This means that the path to the global energy minimum is easy to follow starting anywhere in the energy landscape. The concept of community structure of a network, introduced first for social networks [30], has also been applied, showing that in some cases energy minima split into almost separate groups or communities [27]. This effect is even more spectacular for polypeptides [17]. This kind of information is invaluable for understanding the dynamics induced on the energy landscape such as cluster rearrangements or protein folding.

The *local optima networks* fitness landscape model, described in this chapter, adapts the notion of the inherent network of energy surfaces to the realm of combinatorial (discrete) search spaces. As for energy surfaces (which exist in continuous space), the vertices correspond to solutions that are minima or maxima of the associated combinatorial problem, but edges are defined differently. The combinatorial counterpart considers oriented and weighted edges. In a first version, the weights represent an approximation to the probability of transition between the respective basins in a given direction [32, 33, 41, 44, 45]. This definition, although informative, produced densely connected networks and required exhaustive sampling of the basins of attraction. A second version, *escape edges* was proposed in [43], which does not require a full computation of the basins. Instead, these edges account for the chances of escaping a local optimum after a controlled mutation (e.g. 1 or 2 bit-flips in binary space) followed by hill-climbing. As a first benchmark case in the study of local optima networks, the well studied family of abstract landscapes, the Kauffman's *NK* model, was selected [20, 21]. In this model the ruggedness, and hence the difficulty of the landscape, can be tuned from easy to hard. Two NK models incorporating *neutrality* (i.e. extended regions of equal or quasi-equal fitness) were considered: the NK_p ('probabilistic' *NK*) [3], and NK_q ('quantized' *NK*) [29] families. Subsequently, a more complex and realistic search space was studied. Specifically, the quadratic assignment problem (QAP) introduced by Koopmans and Beckmann [24], which is known to be NP-hard [38].

The local optima network model captures in detail the number and distribution of local optima in the search space; features which are known to be of utmost importance for understanding the search difficulty of the corresponding landscape. This understanding may be exploited when designing efficient search algorithms. For example, it has been observed in many combinatorial landscapes that local optima are not randomly distributed, rather they tend to be clustered in a "central massif" (or "big valley" if we are minimizing). This globally convex landscape structure has been observed in the *NK* family of landscapes [20, 21], and in many combinatorial optimization problems, such as the traveling salesman problem [6], graph bipartitioning [28], and flowshop scheduling [35]. Search algorithms exploiting this global structure have been proposed [6, 35]. For the travelling salesman problem, the big-valley structure holds in much of the search space. However, it has been recently found that the big-valley structure disappears, giving rise to multiple funnels, around local optima that are very close to the global optimum [18]. A specialized crossover operator has been proposed to exploit and overcome this multi-funnel structure [49].

The analysis of local optima networks so far has shown interesting correlations between network features and known search difficulty on the studied combinatorial problems. This chapter overviews the conception and analysis of local optima networks. A brief account of the science of complex networks is given before describing the combinatorial landscapes, relevant definitions and methods employed. A summary of the most relevant results of the analysis is presented, and finally, the prospects of this research effort are discussed.

9.2 The Science of Complex Networks

The last few years have seen an increased interest in the structure of the big networks that form part of our daily environment such as the World Wide Web, the Internet, transportation and electrical power networks, web-based social networks such as

Facebook, and many others. These networks have properties that are unparalleled in simple graphs such as lattices, properties that are akin to those of *complex systems* in general. In these systems, it is difficult or even impossible to infer global behaviors given the rules that are obeyed by the system components and their interactions. For this reason, these big networks are called *complex networks* and their structure gives rise to a wide range of dynamical behaviors. Since this chapter draws heavily on complex network nomenclature and methods, to make it self-contained to a large extent, we give a brief introduction to the field. There exist many references on complex networks: a technical but still very readable introductory book is [12], while [31] is a comprehensive reference.

Mathematically, networks are just *graphs* $G(V,E)$ where V is the set of *vertices* and E is the set of edges that join pairs of vertices. A complex network class that enjoys a precise mathematical description is *random graphs* which are introduced below. Random graphs are a useful abstraction that can sometimes be used to model real networks or, at least, to compare with actual complex networks.

9.2.1 Random Graphs

The random-graph model was formally defined by Erdös and Rényi at the end of the 1950s. In its simplest form, the model consists of N vertices joined by edges that are placed between pairs of vertices uniformly at random. In other words, each of the possible $N(N-1)/2$ edges is present with probability p and absent with probability $1-p$. The model is often referred to as $G_{N,p}$ to point out that, rigorously speaking, there is no such thing as a random graph, but rather an ensemble $G_{N,p}$ of equiprobable graphs.

Another closely related model of a random graph considers the family of graphs $G_{N,M}$ with N vertices and exactly M edges. For $0 \leq M \leq \binom{N}{2}$, there are $s = \binom{N(N-1)/2}{M}$ graphs with M edges. If the probability of selecting any one of them is $1/s$, then the ensemble $G_{N,M}$ is called the family of uniform random graphs. For $M \simeq pN$, the two models are very similar, but we shall use $G_{N,p}$ in what follows.

A few simple facts are worth noting about random graphs. The *average degree* $\bar{k}$ of a graph G is the average of all the vertex degrees in G: $\bar{k} = (1/N)\sum_{j=1}^{N} k_j$, where k_j is the degree of vertex j. If $|E| = M$ is the number of edges in G, then $M = (N\bar{k})/2$, since $\sum_{j=1}^{N} k_j = 2M$ (each edge is counted twice).

The expected number of edges of a random graph belonging to $G_{N,p}$ is clearly $(1/2)N(N-1)p$, but since each edge has two ends, the average number of edge ends is $N(N-1)p$, which in turn means that the average degree of a vertex in a random graph is

$$\bar{k} = \frac{N(N-1)p}{N} = (N-1)p \simeq Np \tag{9.1}$$

for sufficiently large N.

An important property of a connected random graph is that the average path length, i.e the mean distance between nodes, is of the order of $\log N$, which means that any two nodes are only a short distance apart since $\log N$ grows very slowly with increasing N.

9.2.2 Other Network Topologies

Random graphs are interesting objects as they obey, in a probabilistic sense, general mathematical properties. They are also a useful model for generating problem instances for testing network algorithms, and they are used in other ways too. But are random graphs a useful model of the networks that permeate society? Actually, social scientists felt qualitatively as early as the 1950s that social and professional links and acquaintances did not follow a random structure. For example, if a person has some relationship with two others, then the latter two are more likely to know each other than are two arbitrary persons. This does not fit the random-graph model, however, where the likelihood that two given nodes are connected is the same independent of any other consideration. In a ground-breaking paper, Watts and Strogatz [46] proposed a simple network construction algorithm that gives rise to graphs having the following properties: the path length from any node to any other node is short, as in random graphs; but, unlike random graphs, there is local structure in the network. Watts and Strogatz called their networks *small-world networks*, a term that has been in use for a long time in the field of social games to indicate that there is a small separation between any two persons in a large social network.

The discovery of these new properties was made possible by the abundance of online network data and the computer power to analyze these data; something that was not available to social scientists at earlier times. Many networks have been studied since, both man-made and natural: the Internet, the World Wide Web, scientific collaboration and coauthorship networks, metabolic and neural networks, air traffic, telephone calls, e-mail exchanges, and many others [31]. Most of these studies have confirmed that, indeed, real networks are not random in the sense of random-graph theory, and they possess a number of quite interesting properties.

Some definitions of global and local network properties that will be used in the rest of the chapter are described below.

9.2.3 Some Graph Statistics

Drawing and visualizing a network with up to a few tens of nodes may help in understanding its structure. However, when there are thousands of nodes, this is no longer possible. For this reason, a number of statistics have been proposed to describe the main features of a graph. Taken together, these statistics characterize the nature of a network.

Four statistics are particularly useful: the average degree, already defined in Section 9.2.1, the *clustering coefficient*, the *average path length*, and the *degree distribution function*. We shall now briefly describe these graph measures. A fuller treatment can be found in [31].

9.2.3.1 Clustering Coefficient

Here we use the following definition of clustering: consider a particular node j in a graph, and let us assume that it has degree k, i.e. it has k edges connecting it to its k neighboring nodes. If all k vertices in the neighborhood were completely connected to each other, forming a clique, then the number of edges would be equal to $\binom{k}{2}$. The clustering coefficient C_j of node j is defined as the ratio between the e edges that actually exist between the k neighbors and the number of possible edges between these nodes

$$C_j = \frac{e}{\binom{k}{2}} = \frac{2e}{k(k-1)} \tag{9.2}$$

Thus C_j is a measure of the "cliquishness" of a neighborhood: the higher the value of C_j, the more likely it is that two vertices that are adjacent to a third one are also neighbors of each other.

For example, in Figure 9.1, the leftmost case has $C_j = 0$ since none of the links between node j's neighbors is present. In the middle figure, three out of the possible six links are present and thus $C_j = (2 \times 3)/(4 \times 3) = 6/12 = 0.5$, while in the rightmost case $C_j = 1$ since all six links between j's neighbors are present.

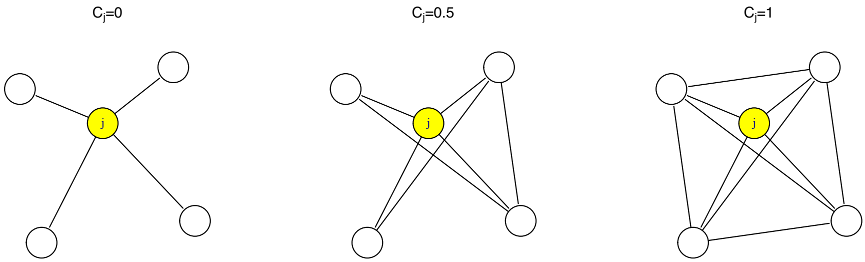


Fig. 9.1 In the left image the clustering coefficient of node j, $C_j = 0$ since there are no links among j's neighbors. In the middle image $C_j = 0.5$ because three out of the possible six edges among the neighbors of j are present. In the right image $C_j = 1$ as all the edges that could be there are actually present (it is a clique).

The *average clustering coefficient* $\bar{C}$ is the average of C_i over all N vertices $i \in V(G)$: $\bar{C} = (1/N)\sum_{i=1}^{N} C_i$. The clustering coefficient of a graph G thus expresses the degree of locality of the connections.

The clustering coefficient of a random graph is simply $\bar{k}/N \simeq p = \bar{C}$, where N is the total number of vertices and p is the probability that there is an edge between any two vertices since all edges are equiprobable and uncorrelated. One thus sees that the clustering coefficient of a random graph decreases with the graph size N and approaches 0 for $N \to \infty$. The clustering coefficient of a complete graph is 1, since each of a node's neighbors are connected to each other by definition.

9.2.3.2 Average Path Length

The *distance* between two nodes i and j is defined as the number of edges between i and j. We denote the shortest path between nodes $i, j \in V(G)$ by l_{ij} as being the path with the shortest distance. The average, or mean, path length $\bar{L}$ of G is then defined as

$$\bar{L} = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j>i} l_{ij} \tag{9.3}$$

The normalizing constant $2/N(N-1)$ is the inverse of the total number of pairs of vertices. If there is no path between any two nodes, their distance is conventionally set to ∞ (note that Equation 9.3 does not hold in this case).

The mean path length gives an idea of "how long" it takes to navigate a connected network. Random graphs and small-world networks share the property that $\bar{L}$ scales as $\log N$ and thus most vertices in these networks are connected by a short path. This is not the case in d-dimensional regular lattice graphs, where $\bar{L}$ scales as $N^{1/d}$. For instance, in a ring $\bar{L}$ scales linearly with N and is inversely proportional to k, the number of neighbors.

9.2.3.3 Degree Distribution Function

The degree distribution $P(k)$ of an undirected graph G is a function that gives the probability that a randomly selected vertex has degree k. $P(k)$ can also be seen as the fraction of vertices in the graph that have degree k. Similar definitions also apply for the in-links and out-links of the vertices in a directed graph for which one can define a degree distribution function for both the outgoing $P_{out}(k)$ and the the incoming $P_{in}(k)$ links.

For a random graph with connection probability p, the probability $P(k)$ that a random node has degree k is given by

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k} \tag{9.4}$$

This is the number of ways in which k edges can be selected from a certain node out of the $N-1$ possible edges, given that the edges can be chosen independently of each other and have the same probability p. Thus $P(k)$ is a binomial distribution peaked at $P(\bar{k}) \simeq Np$, as already found in Equation (9.1). Since this distribution has a rapidly diminishing tail, most nodes will have similar degrees. Low- and high-degree nodes, say a few standard deviations away from the mean, have a negligible probability, since the tails fall off very rapidly. Networks having this degree distribution will thus be rather homogeneous as far as the connectivity is concerned. For large N and for pN constant, the binomial distribution can be well approximated by the Poisson distribution of mean $\bar{k} = Np$:

$$P(k) = e^{-\bar{k}} \frac{\bar{k}^k}{k!} \tag{9.5}$$

Another rapidly-decaying degree distribution function that appears in model graphs is the exponential distribution:

$$P(k) \propto e^{-k/\bar{k}} \tag{9.6}$$

This degree distribution results when nodes are progressively added to a growing network such that a new node has the same probability of forming a link with any of the already existing nodes. Most real networks, however, do not show this kind of behavior. Instead, the so called *scale-free networks*, a model of which was first proposed by Barabásis and Albert [2], seem to be closer to real life networks. In these networks, $P(k)$ follows a power-law distribution:

$$P(k) = ck^{-\gamma}, \tag{9.7}$$

where c and γ are positive constants.

In scale-free networks, while most nodes have a low degree, there is a small but non-negligible number of highly connected nodes. This structure has a profound influence on the dynamics of processes taking place on those networks. It is worth mentioning that this model has been recently criticized [8] as it turns out that, upon close inspection, many empirical data-sets in the literature that were previously assumed to have a power-low distribution are better modeled by alternative distributions.

Poisson, exponential, and power-law distributions are characteristic of model random and scale-free graphs respectively. The empirical distribution functions found for real-life graphs are seldom of this type though, because it is almost impossible to find such "pure" networks among finite sampled ones. However, most real networks have degree distributions that are fat-tailed, i.e. the right part of the distribution extends to regions that would have negligible probability for a Poisson distribution; in other words, nodes with high degree exist with non-negligible probability. Two distributions that have been useful to fit real data are the *power-law with exponential cutoff* and the *stretched exponential*. Both forms take into account that in a finite network there must be a maximum finite degree. As an example, the following is an exponentially-truncated power-law:

$$P(k) \propto k^{-\gamma} \exp(-k/k_c), \tag{9.8}$$

where k_c is a "critical connectivity". When k approaches k_c the exponential term tends to 0 and $P(k)$ decreases faster than a power-law due to the exponential cutoff.

To conclude, we can say that the degree distribution function, together with the other statistics, are a kind of rough "signature" of the type of network and can be helpful in predicting the main aspects of the properties of the network. The reader should be aware that there are other measures beyond those described here, such as those that say which are the most "central" actors. Likewise, the picture has been one of static networks: their dynamical properties, have been neglected here. A good source for advanced material is [31].

9.2.4 Weighted Networks

Weighted networks are a useful extension of the network model. Weights $w(e)$ are assigned to edges $e \in E$ and could represent, for instance, the bandwidth of a communication line, the number of passengers transported on a given air route, the distance between two metro stations, and many other real-life aspects of networks. Here, weights represent transition probabilities among optima and their basins in fitness landscapes. We denote w_{ij} as the transition probability between local optima i and j, which in our model is generally different than the transition from j to i, denoted by w_{ji} (see Section 9.4.2 for more details).

Statistics for weighted networks are more or less straightforward extensions of those used for unweighted networks. Those that will be used in the rest of the chapter are briefly outlined below. The reader is refereed to [4] for more details.

Suitable distribution functions can also be defined for weighted networks. For example, it can be of interest to know the function $P(w_e)$ which indicates the frequency of weight w among the edges e in a given weighted network. Since our networks are directed, we use $P_{in}(w_e)$ and $P_{out}(w_e)$ which give the probability that any given edge e has incoming or outgoing weight w.

9.2.4.1 Strength

The vertex *strength*, s_i, is defined as $s_i = \sum_{j \in \mathscr{V}(i)-\{i\}} w_{ij}$, where the sum is over the set $\mathscr{V}(i) - \{i\}$ of neighbors of vertex i. This metric is, therefore, a generalization of the node's degree giving information about the number and importance of the edges.

9.2.4.2 Weighted Clustering Coefficient

The standard clustering coefficient (described in Section 9.2.3) does not consider weighted edges. We thus use the *weighted clustering* measure proposed by [4], which combines the topological information with the weight distribution of the network:

$$c^w(i) = \frac{1}{s_i(k_i-1)} \sum_{j,h} \frac{w_{ij}+w_{ih}}{2} a_{ij} a_{jh} a_{hi}$$

where s_i is the vertex strength, $s_i = \sum_{j \in \mathscr{V}(i)-\{i\}} w_{ij}$, $a_{nm} = 1$ if $w_{nm} > 0$, $a_{nm} = 0$ if $w_{nm} = 0$ and $k_i = \sum_{j \neq i} a_{ij}$.

This metric $c^w(i)$ counts, for each triple formed in the neighborhood of vertex i (indicated in the equation by $a_{ij}a_{jh}a_{hi}$), the weight of the two participating edges of vertex i. The normalization factor $s_i(k_i - 1)$, ensures that the metric is in the range $[0,1]$. It is customary to define C^w as the weighted clustering coefficient averaged over all vertices of the network.

9.2.4.3 Disparity

The *disparity* measure $Y_2(i)$, gauges the heterogeneity of the contributions of the edges of node i to the total weight (strength):

$$Y_2(i) = \sum_{j \neq i} \left(\frac{w_{ij}}{s_i} \right)^2$$

9.2.4.4 Shortest Paths

For weighted graphs, computing the shortest paths depends on the meaning of the edge weights. For example, if the weight represents e.g. a frequency of interaction, an electrical load, the number of passengers transported or a probability of transition, then the higher the weight, the "nearer" the two end points. In this case the path length between any two connected vertices is taken as the sum of the reciprocal of the weights $\sum 1/w_{ij}$ where the sum is over all edges $\{ij\}$ traversed along the path from the start node to the end node. On the other hand, if weights represent "costs" of some kind and the aim is to have low total cost, then the length is simply the sum of the costs of all edges along the path and the minimal length is that of the path with minimum cost.

In the local optima network model, we measure the shortest distance between two nodes as the expected number of operator moves to go from one node to the other. Given that the transition probability between two nodes i and j is given by w_{ij}, we calculate the distance between them as $d_{ij} = 1/w_{ij}$. The length of a path between two arbitrary connected nodes is, therefore, the sum of these distances along the edges connecting them. The average path length of the whole network is the average value of all the possible shortest paths.

9.2.5 Community Structure in Networks

A last theme in this section that we want to treat briefly is the "intermediate" structure of large networks since it will play a role in the following and is an important feature of complex networks. Model networks grown according to the Barabási–Albert recipe [2] or randomly generated have little structure in the sense that there are few or no recognizable sub-networks. That is, if one looks at a picture of the network it appears to be rather homogeneous on a global scale.

On the contrary, many observed networks, especially those arising from social interactions, show the presence of clusters of nodes. These clusters are called *communities*. It is difficult, if not impossible, to give a precise and unique mathematical definition of a community. An intuitive definition of a community is the following: nodes belonging to a community are more strongly associated with each other than they are with the rest of the network. In other words, the intra-community connectivity is higher than the inter-community connectivity. Of course, the definition is somewhat circular but in the last few years several algorithms have been proposed for community detection. Since this task is a hard computational problem, machine learning algorithms and heuristics have been used and in practice these work satisfactorily.

9.3 Example Combinatorial Landscapes

9.3.1 The NK Model

The idea of an *NK* landscape is to have N “spins“or “loci”, each with two possible values, 0 or 1. The model is a real stochastic function Φ defined on binary strings $s \in \{0,1\}^N$ of length N, $\Phi : s \rightarrow \mathbb{R}_+$. The value of K determines how many other gene values in the string influence a given gene s_i, $i = 1,\dots,N$. The value of Φ is the average of the contributions ϕ_i of all the loci:

$$\Phi(s) = \frac{1}{N}\sum_{i=1}^{N}\phi_i(s_i, s_{i_1}, \dots, s_{i_K})$$

By increasing the value of K from 0 to $N-1$, *NK* landscapes can be tuned from smooth to rugged. For $K = 0$ all contributions can be optimized independently which makes Φ a simple additive function with a single maximum. At the other extreme, when $K = N-1$, the landscape becomes completely random. The probability of any given configuration being the optimum is $1/(N+1)$, and the expected number of local optima is $2^N/(N+1)$. Intermediate values of K interpolate between these two extremes and have a variable degree of “epistasis”, i.e. of gene interaction [21, 22, 25].

The K variables that form the context of the fitness contribution of gene s_i can be chosen according to different models. The two most widely studied models are the *random neighborhood* model, where the K variables are chosen randomly according to a uniform distribution among the $N-1$ variables other than s_i, and the *adjacent neighborhood* model, in which the K variables are closest to s_i in a total ordering $s_1, s_2, \dots, s_N$ (using periodic boundaries). No significant differences between the two models were found in terms of global properties of the respective families of landscapes, such as mean number of local optima or autocorrelation length [21, 48]. Similarly, our preliminary studies on the characteristics of the *NK* landscape optima networks did not show noticeable differences between the two neighborhood models. Therefore, the study in this chapter considers the more general random model.

9.3.2 The Quadratic Assignment Problem

The quadratic assignment problem (QAP) is a combinatorial problem in which a set of facilities with given flows has to be assigned to a set of locations with given distances in such a way that the sum of the product of flows and distances is minimized. A solution to the QAP is generally written as a permutation π of the set $\{1,2,...,n\}$. The cost associated with a permutation π is given by:

$$C(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j}$$

where n denotes the number of facilities/locations and $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are referred to as the distance and flow matrices, respectively. The structure of these two matrices characterizes the class of instances of the QAP problem.

The results presented in this chapter are based on two instance generators proposed in [23]. These generators were originally devised for the multi-objective QAP, but were adapted for the single-objective QAP and used for the local optima network analysis in [9, 10]. In order to perform a statistical analysis of the extracted local optima networks, several problem instances of the two different problem classes were considered. The first generator produces uniformly random instances where all flows and distances are integers sampled from uniform distributions. This leads to the kind of problem known in literature as *Tai*nn*a*, where nn is the problem dimension [40]. The second generator produces flow entries that are non-uniform random values. This procedure, detailed in [23] generates random instances of type *Tai*nn*b* which have the so called “real-like” structure since they resemble the structure of QAP problems found in practical applications.

9.4 The Local Optima Network Model

This section formally describes the local optima network model of combinatorial landscapes. We start by defining the notion of fitness landscapes, and follow by formalizing the notions of nodes and edges of the network model.

A fitness landscape [36] is a triplet (S, V, f) where S is a set of potential solutions i.e. a search space; $V : S \longrightarrow 2^S$, a neighborhood structure, is a function that assigns to every $s \in S$ a set of neighbors $V(s)$, and $f : S \longrightarrow R$ is a fitness function that can be pictured as the *height* of the corresponding solutions.

Local optima networks have been analyzed for the two combinatorial landscapes discussed in Section 9.3. Therefore, two search spaces or solution representations have been studied: binary strings (NK landscapes) and permutations (QAP). For each case, the most basic neighborhood structure is considered, as described in Table 9.1. The single bit-flip operation changes a single bit in a given binary string, whereas the pairwise exchange operation exchanges any two positions in a permutation, thus transforming it into another permutation.

Table 9.1 Search space and neighborhood structure characteristics.

Representation	*Length*	*Search space size*	*Neighborhood*	*Neighborhood size*
Binary	N	2^N	single bit-flip	N
Permutation	N	$N!$	pairwise exchange	$N(N-1)/2$

9.4.1 Definition of Nodes

We start by describing the *HillClimbing* algorithm (Algorithm 4) used to determine the local optima, and therefore define the basins of attraction. The algorithm defines a mapping from the search space S to the set of locally optimal solutions S^*. Hill climbing algorithms differ in their so-called *pivot* or selection rule. In best-improvement local search, the entire neighborhood is explored and the best solution is returned, whereas in first-improvement, a neighbor is selected uniformly at random and is accepted if it improves on the current fitness value. We consider here a best-improvement local search heuristic (see Algorithm 4). For a comparison between first and best-improvement local optima network models, the reader is referred to [33].

This best-improvement local search (or hill-climbing) algorithm is used to determine the local optima. The neighborhoods used for each of the studied representation can be seen in Table 9.1. These local optima will represent the nodes of the network as discussed below.

9.4.1.1 Nodes

A local optimum (LO), which is taken to be a maximum here, is a solution s^* such that $\forall s \in V(s)$, $f(s) \leq f(s^*)$.

Let us denote by $h(s)$ the stochastic operator that associates each solution s to its local optimum, i.e. the solution obtained after applying the best-improvement hill-climbing algorithm (see Algorithm 4) until convergence. The size of the landscape is finite, so we can denote by LO_1, LO_2, $LO_3 \ldots, LO_p$, the local optima. These *LOs* are the vertices of the *local optima network*.

9.4.2 Definition of Edges

Two edge models have been considered: basin-transition and escape edges.

9.4.2.1 Basin-Transition Edges

The basin of attraction of a local optimum $LO_i \in S$ is the set $b_i = \{s \in S \mid h(s) = LO_i\}$. The size of the basin of attraction of a local optimum i is the cardinality

Algorithm 4. Best-improvement local search (hill-climbing).

```
Choose initial solution $s \in S$
repeat
    choose $s^{'} \in V(s)$, such that $f(s^{'}) = max_{x \in V(s)} f(x)$
    if $f(s) < f(s^{'})$ then
        $s \leftarrow s^{'}$
    end if
until $s$ is a Local optimum
```

of b_i, denoted $\sharp b_i$. Notice that for non-neutral[1] fitness landscapes, as are standard NK landscapes, the basins of attraction as defined above produce a partition of the configuration space S. Therefore, $S = \cup_{i \in S^*} b_i$ and $\forall i \in S \, \forall j \neq i$, $b_i \cap b_j = \emptyset$.

We can now define the weight of an edge that connects two feasible solutions in the fitness landscape.

For each pair of solutions s and $s^{'}$, $p(s \rightarrow s^{'})$ is the probability to pass from s to $s^{'}$ with the given neighborhood structure. These probabilities are given below for the two solution representations studied (see Table 9.1), with length N or size N and considering uniform selection of random neighbors.

Binary representation:
if $s^{'} \in V(s)$, $p(s \rightarrow s^{'}) = \frac{1}{N}$ and
if $s^{'} \notin V(s)$, $p(s \rightarrow s^{'}) = 0$.
Permutation representation: if $s^{'} \in V(s)$, $p(s \rightarrow s^{'}) = \frac{1}{N(N-1)/2}$ and
if $s^{'} \notin V(s)$, $p(s \rightarrow s^{'}) = 0$.

The probability ($p(s \rightarrow b_j) \leq 1$) to go from solution $s \in S$ to a solution belonging to the basin b_j, is:

$$p(s \rightarrow b_j) = \sum_{s^{'} \in b_j} p(s \rightarrow s^{'})$$

Thus, the total probability of going from basin b_i to basin b_j, i.e. the weight w_{ij} of edge e_{ij}, is the average over all $s \in b_i$ of the transition probabilities to solutions $s^{'} \in b_j$:

$$p(b_i \rightarrow b_j) = \frac{1}{\sharp b_i} \sum_{s \in b_i} p(s \rightarrow b_j)$$

9.4.2.2 Escape Edges

The escape edges are defined according to a distance function d (minimal number of moves between two solutions), and a positive integer $D > 0$. There is an edge e_{ij} between LO_i and LO_j if a solution s exists such that $d(s, LO_i) \leq D$ and s $h(s) = LO_j$. The weight w_{ij} of this edge is $w_{ij} = \sharp\{s \in S \mid d(s, LO_i) \leq D$ and $h(s) = LO_j\}$. This

[1] For a definition of basins that deals with neutrality, the reader is referred to [45].

weight can be normalized by the number of solutions, $\sharp\{s \in S \mid d(s, LO_i) \leq D\}$, within reach at distance D.

9.4.3 Local Optima Network

The weighted local optima network $G_w = (N, E)$ is the graph where the nodes $n_i \in N$ are the local optima, and there is an edge $e_{ij} \in E$, with weight w_{ij}, between two nodes n_i and n_j if $w_{ij} > 0$.

According to both definitions of edge weights, w_{ij} may be different than w_{ji}. Thus, two weights are needed in general, and we have an oriented transition graph.

Figures 9.2 and 9.3 illustrate the alternative local optima network (LON) models. All figures correspond to a real *NK* landscape with N = 18, K = 2. Figure 9.2 illustrates the basin-transition edges, while Figure 9.3 the escape edges with $D = 1$ (left) and $D = 2$ (right). Notice that the basin-transition edges (Figure 9.2) produce a densely connected network, while the escape edges (Figure 9.3) produce more sparse networks.

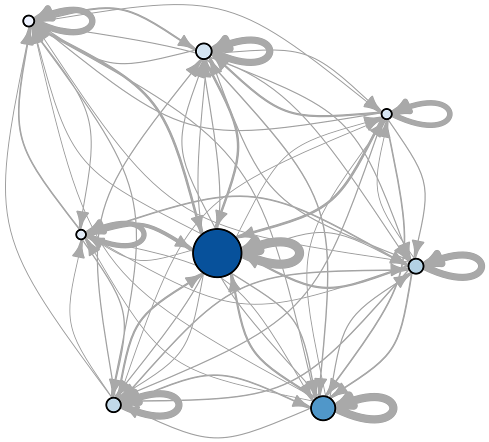
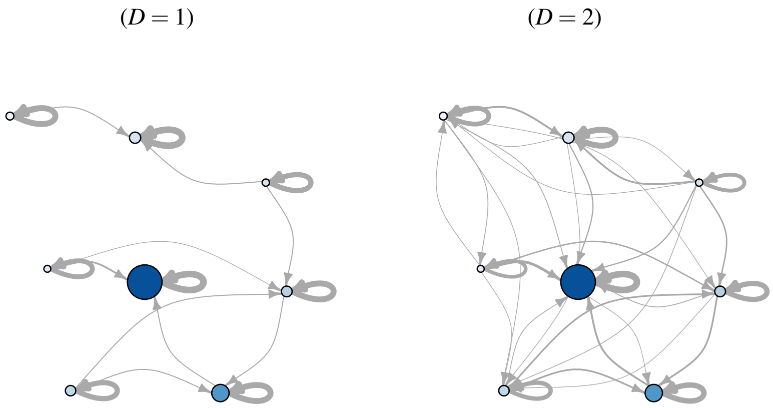


Fig. 9.2 Local optima network with basin-transition edges for an *NK*-landscape instance with N = 18, K = 2. The size of the nodes is proportional to the basin sizes. The nodes' color represent the fitness values: the darker the color, the highest the fitness value. The edges' width scales with the transition probability (weight) between local optima.



$(D = 1)$ $(D = 2)$

Fig. 9.3 Local optima network with escape edges (with $D = 1$ and $D = 2$) for an *NK*-landscape instance with N = 18, K = 2. Since this model does not require the calculation of the basins sizes, these are not depicted in the plots. The nodes' color represent the fitness values: the darker the color, the highest the fitness value. The edges' width scales with the transition probability (weight) between local optima.

9.5 Results of the Network Analysis

The purpose of this section is to give an overview of the main results of the analysis of local optima networks for the two example combinatorial landscapes: *NK* landscapes (Section 9.5.1) and the quadratic assignment problem (Section 9.5.2). For each example, the empirical set up and instances analyzed are discussed. The values obtained from the study of basins of attraction, general network metrics and connectivity, are reported and discussed.

9.5.1 The NK Model

For the *NK* model, the two definitions of edges, i.e. basin-transition and escape edges (Section 9.4.2), are considered. Moreover, an initial study correlating network metrics with search difficulty is also presented.

Results are presented for landscapes with $N = 18$ and varied values of K ($K \in \{2,4,6,8,10,12,14,16,17\}$.$N = 18$ represents the largest size for which an exhaustive sample of the configuration space was computationally feasible in our implementation. Metrics are generally calculated as averages of 30 independent instances for each K value.

9.5.1.1 Basins of Attraction

We start by analyzing the structure of the basins of attraction, namely, their size, shape and fitness of the corresponding local optima. These features are independent of the network edge definition. They are, however, relevant as the time complexity of local search heuristics is known to be linked to the size and spreading of attraction basins [16].

The distribution of basin sizes for given N and K values is not uniform; instead it follows a right-skewed distribution with a faster-than-exponential decay (see Figure 9.4, left, with semi-logarithmic scale). With increasing ruggedness (K values), the distribution shifts to the left and decays faster. This suggest that as the landscape ruggedness increases, the basin sizes decrease. In particular, with increasing ruggedness, the decrease of the relative size of global optimum basin is approximately exponential (Figure 9.4, Right).

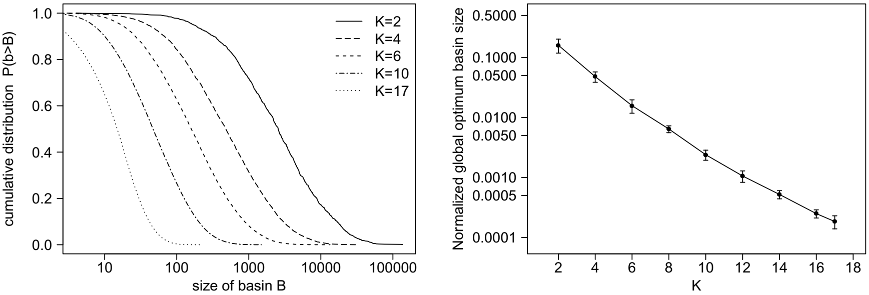


Fig. 9.4 Size of the basins of attraction for NK Landscapes. **Left**: cumulative distribution of basin sizes for landscapes with $N = 18$, and selected values of K. **Right**: average normalized size of the basin of the global optimum. Averages (points) and standard deviation (bars) refer to 30 instances for each K value.

With respect to the fitness of local optima and the size of their basins, a strong positive correlation was observed. Surprisingly, the average Spearman correlation coefficient is above 0.8 for all K values. Figure 9.5 (left) provides an example for $N = 18$, $K = 8$. This is an encouraging feature suggesting that local optima with high fitness should be easier to locate by hill-climbing. NK landscapes can thus be imagined as mountain ranges where wider mountain basins belong to higher peaks. But intuitions can be misleading, a striking finding is that these mountains are hollow; for all the observed instances, the average size of the basin interior is always less than 1% of the size of the basin itself. In other words, most solutions sit on the basin frontier and neighboring basins are richly interconnected [41].

9.5.1.2 General Network Features

Table 9.2 reports some general features for the two network models: basin-transition edges and escape edges (with distances ($D = \{1,2\}$); specifically, the number of

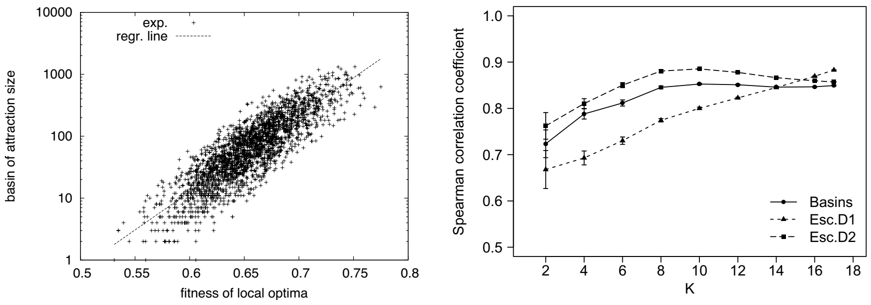


Fig. 9.5 Fitness correlations for *NK* landscapes. **Left**: Basin size and the fitness of its corresponding optima for a representative instance with $N = 18, K = 8$. **Right**: fitness of an optimum and its strength, i.e. the sum of the weights of its incoming transitions. Averages (points) and 0.95 confidence intervals (bars) are estimated by a t-test over 30 instances.

Table 9.2 General network features for *NK* landscapes. K = epistasis value of the corresponding *NK* landscape ($N = 18$); N_v = number of vertices; D_{edge} = density of edges ($N_e/(N_v)^2 \times 100\%$); L_{opt} = average shortest path to reach the global optimum ($d_{ij} = 1/w_{ij}$). Values are averages over 30 random instances, standard deviations are shown as subscripts.

K	N_v	D_{edge} (%)			L_{opt}		
	all	Basin-trans.	Esc.D1	Esc.D2	Basin-trans.	Esc.D1	Esc.D2
2	$43.0_{27.7}$	$74.182_{13.128}$	$8.298_{4.716}$	$22.750_{9.301}$	$21.2_{8.0}$	$16.8_{4.7}$	$33.5_{14.1}$
4	$220.6_{39.1}$	$54.061_{4.413}$	$1.463_{0.231}$	$7.066_{0.810}$	$41.7_{10.5}$	$19.2_{5.1}$	$53.7_{12.4}$
6	$748.4_{70.2}$	$26.343_{1.963}$	$0.469_{0.047}$	$3.466_{0.279}$	$80.0_{19.1}$	$22.2_{3.9}$	$66.7_{12.9}$
8	$1668.8_{73.5}$	$12.709_{0.512}$	$0.228_{0.009}$	$2.201_{0.066}$	$110.1_{13.8}$	$24.0_{4.9}$	$76.6_{9.1}$
10	$3147.6_{109.9}$	$6.269_{0.244}$	$0.132_{0.004}$	$1.531_{0.036}$	$152.8_{19.3}$	$27.3_{5.0}$	$90.7_{8.4}$
12	$5270.3_{103.9}$	$3.240_{0.079}$	$0.088_{0.001}$	$1.115_{0.015}$	$185.1_{23.8}$	$30.3_{6.7}$	$108.3_{12.3}$
14	$8099.6_{121.1}$	$1.774_{0.035}$	$0.064_{0.001}$	$0.838_{0.009}$	$200.2_{16.0}$	$38.9_{9.6}$	$124.7_{8.6}$
16	$11688.1_{101.3}$	$1.030_{0.013}$	$0.051_{0.000}$	$0.647_{0.004}$	$211.8_{15.0}$	$47.9_{11.4}$	$146.2_{11.2}$
17	$13801.0_{74.1}$	$0.801_{0.007}$	$0.047_{0.000}$	$0.574_{0.002}$	$214.3_{17.5}$	$55.7_{12.5}$	$155.9_{12.2}$

nodes (which is independent of the edges model), the relative number or density of edges, and the average path length to the global optimum, where the distance between two nodes i and j is given by $1/w_{ij}$.

The number of local optima (Table 9.2, 2^{nd} column) rapidly increases with the value of K (1^{st} column). Escape edges produce much less dense networks (3^{rd}, 4^{th}, and 5^{th} columns), which confirms the visual inspection of Figures 9.2 and 9.3. For all the models, the density of edges decreases, whereas the path length to the global optimum (6^{th},7^{th}, and 8^{th} columns) increases with increasing values of K. Since a low density of edges and a long path length to the optimum would hinder heuristic search, these observations confirm that the network metrics capture the search difficulty associated with increasing landscape ruggedness. These findings also suggest that the two models of edges are consistent, which is encouraging as calculating the escape-edges is less computationally expensive.

A study of the network's local connectivity shows differences between the two edge models. As Figure 9.6 (left) shows, the basin-transition edges produce networks with higher out degree (i.e. number of transitions leaving a node). Clustering coefficients are also higher in this case (they are indeed higher than those of a random graph), which is probably due to the higher density of basin-transition edges (Figure 9.6, right). There is, however, a common decreasing trend for all models in this metric with increasing K, as seen in Figure 9.6, right. The varying difference between the two models might lie in the size of the basins of attraction. For low K values, large basins produce high edge density and thus high clustering coefficients in the basin-transition model; whereas for large values of K, basins are so small that the two models show a similar structure. The escape-edges reproduce the basin topology.

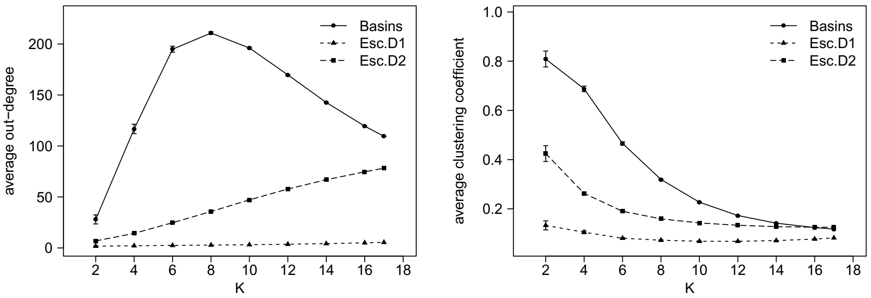


Fig. 9.6 Local connectivity for NK landscapes. **Left**: Average out-degree. **Right**: Average clustering coefficient. Averages (points) and 0.95 confidence intervals (bars) are estimated by a t-test over 30 random instances with $N = 18$.

9.5.1.3 Transitions among Local Optima

Edges weights w_{ij} can be interpreted as the expected number of moves it takes to go from basin b_i to basin b_j (or from local optimum i to basin b_j in the escape-edge model). For both edge models, the weights of self-loops (w_{ii}) are an order of magnitude higher than $w_{ij,j \neq i}$. Therefore, it is more probable for a random move to remain in the same basin than to escape from it. Self-loop probabilities are then correlated with basin sizes, and display a similar exponential decrease with the landscape ruggedness K. We analyze, therefore, in more detail the weights $w_{ij,j \neq i}$. Figure 9.7 (left) shows the cumulative distribution of basin-transition weights for $w_{ij,j \neq i}$ for selected values of K. The curves illustrate that low K values have longer tails, whereas mid and high K values produce a faster decay. Figure 9.7 (right), shows the average weight out-going transition for all edge models and K values. For the escape-edges model, the out-going weights decrease smoothly, with a slower decrease for $D = 1$. The trend is different for basin-transition edges where the out-going weights

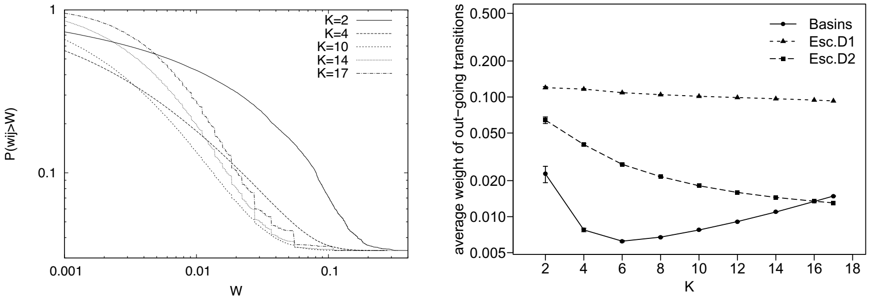


Fig. 9.7 Network transitions for *NK* landscapes. **Left**: Cumulative probability distribution of the network weights $w_{ij,j\neq i}$ with basin-transition edges, $N = 18$, and selected K values. **Right**: Average out-going w_{ij} values for all edge models and K values. Averages (points) and 0.95 confidence intervals (bars) are estimated by a t-test over 30 landscapes with $N = 18$.

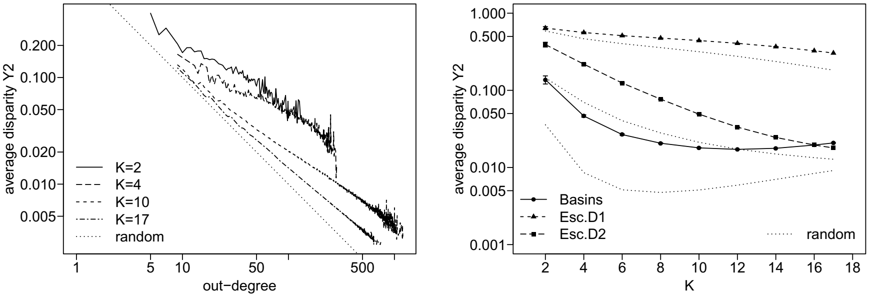


Fig. 9.8 Weight disparity for outgoing edges in *NK* landscapes. **Left**: relationship between disparity and out-degree for the basin-transition edges, and selected values of K. **Right**: relationship between disparity and landscape ruggedness (K) for the two network models. Averages and confidence intervals are estimated on the 30 analyzed instances. Dotted lines (labeled as random) present the limit case $Y_2 = 1/k$, where k is the node out-going degree.

decrease with increasing ruggedness but only up to $K = 6$, and then they increase in value.

A relevant question is whether there are preferential directions when leaving a particular node in the network. Specifically, whether for a given optimum i, all the outgoing weights $w_{ij,j\neq i}$, are equivalent. This can be revealed by the disparity Y_2 metric (discussed in Section 9.2.4), which gauges the heterogeneity of the contributions of the edges of node i to its total strength s_i. If a dominant weight does not exist, the value $Y_2 \approx 1/k$, were k is the node out-degree. Figure 9.8 (left) illustrates the relationship between disparity and out-degree for basin-transition edges and selected K values. The figure also shows the limit case $Y_2 \approx 1/k$ (labeled as random). For calculating this plot, the nodes' disparity values $Y_2(i)$ were grouped and averaged by node out-degree. The curves suggest that there are preferential directions for low values of K. However, with increasing K, the transition probabilities to leave a given basin appear to become more uniform (i.e closer to the limit case $Y_2 = 1/k$). Figure 9.8 (right) shows the disparity metric for all models and K values. In all cases, disparity values are higher than those expected in the limit case ($Y_2 \approx 1/k$, labeled as random), indicating that preferential transitions are present. For the basin-transition edges and the escape edges wit $D = 2$, disparity values get closer to the limit case of large K values.

9.5.1.4 Search Difficulty and Network Metrics

While the previous sections described relevant network features, this section explores correlations between these features and the performance of local search heuristics running on the underlying combinatorial optimization problem. The ultimate goal is to have predictive models of the performance of specific search heuristics when solving a given problem instance, and thus select a method according to this predication.

Daolio et al. [11], conducted a first study using *iterated local search* and the *NK* family of landscapes (with escape edges, $D = 2$). Iterated local search is a relatively simple but powerful single point heuristic. It alternates between a perturbation stage and an improvement stage. This search principle has been rediscovered multiple times, within different research communities and with different names [5]. The term *iterated local search* (ILS) was proposed in [26]. Algorithm 5 outlines the procedure.

Algorithm 5. Iterated Local Search

```
$s_0 \leftarrow$ GenerateInitialSolution
$s^* \leftarrow$ LocalSearch($s_0$)
repeat
    $s' \leftarrow$ Perturbation($s^*$)
    $s'^* \leftarrow$ LocalSearch($s'$)
    $s^* \leftarrow$ AcceptanceCriterion($s^*, s'^*$)
until termination condition met
```

In our implementation, the *LocalSearch* stage corresponds to the best-improvement hill-climber described in Section 9.4 (Algorithm 4), which stops when reaching a local optimum, and uses the single bit-flip move operator. The *Perturbation* stage considers a stronger operator,i.e. 2-bit-flip mutation. A simple greedy acceptance is used (i.e. only improvement moves are accepted). The search terminates at the global optimum, which for benchmark problems is known *a priori*, or when reaching a pre-set limit of fitness evaluations FE_{max}.

As the performance criterion, we selected the expected number of function evaluations to reach the global optimum (*success*) after independent restarts. This measure accounts for both the success rate ($p_s \in (0,1]$) and the convergence speed [1]. The function evaluations limit was set to $1/5$ of the size of the search space, i.e. $FE_{max} \simeq 5.2 \cdot 10^4$, for binary strings of length $N = 18$. The success rate p_s and running time of successful runs T_s were estimated on 500 random restarts per instance.

Figure 9.9 (left), shows the distribution of the expected iterated local search (ILS, Algorithm 5) running times to success with respect to K. As expected, the running times increase steadily with increasing K. As an example of the correlations arising between LON features and the performance of ILS, the right plot in Figure. 9.9, illustrates the relationship between the running time and the shortest path length to the global optimum. A strong positive correlation is observed, suggesting that LON features are able to capture search difficulty in combinatorial landscapes. Other network metrics also revealed positive correlations with search performance, namely, the average out-degree, the average disparity, and the degree of assortativity [11]. A multiple regression analysis was also conducted using the most significant network metrics. The model obtained was able to predict about 85% of the variance observed in the expected running times

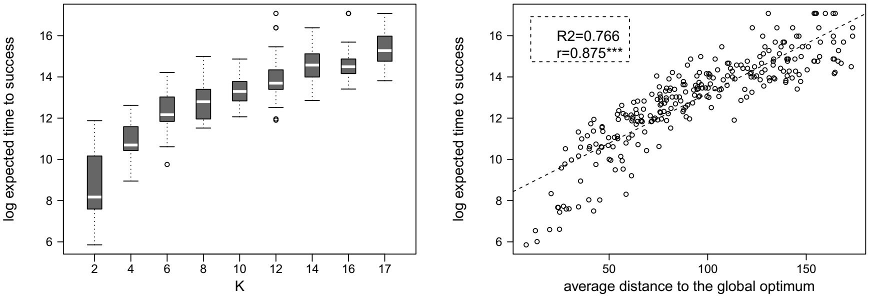


Fig. 9.9 Performance of ILS on *NK* landscapes. **Left**: distribution of the expected running times to success for different *K* values. **Right**: correlation between the expected running times and the average shortest path to the global optimum. The regression line is dashed. The legend gives the ratio of variance explained by the regression, *R*2, and the Pearson correlation coefficient, *r*, with the asterisks indicating its significance level.

9.5.2 The Quadratic Assignment Problem

This section summarises the main results for the QAP. In this case, only the basin-transition edges are considered. The study of escape-edges will be the subject of future work. An analysis of the LON communities structure is also presented. This was not done for the *NK* landscape as our analysis revealed little cluster structure of local optima in these more random landscapes. Their search spaces seem isotropic from the point of view of basin inter-connectivity. An initial study correlating QAP local optima network metrics with heuristic search performance is reported elsewhere [7].

Two QAP instance classes were considered: real-like and uniform instances as described in Section 9.3.2. For the general network analysis, 30 random uniform and 30 random real-like instances have been generated for each problem dimension in $\{5,...,10\}$, and metrics are given as averages of these 30 independent instances. To the specific purpose of community detection, 200 additional instances have been produced and analyzed with size 9 for the random uniform class, and size 11 for the real-like instances class. Problem size 11 is the largest permitting an exhaustive sample of the configuration space. Only the basin-transition edges are studied. The escape-edges will be the subject of future work.

9.5.2.1 Basins of Attraction

Figure 9.10 (left) shows the size of the global optimum basin of attraction (normalized by the whole search space size). This value decreases exponentially with the problem size for both instance classes. The real-like instances have larger global optimum basins, which can be explained by their smaller local optima networks (as discussed below). The relative size of the global optimum basin is related to the probability of finding the best solution with a local search algorithm from a random starting point. The exponential decrease confirms that the larger the problem, the smaller the probability for a local search algorithm to locate the global optimum. The separation between the curves in Figure 9.10 (left) is consistent with recent empirical results indicating that real-like instances are easier to solve than uniform instances for heuristic search algorithms such as simulated annealing and genetic algorithms [7].

Figure 9.10 (right), shows the correlation between the fitness value of a local optimum and the size of its basin of attraction. As with the *NK* landscape, there is a strong positive correlation between the fitness of a local optimum and the size of its basin. This is an encouraging feature suggesting that local optima with high fitness should be easier to locate by stochastic local search. The correlation coefficients are generally higher for the uniform instances, which also show noticeably lower variance. More details about QAP basins of attraction and network features can be found in [10].

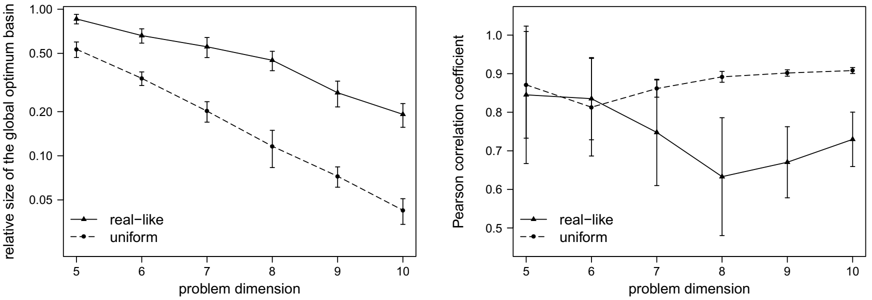


Fig. 9.10 Basins of attraction in QAP instances. **Left**: Normalized size of the basin of the global optimum. **Right**: Pearson correlation coefficient between the fitness value of an optimum and the logarithmic size of its basin. Averages (points) and 0.95 confidence intervals (bars) are estimated with a t-test over 30 instances for each combination of problem class and size.

9.5.2.2 General Network Features

Table 9.3 reports relevant features for the two classes of QAP instances and problem sizes from 5 to 10; specifically, the number of vertices (N_v), the density of edges ($D_{edge} = N_e/(N_v)^2$), the weighted clustering coefficient (C^w) and the disparity in out-going transitions (Y_2). The number of local optima grows exponentially with the problem dimension for both instance classes. For a given problem size, however, real-like instances produce much smaller networks, i.e. they have significantly fewer local optima. The size difference between the two instance classes also grows almost exponentially with the problem dimension. This is again consistent with the empirical studies indicating that real-like instances are easier to solve by common metaheuristics [7].

The QAP networks are notably dense, with density of edges close to one (Table 9.3, D_{edge}), much more dense than than those of the NK landscapes which operate on binary spaces. This is not surprising as the neighborhood size is larger for permutation search spaces (see Table 9.1). Local optima networks are almost complete graphs for QAP. Moreover, the average weighted clustering coefficient C^w is higher than what would be expected from network density alone, indicating that the interconnected triples are more likely formed by edges with larger weight. The studied QAP instances show very high local connectivity. The clustering coefficient decreases with the problem dimension and is higher for real-like instances.

The last row in Table 9.3 reports the disparity coefficient in out-going transitions for both classes with respect to the problem dimension. High diversity indicates preferential transitions. The decreasing trend reflects that, with increasing problem size, the out-going transition to neighbouring optima tend to become equally probable. This trend is more evident for uniform instances whose LONs have higher cardinality.

Table 9.3 General network features for QAP instances. N_v = number of vertices; D_{edge} = density of edges $(N_e/(N_v)^2)$; C^w = weighted clustering coefficient; Y_2 = disparity in out-going transitions. Values are averages over 30 instances with standard deviations in subscripts.

	class	size					
		5	6	7	8	9	10
N_v	real-like	$1.667_{0.802}$	$2.767_{1.48}$	$3.900_{2.25}$	$6.133_{2.99}$	$12.567_{5.73}$	$25.700_{13.8}$
	uniform	$3.333_{1.27}$	$6.800_{2.37}$	$19.100_{7.39}$	$51.300_{20.53}$	$137.300_{54.84}$	$414.133_{177.5}$
D_{edge}	real-like	$1.000_{0.000}$	$0.993_{0.026}$	$0.994_{0.030}$	$0.999_{0.006}$	$0.992_{0.025}$	$0.988_{0.035}$
	uniform	$0.998_{0.007}$	$0.993_{0.019}$	$0.969_{0.030}$	$0.940_{0.036}$	$0.909_{0.035}$	$0.874_{0.053}$
C^w	real-like	$1.000_{0.000}$	$0.988_{0.032}$	$0.995_{0.024}$	$0.999_{0.005}$	$0.995_{0.015}$	$0.993_{0.020}$
	uniform	$0.998_{0.008}$	$0.995_{0.014}$	$0.982_{0.016}$	$0.970_{0.017}$	$0.961_{0.015}$	$0.952_{0.020}$
Y_2	real-like	$0.888_{0.193}$	$0.739_{0.251}$	$0.533_{0.270}$	$0.369_{0.171}$	$0.221_{0.118}$	$0.143_{0.058}$
	uniform	$0.649_{0.271}$	$0.286_{0.093}$	$0.136_{0.061}$	$0.074_{0.048}$	$0.040_{0.013}$	$0.023_{0.008}$

9.5.2.3 Path Lengths

Figure 9.11 (left) displays the average shortest path length between optima and the average shortest path length to the global optimum. Both metrics clearly increase with problem size. Values are noticeably higher for the uniform instances, which have a larger number of local optima than the real-like instances for the same problem dimension. The figures support that the search difficulty increases with the problem size and the number of local optima.

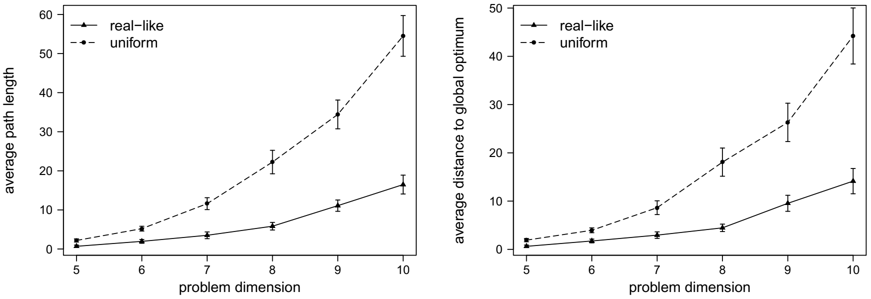


Fig. 9.11 Shortest paths in QAP instances. **Left**: average path length. **Right**: average length of the shortest paths to the global optimum. Averages (points) and 0.95 confidence intervals (bars) are estimated with a t-test over 30 instances for each combination of problem class and size.

9.5.2.4 Clustering of Local Optima

The manner in which local optima are distributed in the configuration space is relevant for heuristic search. Several questions can be raised. Are they uniformly distributed, or do they cluster in some non-homogeneous way? If the latter, what is the relation between objective function values within and among different clusters and how easy is it to go from one to another? As discussed in Section 9.2.5, clusters or communities in networks are groups of nodes that are strongly connected between them and poorly connected with the rest of the graph. The topological distribution of local optima can be directly investigated by means of community detection on the local optima network. In [9], we conducted a community detection study on the two classes of QAP instances. Problems of size of 11 for the real-like class and 9 for the uniform class were considered as LONs for these two cases have comparable sizes in terms of number of vertices.

Community detection is a difficult task, but today several good approximate algorithms are available [15]. In [9], we used two of them: (i) a method based on greedy modularity optimization, and (ii) a spin glass ground state-based algorithm in order to double check the community partition results. Figure 9.12 shows the modularity score (Q) distribution for each algorithm/instance-class. The higher the Q score of a partition, the crisper the community structure [15]. The plot indicates that the two instance classes are well separated in terms of Q, regardless of the algorithm used.

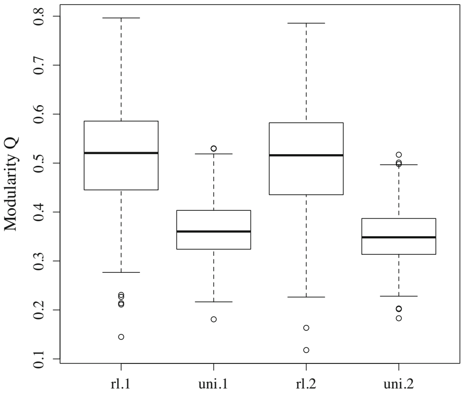


Fig. 9.12 Network modularity for QAP instances. Boxplots of the modularity score Q on the y-axis with respect to class problem (rl stands for real-like and uni stands for random uniform) and community detection algorithm (1 stands for fast greedy modularity optimization and 2 stands for spin glass search algorithm)

The real-like instances have significantly more cluster structure than the uniform instances. This can be appreciated visually in Figure 9.13 illustrating the community structures of two particular instances. These two selected cases have the highest Q values of their respective classes, but they represent a general trend. For the real-like instance (Figure 9.13, right) the groups of local optima are recognizable and form well separated clusters (encircled with dotted lines), which is also reflected in the high corresponding modularity value $Q = 0.79$. In contrast, the LON of the uniform instance (Figure 9.13, left) has some modularity, with a $Q = 0.53$, but the communities are hard to represent graphically, and thus are not shown in the picture.

The LONs community structure is likely to have consequences on the heuristic algorithms used to search the corresponding landscapes. According to the level of modularity, different search strategies would be more efficient. For example, we can envision that for real-like instances, a local search algorithm may require stronger perturbation mechanisms to escape a cluster of local optima with poor quality solutions.

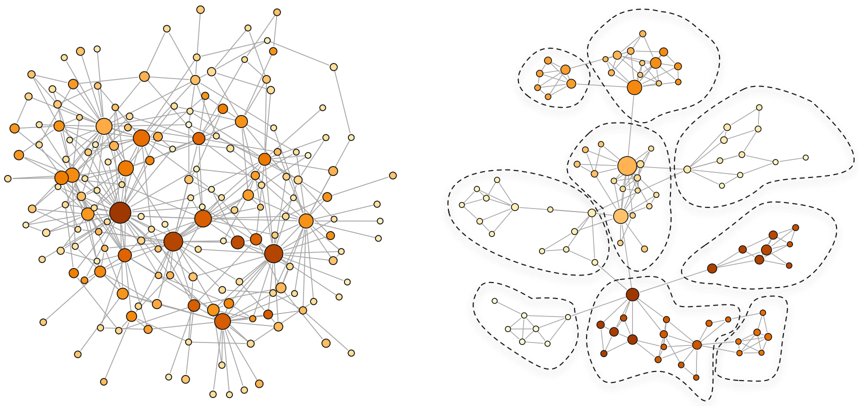


Fig. 9.13 Visualization of network communities in QAP instances. **Left**: a uniform instance of size 9. **Right**: a real-like instance of size. Node size is proportional to basin size, and node color to fitness (the darker, the better). Communities are highlighted in the right picture, which was not possible in the left one.

9.6 Conclusions

A network-based model of combinatorial landscapes is described and a thorough analysis is presented for two example combinatorial landscapes: *NK* landscapes and the quadratic assignment problem. A network model requires defining its nodes and edges. Nodes are the landscape local optima obtained with a best-improvement hill-climbing algorithm; edges are defined in two alternative ways: one is based on the transition probabilities between basins of attraction, the other on the transition (escape) probabilities from local optima. The model, therefore, compresses the fitness landscapes into a more manageable mathematical object. New features can be measured in this model, coming from the science of complex networks such as the degree distribution, clustering coefficient, shortest path length, disparity and community structure. Results from the studied landscapes show that local optima networks share some features with complex networks: basin sizes are not uniformly distributed, weight distributions are not normal, path lengths to the global optimum can be short, clustering coefficients can be high, and networks can have community structure.

The results clearly show that the search difficulty on the studied landscapes, which may be either known a priori or empirically estimated, correlate with some fundamental LON features such as the number of nodes, size of basins, shortest path length to the global optimum, out-degree, disparity, and degree of assortativity. Indeed, some of these metrics were used successfully to construct a statistical predictive model of search performance. The network analysis also revealed interesting topological differences on the distribution of local optima for different classes of problem instances. These differences may lead to designing search heuristics that can adapt and thus exploit the landscape structure.

Local optima networks can be seen as a generic model of combinatorial landscapes based on defining convergence points of simple heuristics in the search space. In this work, the convergence points are local optima, and edges are transition probabilities between these points which also reflect the path of a local search. From a mathematical point of view, LONs reduce the study of the whole transition matrix between solutions of a local search by a smaller transition matrix between local optima. The search dynamic is then decomposed into two time scales: one to reach local optima, the other to traverse between local optima. From LON graphs and their corresponding transition matrices, it should be possible to conduct a Markov chain analysis and thus compute running times or expected performance.

Another research direction is to use LONs for automated parameter tuning and design of heuristic search methods. Some network metrics can be estimated without knowing the global optimum beforehand. These metrics coupled with adequate performance prediction models open up exciting possibilities. Our current analysis requires the exhaustive enumeration of the search space; with standard sampling methods, larger search spaces could be studied. We plan to continue working on the afore mentioned directions and extend this analysis to other combinatorial optimization problems.

References

[1] Auger, A., Hansen, N.: Performance evaluation of an advanced local search evolutionary algorithm. In: Corne, D. (ed.) Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, vol. 2, pp. 1777–1784. IEEE (2005)

[2] Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)

[3] Barnett, L.: Ruggedness and neutrality - the NKp family of fitness landscapes. In: Adami, C., Belew, R.K., Kitano, H., Taylor, C. (eds.) Proceedings of the Sixth International Conference on Artificial Life, ALIFE VI, pp. 18–27. MIT Press (1998)

[4] Barthélemy, M., Barrat, A., Pastor-Satorras, R., Vespignani, A.: Characterization and modeling of weighted networks. Physica A 346, 34–43 (2005)

[5] Battiti, R., Brunato, M., Mascia, F.: Reactive Search and Intelligent Optimization. Operations Research/Computer Science Interfaces Series, vol. 45. Springer (2009)

[6] Boese, K.D., Kahng, A.B., Muddu, S.: A new adaptive multi-start technique for combinatorial global optimizations. Operations Research Letters 16, 101–113 (1994)

[7] Chicano, F., Daolio, F., Ochoa, G., Vérel, S., Tomassini, M., Alba, E.: Local optima networks, landscape autocorrelation and heuristic search performance. In: Coello Coello, C.A., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part II. LNCS, vol. 7492, pp. 337–347. Springer, Heidelberg (2012)

[8] Clauset, A., Shalizi, C., Newman, M.: Power-law distributions in empirical data. SIAM Review 51(4), 661–703 (2009)

[9] Daolio, F., Tomassini, M., Verel, S., Ochoa, G.: Communities of minima in local optima networks of combinatorial spaces. Physica A: Statistical Mechanics and its Applications 390(9), 1684–1694 (2011)

[10] Daolio, F., Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of the quadratic assignment problem. In: Fogel, G., et al. (eds.) Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, pp. 1–8. IEEE Press (2010)

[11] Daolio, F., Verel, S., Ochoa, G., Tomassini, M.: Local optima networks and the performance of iterated local search. In: Soule, T., Moore, J.H. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2012, pp. 369–376. ACM (2012)

[12] Dorogotsev, S.N.: Lectures on Complex Networks. Oxford University Press, Oxford (2010)

[13] Doye, J.P.K.: The network topology of a potential energy landscape: a static scale-free network. Physical Review Letter 88, 238701 (2002)

[14] Doye, J.P.K., Massen, C.P.: Characterizing the network topology of the energy landscapes of atomic clusters. Journal of Chemical Physics 122, 84105 (2005)

[15] Fortunato, S.: Community detection in graphs. Physics Reports 486, 75–174 (2010)

[16] Garnier, J., Kallel, L.: Efficiency of local search with multiple local optima. SIAM Journal on Discrete Mathematics 15(1), 122–141 (2001)

[17] Gfeller, D., De Los Rios, P., Caflisch, A., Rao, F.: Complex network analysis of free-energy landscapes. Proc. Nat. Acad. Sci. USA 104(6), 1817–1822 (2007)

[18] Hains, D., Whitley, L.D., Howe, A.E.: Revisiting the big valley search space structure in the TSP. Journal of the Operational Research Society 62(2), 305–312 (2011)

[19] Jones, T.: Evolutionary Algorithms, Fitness Landscapes and Search. PhD thesis, University of New Mexico, Albuquerque (1995)

[20] Kauffman, S., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. Journal of Theoretical Biology 128, 11–45 (1987)

[21] Kauffman, S.A.: The Origins of Order. Oxford University Press, New York (1993)

[22] Kaul, H., Jacobson, S.H.: New global optima results for the Kauffman *NK* model: handling dependency. Mathematical Programming, Ser. B 108(2-3), 475–494 (2006)

[23] Knowles, J.D., Corne, D.W.: Instance generators and test suites for the multiobjective quadratic assignment problem. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 295–310. Springer, Heidelberg (2003)

[24] Koopmans, T.C., Beckmann, M.: Assignment problems and the location of economic activities. Econometrica 25(1), 53–76 (1957)

[25] Limic, V., Pemantle, R.: More rigorous results on the Kauffman-Levin model of evolution. Annals of Probability 32, 2149 (2004)

[26] Lourenço, H.R., Martin, O., Stützle, T.: Iterated local search. In: Glover, F., Kochenberger, G.A. (eds.) International Series in Operations Research & Management Science, vol. 57, pp. 321–353. Kluwer Academic Publishers (2002)

[27] Massen, C.P., Doye, J.P.K.: Identifying communities within energy landscapes. Physical Review E 71, 046101 (2005)

[28] Merz, P., Freisleben, B.: Memetic algorithms and the fitness landscape of the graph bipartitioning problem. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN V 1998. LNCS, vol. 1498, pp. 765–774. Springer, Heidelberg (1998)

[29] Newman, M., Engelhardt, R.: Effect of neutral selection on the evolution of molecular species. Proc. Royal Society London B 256, 1333–1338 (1998)

[30] Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)

[31] Newman, M.E.J.: Networks: An Introduction. Oxford University Press, Oxford (2010)

[32] Ochoa, G., Tomassini, M., Verel, S., Darabos, C.: A study of NK landscapes' basins and local optima networks. In: Ryan, C., Keijzer, M. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008, pp. 555–562. ACM (2008)

[33] Ochoa, G., Verel, S., Tomassini, M.: First-improvement vs. best-improvement local optima networks of nk landscapes. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI, Part I. LNCS, vol. 6238, pp. 104–113. Springer, Heidelberg (2010)

[34] Rao, F., Caflisch, A.: The network topology of a potential energy landscape: a static scale-free network. Journal of Molecular Biology 342, 299–306 (2004)

[35] Reeves, C.R.: Landscapes, operators and heuristic search. Annals of Operations Research 86, 473–490 (1999)

[36] Reidys, C.M., Stadler, P.F.: Combinatorial landscapes. SIAM Review 44(1), 3–54 (2002)

[37] Rosé, H., Ebeling, W., Asselmeyer, T.: The density of states - a measure of the difficulty of optimisation problems. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN IV 1996. LNCS, vol. 1141, pp. 208–217. Springer, Heidelberg (1996)

[38] Sahni, S., Gonzalez, T.: P-complete approximation problems. Journal of the ACM (JACM) 23(3), 555–565 (1976)

[39] Stillinger, F.H.: A topographic view of supercooled liquids and glass formation. Science 267, 1935–1939 (1995)

[40] Taillard, É.D.: Comparison of iterative searches for the quadratic assignment problem. Location Science 3(2), 87–105 (1995)

[41] Tomassini, M., Verel, S., Ochoa, G.: Complex-network analysis of combinatorial spaces: The NK landscape case. Physical Review E 78(6), 066114 (2008)

[42] Vanneschi, L., Tomassini, M., Collard, P., Vérel, S.: Negative slope coefficient a measure to characterize genetic programming fitness landscapes. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 178–189. Springer, Heidelberg (2006)

[43] Vérel, S., Daolio, F., Ochoa, G., Tomassini, M.: Local optima networks with escape edges. In: Hao, J.-K., Legrand, P., Collet, P., Monmarché, N., Lutton, E., Schoenauer, M. (eds.) EA 2011. LNCS, vol. 7401, pp. 49–60. Springer, Heidelberg (2012)

[44] Verel, S., Ochoa, G., Tomassini, M.: The connectivity of NK landscapes' basins: a network analysis. In: Bullock, S., et al. (eds.) Proceedings of the Eleventh International Conference on Artificial Life, ALIFE XI, pp. 648–655. MIT Press, Cambridge (2008)

[45] Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of NK landscapes with neutrality. IEEE Transactions on Evolutionary Computation 15(6), 783–797 (2011)

[46] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)

[47] Weinberger, E.D.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cybernetics 63, 325–336 (1990)

[48] Weinberger, E.D.: Local properties of Kauffman's NK model, a tuneably rugged energy landscape. Phys. Rev. A 44, 6399–6413 (1991)

[49] Whitley, D., Hains, D., Howe, A.E.: A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 566–575. Springer, Heidelberg (2010)

[50] Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones, D.F. (ed.) Proceedings of the Sixth International Congress on Genetics, vol. 1, pp. 356–366 (1932)

Part III
Coevolution and Dynamics

Chapter 10
Fitness Landscapes That Depend on Time

Hendrik Richter

Abstract. Landscapes whose fitness values change with time occur in several contexts. A first is that the evolutionary process takes place in a dynamic environment. Dynamics may be connected to optimization problems with changing objective functions, or generally that conditions apart from the genetic makeup of the population, but massively influencing the evolutionary outcome, are not constant. Mathematically, such dynamic fitness landscapes can be described either by static landscapes that are externally driven to change with time, or by spatially extended dynamical systems which internally and simultaneously define topology and dynamics of the landscape. Another setting for time–dependent fitness are coevolutionary processes where the fitness of a given individual depends on the fitness and the genotype of other individuals in a temporal or spatial fashion. This is known to create coupled, interactive, tunable or deformable landscapes. Such coevolutionary processes induce time–dependence that is population–based and produce landscapes that are codynamic. In this chapter we intend to give an unified overview about issues in and problems of time–dependent fitness landscapes and particularly highlight several types of mathematical descriptions and their properties, similarities and differences.

10.1 Introduction

For understanding evolutionary dynamics, it is useful to have a notion of how the individuals' movements are related to, are partly controlled by, partly directed to and partly influenced by possible paths of increasing or decreasing fitness. One way to obtain such a notion is by fitness landscapes. These landscapes are an influential and important concept in evolutionary biology and evolutionary computation, and recent

Hendrik Richter
HTWK Leipzig University of Applied Sciences,
Faculty of Electrical Engineering and Information Technology, D–04251 Leipzig, Germany
e-mail: richter@eit.htwk-leipzig.de

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_10, © Springer-Verlag Berlin Heidelberg 2014

advances in their understanding are the topic of this book. A conventional view on fitness landscapes is that the fitness of a given point of the landscape is not changing its value for the time frame of consideration. A time frame of consideration may be a run of an evolutionary search algorithm, or an interval of evolutionary development to be observed and studied, or an analysis of the fitness landscape, for instance by a random walk on the landscape. Thus, such kind of fitness landscapes are essentially a static concept. However, if looking at the biological reality on the one hand, or the true nature of optimization problems on the other, we notice that the static view is a significant restriction in understanding interesting phenomena. In biology, there are usually several species and several individuals. They can compete or cooperate with the aim to maximize the share they are getting from limited resources. The outcome of competition or cooperation (or mixing both in an overall strategy) definitely influences survival and reproduction processes, and hence fitness in a timely fashion. Therefore, fitness of a given phenotype is affected by the actions (and therefore by the fitness) of other phenotypes for any interesting time interval. Moreover, environmental conditions are changing frequently and hence bringing timely changes to the fitness of a given phenotypic makeup. The same is true if we consider to solve optimization problems using methods of evolutionary computation. Here the objective function of a multitude of real–world problems is frequently affected by timely changes, for instance in dynamic vehicle [4, 73] and network [12, 13] routing, or job scheduling [35].

In recent years we have seen various attempts to describe, model and understand the phenomena connected to fitness that changes over time [9, 33, 52, 74]. In evolutionary computation this was particularly driven by attempts to design algorithms that perform and behave well in such dynamic environments. In evolutionary biology a main interest is in understanding the role changing conditions play in the overall evolutionary process and in particular what role environmental changes play in survival and reproduction success [28, 39, 69]. This chapter deals with a landscape view on such changing fitness. In particular, we will review in which contexts dynamic fitness landscapes occur, what mathematical descriptions are suitable to be implemented in computable models for conducting numerical experiments, and how the descriptions reflect properties of the underlying dynamic structure. In the following we will focus on landscape approaches in evolutionary computation. This means that the landscape's configuration space is a search space usually originating from an optimization problem and that the landscape may be populated by individuals of an evolutionary search algorithm. However, almost all the discussion applies likewise to a more biological context, where the search space is replaced by a genotypic space that is inhabited by phenotypic realizations.

Dynamic landscape phenomena may occur in different contexts. A first and most obvious is that fitness is directly dependent on time and space. This leads to solving dynamic optimization problems and dynamic fitness landscapes. We will call that explicit time– or space–dependent and will distinguish between *internal dynamics* that is proprietary to the mathematical description of the fitness landscape and *external dynamics* that is generated by a separate driving system and subsequently imprinted on the landscape. Another context is that fitness of one individual depends on the fitness or the phenotype of others. This is the case for different types of coevolutionary settings and creates coupled fitness landscapes [34] (also sometimes called interacting fitness landscapes). If it happens within the frame of the time considered, we also have a situation where fitness is changing with time. This time–dependence is implicit and will be called *population–based dynamics*. However, although there is a substantial amount of work on coevolutionary phenomena, see for instance [49] and references cited there, how this creates dynamics and how the dynamics affect the evolutionary search is rarely addressed. One attempt to make the population–based dynamics in coevolution explicit are deformable fitness landscapes [17, 19]. In the following we aim at giving a unified view about fitness landscapes whose fitness values change with time, and in particular to relate dynamic fitness landscapes to coupled and deformable ones.

In the next section, we will briefly review the current understanding of dynamics in evolutionary processes. This is followed by defining and discussing dynamic fitness landscapes and dynamic optimization problems. Section 10.3 deals with coevolutionary processes. We recall the main ideas and issues of coevolutionary models and algorithms and show how coupled and deformable fitness landscapes derive from them. In particular, we draw connections between these landscapes and the dynamic fitness landscapes considered before. In the concluding section we provide a broader view on fitness landscapes that change with time and outline potential directions of future research.

10.2 Dynamic Fitness Landscapes

10.2.1 Dynamics? What Dynamics?

It seems to be generally acknowledged that every timely change of any given quantity is connected with dynamics. This is in agreement with the word's origin from the ancient Greek *dynamis* (δύναμις), which can be translated as 'ability', 'capacity', 'potentia' or 'power' for doing or moving or changing something. However, there is some confusion as to whether dynamics is the cause, the effect or the process of changing. In addition, the nature of the changes may differ substantially from case to case with the result that the changes may or may not affect what we consider in a particular study. A defining factor to answer this question is how the time frame of what is considered relates to the time frame of the changes. The relations between these time frames can be interpreted as to define a relative speed of the changes. If the speed of the changes is below a certain threshold, it can be neglected as it does not influence results and outcome. Moreover, sometimes even changes of relevant speed may not be taken into account as they are (or can be considered as) spatially, functionally or structurally separated from the object of study. This clearly applies to studies in natural and artificial evolution. Here, it is frequently of interest what severity the change has in terms of fitness. However, if evolutionary speed is defined as the magnitude of the derivative of the evolutionary relevant quantity with respect to time, severe changes are tantamount to high speed dynamics.

Two types of dynamics can be distinguished. A first is evolutionary dynamics. Evolutionary dynamics describes how individuals of an evolutionary search algorithm move in the search space as generations go by [2, 23]. As each point in a search space has a fitness value, and this fitness value can be seen as proprietary to the individual, the movement can be pictured as to happen on the fitness landscape's surface (also see Chapter 1 of this book). This view remains valid in a more evolutionary biology point of view. Evolutionary dynamics here means the movements of phenotypic realization through the genotypic space. Most important here is that evolutionary dynamics is expressed and counted on a generational base. This is in line with the genetic makeup of an individual being constant during its lifetime. In this view adaption within the lifetime of the individual such as learning, phenotypic adaption, plasticity and polymorphism is not directly taken into account. Of course, such traits may tune the fitness of the individual, but the ability to do so is clearly fixed in the phenotype. So, in the end only genetically inheritable traits form the base of fitness, which is considered to be a unique (and therefore constant) value for the time interval of a generation. This allows the interpretation that fitness of an individual is a single value of 'lifetime fitness', the calculation of which is subject to all activities within the lifetime. Considering this, it should be clearly pointed out that evolutionary dynamics as such takes place on a static fitness landscape.

Next to evolutionary dynamics, another type of timely change is of interest here, which we call environmental dynamics. Under environmental dynamics we subsume changes outside the considered individual that have influence on its fitness. That might be abiotic changes such as shifts in climate or other physical attributes of the living space, or biotic changes such as alterations in the food and/or prey abundance, or predator and/or parasites existence, or interaction between species or within them. Our interest is in evaluating the effect these environmental changes have on the fitness of individuals. In other words, we study how environmental dynamics relates to evolutionary dynamics. Therefore, it is sensible to relate the time scale of the environmental changes to the generational time of evolutionary dynamics. As fitness of an individual is a single (constant) value for each generation, there are two consequences. A first is that changes that occur over the lifetime of an individual should be considered as to aggregate in their effects in order to result in a lifetime fitness; a second is that the effects of environmental changes should be counted at discrete points of time. From these thoughts it also follows that the environmental changes are the source for the dynamics in dynamic fitness landscapes.

To summarize we define a fitness landscape to be dynamic if (and only if) its fitness takes different values for the same search space point over the time frame of consideration. In our understanding there are many different values over time (meaning clearly more than one) and the changes are scattered over the time frame. A most obvious example for that is a fitness landscape that changes frequently over the run–time of the evolutionary search algorithm. In a more biological context this relates to substantial environmental dynamics while studying a long series of generational evolution of a single species. This is in contrast to situations where the fitness landscape might be subject to some parameters, which can be tuned or adapted for each time frame of consideration. This is known as tuneable fitness landscape and almost always means that fitness values of the landscape are affected, that is tuned, but only once and before the time frame of consideration. This is particularly relevant in considering NK and NKC landscapes and will be discussed below. The time frame of consideration may be an evolutionary run, or studying an interval of an evolutionary development, or a random walk if we design a landscape analysis. Mathematically, this understanding implies that the fitness is a variable depending on time, as opposed to fitness as a parameter to be set before a (numerical) experiment.

In evolutionary computation a fitness landscape is often linked to an optimization problem. We next extend this view to a dynamical setting. We first look at a static optimization problem. It consists of an objective function (frequently equated with a fitness function in evolutionary computation) $f(x)$ defined over some search space S with $x \in S$. We assume that the search space is metric, or that there are some other arrangements about which search space points are next to each other. Optimization means to find the lowest (or highest) value of $f(x)$ and its coordinates among all $x \in S$:

$$f_S = \max_{x \in S} \quad f(x), \tag{10.1}$$

with the location $x_S = arg\, f_S$.

The static optimization problem in Equation (10.1) can be thought of as becoming dynamic by solving it not just once, but somehow modified for a second time. For a modified fitness function $f^*(x)$, we may write the modified problem as

$$f_S^* = \max_{x \in S} \quad f^*(x). \tag{10.2}$$

To rewrite the two static problems in Equations (10.1) and (10.2) as one dynamic problem, we introduce the time variable $k \in \mathbb{N}_0$ (which is nothing other than a counting variable) and define the dynamic fitness function $f(x,k)$, where

$$f(x,0) = f(x), \quad f(x,1) = f^*(x).$$

We may carry on with doing so for the next modification of $f(x)$ to obtain $f(x,2)$, and so on. Hence, a dynamic optimization problem is

$$f_S(k) = \max_{x \in S} \quad f(x,k), \quad \forall k \geq 0 \tag{10.3}$$

with the solution trajectory $x_S(k) = arg\, f_S(k)$. To define a dynamic optimization problem in the given way suggests that there might be more than one way to define a dynamic problem out of a series of static problems, or that we may obtain different dynamic problems out of modifying one and the same static problem. The discussion above may also imply that the modifications alter the problem only lightly. However, even if the modified fitness function $f^*(x)$ is fundamentally different from the function $f(x)$ it is still possible to define both problems such that the one dynamically originates from the other. However, such a point of view is rather futile in terms of usefulness for evolutionary computation. As both problems here have hardly any relations to each other, the solution of one of them gives no information employable for evolutionary solving the other more efficiently. The best way to solve them both by evolutionary means is to let the algorithm run independently for both problems. In other words, to consider a sequence of problems to be dynamic makes most likely more sense if the problems show some alikeness and relation to each other. A main assumption here is that similar problems are best solved with similar algorithms. Moreover, there should be a way to utilize these relations for equipping the evolutionary search algorithm with favorable settings. These settings might concern parameters (for instance general parameters such as population size or diversity management parameters such as hyper–mutation rate or random immigrant proportion) or genetic operators (what kind of selection, recombination or mutation) or collectible information for equipping triggered diversity management schemes such as memory or anticipation/prediction.

10.2.2 Definition of Dynamic Fitness Landscapes

In the last section we have shown how a series of static problems can create a dynamic problem. In the same way we may regard a series of static fitness landscapes as a dynamic fitness landscape. Next, we will formalize this approach. A static fitness landscape Λ_S can be defined by (see e.g. [24, 68])

$$\Lambda_S = (S, n, f), \tag{10.4}$$

where S is the search space with elements $x \in S$, $n(x)$ is a neighborhood function which orders for every $x \in S$ a set of direct and possibly also more distant neighbors (and hence makes the space at least measurable, if not metric), and $f(x) : S \to \mathbb{R}$ is the fitness function giving every search space element a fitness value. The search space is either the product of a genotype–to–phenotype–to–fitness mapping or constructed from encoding and distancing the set of all possible solutions of an optimization problem. Either way it is basically the representation that the evolutionary algorithm uses (for instance binary, integer, real, tree etc.) and the design of the genetic operators that defines the search space, and also its neighborhood structure. This is in line with the general understanding that the concept of fitness landscapes is particularly useful for studying how the evolutionary search algorithm interrelates with the fitness function [24, 67, 71]. Moreover, as shown in [23], the neighborhood structure of a fitness landscape may vary with variation of the genetic operators. Hence, an analysis of the fitness landscape can be helpful for designing genetic operators as it gives insight into which design is more likely than others to belong to the landscape easiest searchable on average [71]. If the representation is fixed, for instance as real numbers, then the search space frequently has a metric and the neighborhood structure is inherent. The exact design and the parameters of the genetic operators, for instance the mutation strength, define which points can be reached on average from a given starting point in the landscape within one generation.

The geometrical interpretation that is central to the intuitive understanding of fitness landscapes is particularly visible if $S \subseteq \mathbb{R}^2$ (see Figure 10.1). The interpretation means that a search space in connection with a neighborhood structure forms a location (and hence makes the space measurable) and that fitness can be viewed as a height over the location. In other words, fitness is a property belonging to the search space as the space's orthogonal projection. Therefore, search space points with high fitness appear as peaks, while low fitness regions are valleys, and points with the same fitness form plateaus. Solving the corresponding optimization problem hence means, according to Equation (10.1), to find the highest peak $f_S = \max_{x \in S} f(x)$ and its location $x_S = arg\, f_S$. The individuals of an evolutionary algorithm used to solve the given optimization problem can be thought of as to populate the fitness landscape. The design of the genetic operators (basically selection, recombination, and mutation) is meant to organize such that they (at least in average) perform a climbing of the hills in the fitness landscape and ideally find the highest one, even in the presence of other (but smaller) hills. With the movement, the individuals contribute to the dynamics of the population and hence to the evolutionary dynamics. Note that in analyzing these processes a distinction can be made between the dynamics generated by genetic variation (mutation and recombination) and the dynamics generated by corrective guidance (selection), which is of considerable interest for fine–tuning the genetic operators.

In order to define a dynamic fitness landscape, there is the need to set how the elements in Equation (10.4) change over time. With respect to the view of a dynamic fitness landscape that is generated by a series of static landscapes, defining dynamics means to explain how one landscape produces the temporary subsequent one. All of the landscape's three defining ingredients (i.e. search space S, fitness function $f(x)$, and neighborhood structure $n(x)$) can, at least in principle, be dynamically changing. Hence, we additionally need a time set and mappings that indicate how S, $f(x)$ and/or $n(x)$ evolve with time [58, 59]. Dynamic optimization problems considered in the literature so far address all these possibilities of change to some extend. Whereas a real alteration of the fundamental components of a search space such as dimensionality or representation (binary, integer, discrete, real, etc.) is really rare, a change in the feasibility of individuals is another and less substantial kind of dynamic search space and is discussed within the problem setting of dynamic constraints [43, 60, 61]. Next is a changing neighborhood structure which can partially be found in the context of works on dynamic routing [4, 73]. However, most of the work so far has been devoted to time–dependent fitness functions [37, 42, 56, 62, 63, 65, 72, 78]. For these reasons, and to keep the next definition simple, a dynamically changing search space and neighborhood structure is omitted in the following, only the fitness function is time–dependent. A definition encompassing all three landscape components to be dynamic can be obtained by adding transition maps for the other quantities.

We define a dynamic fitness landscape as

$$\Lambda_D = (S, n, K, F, \Phi_f). \tag{10.5}$$

Equivalent to the static landscape of Equation (10.4) the search space S represents all possible solutions to the optimization problem and the neighborhood function

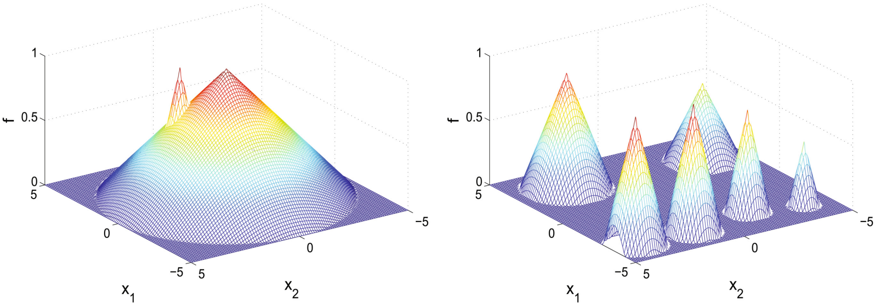


Fig. 10.1 A static fitness landscape in 2D: static peaks that may become moving peaks

$n(x)$ gives a set of neighbors to every search space point. The time set $K \subseteq \mathbb{Z}$ provides a scale for measuring and ordering dynamic changes; F is the set of all fitness functions in time $k \in K$ and so every $f \in F$ with $f : S \times K \rightarrow \mathbb{R}$ also depends on time and gives fitness values to every search space point for any $k \in K$. The transition map $\Phi_f : F \times S \times K \rightarrow F$ defines how the fitness function changes over time. The map must satisfy the temporal identity and composition conditions, that is $\Phi_f(f,x,0) = f(x,0)$ and $\Phi_f(f,x,k_1+k_2) = \Phi_f(\Phi_f(f,x,k_1),x,k_2)$, $\forall f \in F$, $\forall x \in S$, $\forall k_1,k_2 \in K$ and the spatial boundary conditions $\Phi_f(f,x_{bound},k) = f(x_{bound},k)$, $\forall f \in F$, $\forall k \in K$ where x_{bound} is the boundary set of search space S. With these definitions we assume that the changes in the fitness landscape happen (or come into effect) at discrete points in time and are the result of comparing the landscape at points in time k to the following points $k+1$. This is in line with fitness landscapes being a tool for analyzing the behavior of evolutionary algorithms. A generation of an evolutionary algorithm can be defined as the time interval between subsequent fitness function evaluations of the whole population. In other words, a generation indicates the time between serial and self–contained steps in the solution finding process. As fitness evaluation in an evolutionary algorithm usually takes place just once in a generation, a difference in fitness can only be noticed by the algorithm at discrete points in time. Hence, if we model the changes by a fitness landscape, the most natural and straightforward time regime is discrete time.

An intuitive geometrical interpretation of a static fitness landscape still holds to some extend for the dynamic case. The main difference is that the hills and valleys move within the search space and/or change their topological form. This includes that hills grow and shrink, valleys deepen or flatten, or the landscape completely or partially turns inside out. According to Equation (10.3), the corresponding dynamic optimization problem now reads $f_S(k) = \max_{x \in S} f(x,k), \forall k \geq 0$ which yields the temporarily highest fitness $f_S(k)$ and its solution trajectory $x_S(k) = arg\, f_S(k), \forall k \geq 0$.

As before the individuals of the evolutionary algorithm are meant to climb the hills, and moreover to follow if they are moving and find hills that dynamically appear. Even from such a simplifying picture it becomes obvious that the standard genetic operators (selection, recombination, mutation) might not be sufficient to perform the task. Indeed, there exists a multitude of modifications to deal with the changes induced by a dynamic fitness landscape, for instance different types of memory [7, 60, 65], random–based diversity enhancement techniques [42, 72] or anticipation and prediction schemes [5, 62, 66]. Equation (10.3) describes the dynamic optimization problem and its solutions. However, for evaluating the performance and the behavior of an evolutionary algorithm used to deliver such solutions, other quantities can be more interesting, meaningful and significant. These quantities usually generalize the solution trajectory in Equation (10.3) over the run time and/or runs of the evolutionary algorithm, may include data from the evolving population's fitness and make them statistically evaluable. See [1, 75] but also Chapter 14 and references cited there for an overview about dynamic performance evaluation.

Above, we have argued for dynamic fitness landscapes useful in evolutionary computation to have a time regime that only allows changes at discrete points in time. We will end this section with relaxing this kind of modelling and reviewing the effects this has for mathematical descriptions of dynamic fitness landscapes. At first it can be noted that the definition in Equation (10.5) is still valid and meaningful if we consider a real value time set $K = \mathbb{R}$. For not confusing the elements of the discrete valued time set with the real one, we write $t \in \mathbb{R}$. A main consequence of such a real valued time set is that we can mathematically describe dynamic fitness landscapes that change continuously in time. Once we have defined a search space that is metric (or at least measurable) a fitness function can be defined in very general terms. Every $f : S \to \mathbb{R}$ that maps points from the said search space to a scalar variable can be interpreted as a fitness function. Hence, defining the timely change of the function $f(x,t)$ defines a dynamic fitness function. If there are a countable number of possible solutions x_i (and it hence is a combinatorial optimization problem) the dynamics of each of the solutions can be put as depending on the fitness of the solution x_i at time t as well as on the fitness of the μ other solutions

$$\frac{df(x_i,t)}{dt} = \psi_i \left(f(x_1,t),\ldots,f(x_i,t),\ldots,f(x_\mu,t) \right). \tag{10.6}$$

With Equation (10.6) we obtain as dynamic fitness landscape a lattice of (nonlinear) ordinary differential equations (ODEs). There are a considerable number of dynamic combinatorial optimization problems for which the timely evolution of the fitness $f(x_i,t)$ does not depend on all the μ solutions but only on solutions neighboring x_i. Hence, Equation (10.6) modifies to

$$\frac{df(x_i,t)}{dt} = \psi_i^* \left(f(x_i,t), f(n(x_i),t) \right)). \tag{10.7}$$

For a non–countable number of possible solutions the dynamics of the fitness $f(x,t)$ may be expressible by a nonlinear partial differential equation (PDE). If the properties of the search space allow to define spatial derivatives, we obtain (for phenotypes described by n–dimensional vectors) the PDE

$$\frac{\partial f}{\partial t} = \psi \left(f(x,t), \frac{\partial f}{\partial x_1}, \cdots \frac{\partial f}{\partial x_n}, \ldots, \frac{\partial^2 f}{\partial x_1^2}, \ldots, \frac{\partial^2 f}{\partial x_1 \partial x_n} \cdots \right). \tag{10.8}$$

With Equation (10.8), we have a description of the changing fitness as a spatially extended dynamical system. The timely evolution of fitness values specified by the PDE does not depend on all the other fitness values or an explicitly assignable neighborhood, but on the spatial derivatives $\frac{\partial f}{\partial x_i}$ of first and higher order. Geometrically interpreted, this means the timely evolution of the fitness of any search space point is subject to differences in fitness of points laying infinitesimally around that point. To express it differently, the fitness evolution is a function of the curvature the fitness creates in the landscape, or how the fitness deforms the landscape spatially.

It could be argued that Equations (10.6), (10.7) and (10.8) are the most basic way of describing the timely evolution of a fitness landscape as any change in nature must be a change in physical entities whose dynamics is in continuous time. However, it is most unlikely that the equations can be directly used as computational models because they do only in exceptional cases have an analytic formula–like closed solution. Any computational approach therefore requires to discretize time for Equations (10.6) and (10.7) and time as well as space for Equation (10.8). At the end of the process there should be an iterative generation law describing how a fitness value $f(x,t)$ evolves into $f(x,t+\delta t)$, with δt being a small time increment. For Equation (10.7) (likewise for Equation (10.6)) we thus end up with

$$f(x_i,t+\delta t) = \Psi_i\left(f(x_i,t), f(n(x_i),t)\right) \tag{10.9}$$

and for Equation (10.8) with

$$f(x,t+\delta t) = \Psi\left(f(x,t), f(x_1+\delta x_1,t), f(x_2+\delta x_2,t), \ldots, f(x_n+\delta x_n,t)\right) \tag{10.10}$$

where Ψ_i and Ψ are some generator mappings. With these equations we obtain a temporal and spatial discretization employable in numerical algorithms for calculating the evolution of fitness values recursively forward in time. To formalize this computational approach we go back to the discrete time variable k and introduce a temporal renormalization. With this, and because $x_i + \delta x_i$ imposes a spatial discretization and can be interpreted as a neighborhood, we may rewrite and generalize Equations (10.9) and (10.10) as

$$f(x_i,k+1) = \phi_i\left(f(x_i,k), f(n(x_i),k)\right) \tag{10.11}$$

with ϕ_i being a generator mapping.

Note that this neighborhood $n(x_i)$ is not necessarily the same neighborhood used in laying out and distancing the search space according to the definition in Equation (10.5). There might be one neighborhood that defines which search space points are next to a given point, and another neighborhood that expresses which fitness values are influencing the timely evolution of that point. With formulating Equation (10.11) we argue that this recursive evolution law is not only indispensable in calculating a dynamic fitness landscape but also an integral part in defining it. Solving Equation (10.11) yields the transition map Φ_f in the definition in Equation (10.5). So, no definition of a dynamic fitness landscape is complete without giving such an equation. The next section reviews some examples of dynamic fitness landscapes and such equational descriptions are explicitly given.

10.2.3 Examples of Dynamic Fitness Landscapes

As shown in the definition given in Equation (10.4), algebraic equations can be used to describe static fitness landscapes. As further shown in Equation (10.5), for dynamic fitness landscapes we need to add a mathematical prescription for evolving the fitness values forward in time. In principle, there are two ways for doing so. A first is to take the algebraic description of a static fitness function, select some terms from these equations, and define how these terms depend on an additional (and usually external defined) time regime. Geometrically speaking, this means that we define dynamics laws for how selected topological features in the landscape evolve with time. This only implicitly results in explaining the landscape's dynamics in the whole. On the other hand, this also implies that we have, in fact, dynamics only for certain elements in the landscape. By changing these elements dynamically, we implicitly also describe how the neighborhood of the peaks behave with time. We will call this kind of dynamics generation *external*.

The easiest and most obvious way for defining such time–dependent landscapes is to use a static landscape and a set of rules for changing certain features in the landscape with time. A well–known example is the so–called moving peak benchmark [7, 41], which uses as fitness function a static n–dimensional field of peaks (or cones) on a zero plane $f(x) = \max\left\{ 0\,,\, \max_{1 \leq i \leq N} [h_i - s_i \| x - c_i \|] \right\}$ (see also Figure 10.1). Here, $x \in S$ is an element in search space $S \subset \mathbb{R}^n$, c_i is the coordinate of the i–th cone, there are N cones in total, and h_i and s_i are the height and slope, respectively, of the i–th cone. By using the discrete time variable k, we may set dynamic sequences for coordinates $c(k)$, heights $h(k)$ and slopes $s(k)$ to obtain a dynamic fitness landscape with moving peaks

$$f(x,k) = \max\left\{ 0\,,\, \max_{1 \leq i \leq N} [h_i(k) - s_i(k) \| x - c_i(k) \|] \right\}. \tag{10.12}$$

For all static landscapes $f(x)$ we may similarly identify topological features in the the landscape's mathematical description and change elements of the mathematical description dynamically. There are other problems such as dynamic sphere, dynamic Ackley, dynamic Rosenbrook etc., but also dynamic combinatorial optimization problems such a dynamic knapsack, dynamic royal road or dynamic bit–matching [70] that fit into this category, see e.g [15, 44] for an overview of such kind of dynamic problems.

If a dynamic fitness landscape relies on external dynamics, the question of how to generate dynamic sequences arises. A first step is to select terms in the algebraic fitness landscape description that are to change with time. For the moving peak benchmark in Equation (10.12), these are $c_i(k)$, $h_i(k)$ and $s_i(k)$. The dynamic changes are induced by moving sequences

$$z = (z(0), z(1), \ldots, z(k), z(k+1), \ldots) \tag{10.13}$$

that is

$$c_i(k) = z_{ci}(k), \quad h_i(k) = z_{hi}(k), \quad s_i(k) = z_{si}(k). \tag{10.14}$$

In principle, the dynamic changes can be of three types:

- regular dynamics
- chaotic dynamics
- random dynamics.

The moving sequences $z(k)$ can be generated according to these types. Regular changes are usually obtained by analytic coordinate transformations, for instance cyclic dynamics where each $z(k)$ repeats itself after a certain period of time and shows recurrence or translatory dynamics where the quantities ascribe a pre–defined track or tour. The period of the recurrence and the cycle width can be adjusted and normalized so that moving sequences become comparable. For instance, cyclic dynamics can be generated by

$$z_i(k) = g_i(k) = \sin(\omega_i k + \delta_i) \tag{10.15}$$

with ω_i and δ_i appropriate frequencies and phases. Note that an analytic equation such as Equation (10.15) enjoys the property to be not recursive. The value of $z(k)$ can be calculated without knowing $z(k-1)$. This also means a fitness landscape externally driven by Equation (10.15) has strictly speaking no recursive evolution law as given in Equation (10.11), but is knowable beforehand for any given point in time k. It is hence completely predictable.

Chaotic changes can be generated by a discrete–time dynamical system,

$$z(k+1) = g(z(k)), \tag{10.16}$$

which is recursive. Such systems are known to show chaotic behavior for certain parameter values and initial states $z(0)$, for instance the generalized Hénon map. Refer to [55, 56] for details of the generation process. For using these moving sequences in numerical experiments, there might be the need for preprocessing as (depending on the dynamical systems used) the amplitudes $z(k+1)$ might be not unitary. If so, a re–normalization should take place. We get random changes if we select that each $c_i(k)$, $h_i(k)$, $s_i(k)$ for each k is an independent realization of, for example, a normally or uniformly distributed random variable. Again, the statistical properties of the random variable should guarantee (maybe after renormalization) comparability. In some sense, also fitness landscapes externally driven by a random process have no recursive evolution law as given in Equation (10.11). The main difference is here that the value of $z(k-1)$ does in no way specify the value of $z(k)$. The evolution of a random fitness landscape is only statistically describable. A general feature of the three types of dynamics is that regular dynamics is completely predictable, chaotic dynamics is short–term predictable, and random dynamics unpredictable. This property transforms to fitness landscapes externally driven by these dynamics.

A second example for external drive is the XOR-generator by Yang [77, 78]. This generator can be used for any binary–encoded static function $f(x)$ as follows. For each discrete step in landscape time k, a XOR mask $M(k)$ is incrementally calculated by

$$M(k) = M(k-1) \oplus T(k), \tag{10.17}$$

where "$\oplus$" is a bitwise exclusive-or (XOR) operator (i.e., $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, and $0 \oplus 0 = 0$) and $T(k)$ is an intermediate binary template generated for time step k. $T(k)$ is generated with $\rho \times l$ ($\rho \in (0.0, 1.0]$) randomly selected bits set to 1 while the remaining bits are set to 0. For the initial step $k = 1$, $M(0)$ is set to a zero vector, i.e., $M(0) = 0$. The static fitness function $f(x)$ finally becomes dynamic by

$$f(x,k) = f(x \oplus M(k)). \tag{10.18}$$

The XOR generator is hence a way to externally drive a binary fitness function by the recursive law in Equation (10.17).

In a second approach to describe dynamic changes, we may formulate a general law for the fitness landscape's time evolution that applies to all fitness values in the search landscape. Hence, the specification of the timely changes are part of the mathematical description of the landscape. Thus, the fitness of every point $f(x, k+1)$ may depend on the fitness one time step before, $f(x,k)$ and the (element–wise) fitness values of all of its neighbors, $f(n(x),k)$ (see Equation (10.11)). With such a description we have formulated a dynamic fitness landscape as a spatially extended dynamical system. To have such description means that the topology and the dynamics of the fitness landscape are generated simultaneously and by the same equation. In other words, the dynamics here is *internal* to the fitness landscapes. An example for such an evolution law is fitness landscapes constructed from ordinary differential equations (ODE), partial differential equations (PDE) and coupled map lattices (CML). A special property of this type of dynamic fitness landscape is that not only is time discrete but also the search space has a countable number of elements. Such a search space characteristic corresponds strictly speaking to combinatorial optimization problems (see [58, 59] for further discussion).

For a two–dimensional real valued search space S, a dynamic fitness landscape can be viewed as the time evolution of the surface over a 2D plane at point x and time t. Such a general dynamic 2D fitness landscape that describes the dynamics of the fitness value $f(x_1, x_2, t)$ with continuous spatial variables (x_1, x_2) and continuous time t can be modelled by the parabolic PDE

$$\frac{\partial f}{\partial t} = a_1 \left(\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} \right) - a_2 g_1 \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right) + g_2(f), \tag{10.19}$$

where a_1, a_2 are coefficients and g_1, g_2 are mappings. The dynamic fitness landscape in Equation (10.19) can be interpreted as a reaction–diffusion system with an additional nonlinear term and is a special case of the general description given in Equation (10.8). This type of PDE has close resemblance to the Kardar–Parisi–Zhang (KPZ) equation [27], which has been proposed to model surface growth. The main difference is that the KPZ equation includes an explicit stochastic (Gaussian noise) term. Recently, the KPZ equation has been intensively studied [30, 36, 40], with the relation to coupled map lattices (CML) as a central question. Clearly, both are models of extended dynamical systems. Also, and as mentioned before, a numerical solution to a PDE always requires some kind of discretization of space and time. An alternative strategy to a study by any of the methods for numerically solving the PDE therefore appears to consist of a study of the corresponding CML and their mutual dynamical properties.

We will next relate the PDE–based landscape, characterized in Equation (10.19), to a landscape based on a CML [57, 58]. It has been shown that the CML landscape can be obtained by a spatial and temporal discretization of the PDE landscape [59]. For the CML, we lay out a lattice grid with $I_1 \times I_2$ equally sized cells, which builds a $2D$–structure. For every discrete time step k, $k = 0,1,2,\ldots$, each cell is characterized by its height

$$f(x_1,x_2,k), \quad x_1 = 1,2,\ldots,I_1, \quad x_2 = 1,2,\ldots,I_2, \tag{10.20}$$

where (x_1,x_2) denote the spatial indices in vertical and horizontal directions, respectively (refer to Figure 10.2). We interpret this height, $f(x_1,x_2,k)$, as fitness according to the geometrical metaphor of a fitness landscape. It is subject to changes over time, which are described by the two–dimensional CML with nearest–neighbor coupled interaction [10, 25]:

$$\begin{aligned} f(x_1,x_2,k+1) = (1-\varepsilon)g(f(x_1,x_2,k)) + \frac{\varepsilon}{4}\Big[\, & g\,(f(x_1-1,x_2,k)) + g\,(f(x_1+1,x_2,k)) \\ & + g\,(f(x_1,x_2-1,k)) + g\,(f(x_1,x_2+1,k))\,\Big], \end{aligned} \tag{10.21}$$

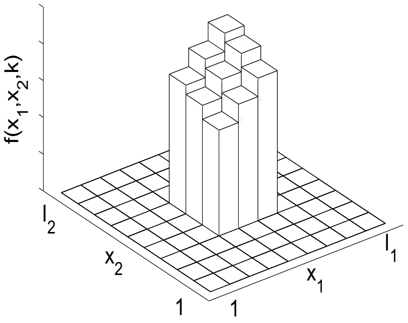


Fig. 10.2 The coupled map lattice (CML) of Equation (10.21) as an example for internal dynamics in a fitness landscape

where $g(f(x_1,x_2,k))$ is a local mapping function and ε is the diffusion coupling strength. As local mapping function we use the logistic map,

$$g(f(x_1,x_2,k)) = \alpha f(x_1,x_2,k)(1 - f(x_1,x_2,k)). \tag{10.22}$$

It is a nonlinear map with the parameter $0 < \alpha < 4$, which is defined for the unit interval $f \in [0,1]$. For some parameter α, the map given by Equation (10.22) exhibits chaotic behavior, for instance in the parameter interval $\alpha \in [3.57,4]$. The local chaotic behavior is distributed to other areas of the lattice by coupling. So, it is the source of spatio–temporal chaos in the extended dynamical system. Finally, we need to set the period boundary conditions as

$$\begin{aligned} f(I_1+1,x_2,k) &= f(1,x_2,k), \\ f(x_1,I_2+1,k) &= f(x_1,1,k). \end{aligned} \tag{10.23}$$

Initialization of the CML is done by initial heights $f(x_1,x_2,0)$, being realizations of a random variable uniformly distributed on $[0,1]$. The spatio–temporal behavior of the CML depends on the lattice size $I_1 \times I_2$ and two parameters, the coupling strength ε and the nonlinear parameter α. The timely evolution of the CML given in Equation (10.21) also provides the recursive evolution law of the dynamic fitness landscape of Equation (10.11).

The CML is known to exhibit a rich spatio–temporal behavior, including different types of spatio–temporal periodicity and chaos, quasi–periodicity and pattern formation. The CML is therefore an instructive example for the principle of generating high–dimensional complex spatio–temporal dynamics by using local chaos created by a low–dimensional mechanism that is transmitted to a spatial extension by coupling. The obtained fitness landscape shows a similar complex behavior. Refer to [57, 58, 59] for detailed studies.

10.3 Coevolution, Codynamics and Their Fitness Landscapes

10.3.1 Coevolutionary Dynamics

This section reviews coevolutionary algorithms and their fitness landscapes. This is done with the aim to provide background for relating coevolutionary fitness landscapes, namely coupled and deformable landscapes, in the framework of dynamic fitness landscapes discussed in the previous section. There is an ongoing debate about the question whether or not fitness landscapes that originate from coevolutionary processes could (or should) be treated as dynamic and what possible benefits (if any) such a treatment could have. In the following we will not argue strongly for either view, but we will demonstrate that such a view is possible.

Coevolutionary algorithms differ structurally from evolutionary algorithms for several reasons. First, there is usually more than one population. Thus, there is no longer only one genotypic space populated by individuals of one species, but several spaces in which the evolutionary dynamics of other species take place. In this view, coevolution operates on the level of phenotypes. Anyway, coevolution also has an impact on the fitness landscape as a whole, that is, on the level of genotypes. For the dynamic interdependency on this level of the landscape we will introduce the term *codynamics*. Coevolution between species results in codynamics between the respective fitness landscapes. We will restrict ourselves here to coevolution between two species (and hence the dynamic interaction, that is, codynamics, of two fitness landscapes). Systems with more than two species can, at least in principle, be treated along the same mathematical framework.

A second reason is that the fitness of the individuals of both species are interdependent and as a consequence coevolution relies upon an alternative concept of how the fitness of an individual is generally defined. This is the crucial element of coevolution. There are some implementations [18] that only use a single population to create coevolutionary effects, but these implementations as well as the ones with multiple populations all share the following property: they use a population–dependent fitness. This fitness is subject to the progress the search process itself is making. It therefore is also called a subjective fitness, as opposed to objective fitness in conventional evolutionary computation. As the search process is dynamic, this necessarily results in dynamic fitness. This is in contrast to fitness evaluation considered so far. For an evolutionary algorithm (and for static problems) the fitness value is a property of a search space point (or a given genotype) and is constant for the entire run–time. So, if in the course of the evolutionary search, the same point were to be visited again by an individual, it would yield the same fitness value as in the visit before. In other words, a reevaluation of a specific genotype always gives the same fitness value. For an evolutionary algorithm and a dynamic problem the search space points may change their fitness values but this happens because the landscape is internally or externally changing with time. This takes place independently from the evolutionary search process and is hence unconnected to the fitness values of the population's individuals. There is no feedback from the evolutionary search to the landscape.

In coevolutionary algorithms fitness is assigned differently. The fitness of an individual is the result of interaction with other individuals. The individuals that serve to interact with the one for which a fitness value is to be assigned are called evaluators. The fitness of an individual at a given point in run–time therefore depends on which individuals are actually selected to interact with, and also on the current fitness of these evaluators. As a consequence, a given point in search space (a genotype) can have a completely different fitness value if it were to be reevaluated. Therefore, a specific genotype frequently has variable, time–dependent fitness where the time–dependence is induced explicitly by the generational search. This is because in the reevaluation other individuals may act as evaluators and even if the same individuals are taken a second time, they might have different fitness values. Only if the evaluators remain the same, and their fitness values do not change in the evolutionary run, we would fall back to the situation where a genotype has a fixed fitness value. This, of course, would make a pointless coevolutionary algorithm. In addition, there are even coevolutionary algorithm implementations that deviate from the simple design principles set out above. Elevators are usually a subset of a population as an interaction with all the members of all possible populations, called full mixing or complete mixing, might be time–consuming and therefore computationally expensive. There are for instance implementations that vary the number of individuals for interaction or the number of interactions with time, either depending on the run–time of the algorithm [46] or depending on some internal states of the algorithm [47]. Other implementations use an archive of past solutions to select evaluators from [45, 48, 64]. All these mechanisms for calculating fitness dramatically complicate and obscure the relationship between fitness and genotype in coevolutionary processes. Clearly these relationships are dynamic but not in the way fitness landscapes are dynamic as discussed in the previous section.

It should be pointed out that much of the work on coevolutionary algorithms has been centered on the question of how to select evaluators and how to set the rules of interaction. A main difference in algorithmic design is drawn between algorithms that have just one population (single population coevolutionary algorithm) from which the evaluators are taken, and algorithms that have many populations (multi population coevolutionary algorithm) with possibly complex rules as to which population may supply evaluators for assigning the fitness of individuals of another population. This may include the case that some populations are only used to provide evaluators. Another difference frequently discussed is the exact nature of the interaction. The main distinctive feature that is used is if the interaction is cooperative or competitive [53, 64]. Cooperative means that the individual and the evaluators interact and collaborate to solve a problem that is harder or impossible to solve by each of them alone. The better they perform together the higher the reward and hence the fitness. This sometimes even means that the same fitness is given to all cooperating individuals. In competitive interaction the individual is rewarded for out–performing the evaluators, which sometimes means that the fitness of one individual is increased at the expense of the others. The terms *cooperative* and *competitive* are frequently used in computer science approaches to coevolution. In the biological and ecological literature, see for instance [3, 8], these terms are sometimes replaced by mutualistic coevolution and antagonistic coevolution.

Further issues occur in the case of the fitness of an individual being the subject of more than one interaction and hence consist of several partial finesses that have to be aggregated. Another topic is relevant if there are multiple populations and communication between them needs to be organized. The most common here is to have a centralized clock that defines the generational time for each population and also specifies at which points in time exchange of information and interaction takes place. We will assume this in the discussion to follow. All these questions are important for coevolutionary search and deeply influence the behavior and performance of such algorithms. On the other hand, they only touch on the main question of this chapter, namely how the fitness landscape describing coevolution relate to dynamics and creates codynamics.

10.3.2 Codynamic Fitness Landscapes

Basically, in coevolution there are as many fitness landscapes as there are populations. Since the focus here is on two coevolving populations, we have, in fact, two fitness landscapes. They might or might not have the same search space and neighborhood structure according to the definition set out in Equation (10.4). Let us assume they have (the case of unequal search spaces would not alter the following discussion substantially), but for distinguishing them we write S_x for the one, and S_y for the other. We now consider that the individuals of population $P_1(k)$ can take possible values $x \in S_x$ and the population $P_2(k)$ may have the values $y \in S_y$. So far we are still in line with the definitions of static (see Equation (10.4)) and dynamic (Equation (10.5)) fitness landscapes as given in Section 10.2.2. However, as a consequence of the coevolutionary allocation of fitness values discussed above, we now face the question of defining fitness values $f(x)$ and $f(y)$, which cannot be done by considering the search spaces separated from each other. This only becomes possible by considering the interaction between both populations. As mentioned before, the fitness here is no longer a constant property of a point in search space; fitness is generated on–the–fly by the process of coevolution. Hence, we might be able to formulate a recursive evolution law just as in Equation (10.11).

To illustrate this fact, it is instructive to combine both landscapes into one shared landscape $S = \{S_x, S_y\}$. This may result in a unique (static) landscape for simple coevolutionary scenarios and is particularly convenient if the coevolving populations are one–dimensional. Therefore, combining the two one–dimensional landscapes leads to a shared two–dimensional landscape. This approach has been followed in previous research on understanding coevolutionary phenomena by fitness landscapes [50, 51]. We use this example to describe the dynamics in coevolving fitness landscapes. Therefore, we consider the coevolution on simple functions, for instance ridge functions as suggested in [50, 51]. The simplest function has one ridge:

$$f(x,y) = n + 2\min(x,y) - \max(x,y) \tag{10.24}$$

with $x, y \in \mathbb{R}$ and n is a parameter that sets the size and the hight of the landscape (see Figure 10.3a). The landscape has a single maximum at $f(n,n) = 2n$ and a ridge diagonally from $f(0,0) = n$ to $f(n,n)$ that separates two planar surfaces. There are two minima at $f(0,n) = f(n,0) = 0$. Equation (10.24) is the fitness function for both populations P_1 and P_2 and can be interpreted as the static shared fitness landscape $S = \{S_x, S_y\}$ of the coevolution process.

The coevolutionary process we consider works as follows. Both populations evolve subsequently and alternatively along the conventional evolutionary algorithm's generational process (fitness evaluation followed by selection, recombination and possibly mutation); that is, population P_1 starts, and after it has finished its first generation, P_2 takes over, then P_1 starts again in the second generation and so on. The main difference to traditional evolutionary computation is how the fitness of either population is calculated, because the fitness evaluation in P_1 is subject to evaluators from P_2, and vice versa. As the populations take turns in evolving, this creates a coupling via the (time–dependent) fitness values (or some quantities derived from them) of the respective population. As an effect, both populations coevolve, and the landscapes show codynamics. Let us now look at how this dynamic behavior is reflected in their fitness landscapes.

We first need to define how the interaction between individuals of one population with evaluators from the other is organized, how the solution of the interaction is calculated, and how the solution translates to (personal or collective) fitness of the individuals. These questions can be addressed by employing the framework of *interactive domains* and *solution concepts* [49]. We here use a very simple version of this framework, as suggested in [51]. The interactive domain to calculate the fitness of population $P_2(k)$ consists of the individuals in P_2 interacting with population P_1 by picking the current best individual $x_{best}(k)$ of $P_1(k)$. The solution concept is as simple as just taking this value and inserting it in Equation (10.24). For P_1 it is the same but reverse, i.e. take the current best individual $y_{best}(k-1)$ and insert in Equation (10.24) to calculate the fitness of population $P_1(k)$. Because the populations take turns in coevolving with population P_1 starting and P_2 following, this calculation has to deal with P_2 lagging behind for one generation. We resolve this by taking a random $y = y_{rand}$ from population P_2 in the initial generation at $k = 1$; that is $y_{best}(0) := y_{rand}$. For $k = 2$, we take the value $y_{best}(1)$ and so on.

Interestingly, for this simple example it is still undefined if the given interaction is either cooperative or competitive. This only depends on the notion of what constitutes the best individual of either population, $x_{best}(k)$ and $y_{best}(k)$, respectively. As shown in [50, 51] a cooperative interaction is imposed if the task for both populations is the same, that is, both are to find the maximum or minimum of the fitness function (10.24). A competitive interaction takes place if one population is to search for the maximum of Equation (10.24), while the other is to find the minimum of Equation (10.24). Either way, we obtain the same codynamics expressed as a coupled fitness landscape. From the perspective of population P_1 the fitness landscape is dynamic by

$$f(x,k) = n + 2\min\left(x, y_{best}(k-1)\right) - \max\left(x, y_{best}(k-1)\right), \tag{10.25}$$

while from the perspective of population P_2 we get

$$f(y,k) = n + 2\min\left(x_{best}(k), y\right) - \max\left(x_{best}(k), y\right). \tag{10.26}$$

From either perspective alone it appears that fitness is calculated on–the–fly while the coevolutionary algorithm is running. Due to the simplicity of the example the codynamic fitness landscape can be depicted as a function of time. See Figure 10.3b as an example of cooperative interaction. This figure can be directly derived from the shared fitness landscape in Figure 10.3a by looking from the x–axis and considering the value for $y = y_{best}(k)$ as slices of the S_x space. For the first and the second generation the relatively low values of $y_{best}(k)$ lead to a comparably flat landscape to be searchable for population P_1. Hence, the maximum fitness that can be obtained is relatively low. The theoretically possible maximum cannot be reached at all; the dynamic fitness landscape simply does not include it for these generations. After a certain number of generations, the landscape curves up (due to better values of

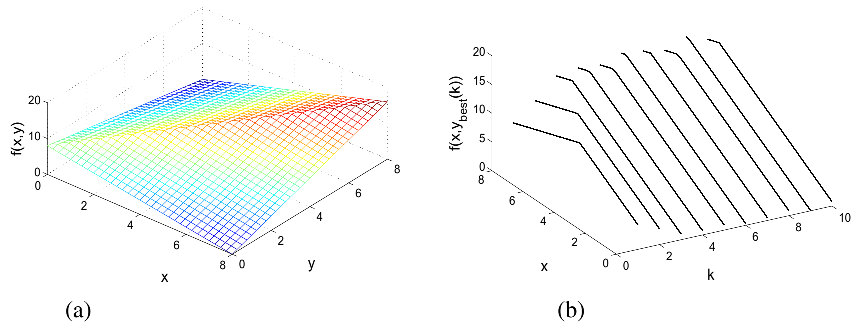


Fig. 10.3 (a) The oneRidge function of Equation (10.24) for $n = 8$. (b) The codynamic fitness landscape $f(x,k)$, see Equation (10.25), depending on k.

$y_{best}(k)$) and the overall maximum becomes finally accessible. It should be noted that this figure is an illustration, and not a verifiable numerical result. This is the landscape for one run, another run might produce a landscape that is similar but different in detail.

These results could have been obtained and presented in this way due to the extreme simplicity of the interaction and the solution concepts of the example. Only so, there is this unique relationship between the codynamic fitness landscapes of Equations (10.25) and (10.26) on the one hand, and the static shared landscape of Equation (10.24) on the other. Even if this simple example would be made more complicated the clear–cut relationship would cease. For instance, the interaction could use not only the current best, but past bests, or it could not only take the best, but a group of high fitness individuals including the best. The solution concept could entail a comparison or other operations of the fitness of picked individuals from P_1 and P_2. All these modifications would make the relationships between the codynamic landscape more complicated and finally question if the static shared landscape like the one given in Equation (10.24) can be uniquely decomposed into codynamic landscapes expressed as Equations (10.25) and (10.26). However, as long as the interaction produces a phenotypic point as a result and the solution concept gives it a fitness value that is unique and constant for the coevolutionary run, codynamic landscapes can be obtained from the overall landscape. Our conjecture is that if a fitness function such as Equation (10.24) is used as solution concept, the codynamic fitness landscapes are subspaces (for instance slices) of the fitness function that obtain their dynamics by being spanned every generation. Let us for instance again consider the example above and assume that the interaction is to pick a certain number of current best individuals from the other population, compare it with another number of best individuals from past generations, discard some individuals due to their low fitness and save the rest, and calculate as the solution of this interaction the weighted average of all individuals that were saved. Surely this is a complicated process, but it again produces a phenotypic point in the end. If the fitness is assigned by the fitness function in Equation (10.24), the situation is not altered significantly. No matter what the interaction yields, the result can finally be interpreted as an individual (or a group of individuals that can be conflated) and that can be inserted in a fitness function such as Equation (10.24). In some way, it could be argued that this kind of coevolution has an objective fitness after all. To generalize, for a shared fitness landscape $f(x,y)$ and the operator $\mathit{interact}(P_1,P_2,\mathit{archive})$ describing the interactive domain (possibly supported by an archive), we obtain the codynamic fitness landscape

$$f(x,k) = f(x, \mathit{interact}\,(P_1(k), P_2(k), \mathit{archive}(k))\,. \tag{10.27}$$

For the variable y describing the coevolution of P_2, we obtain a similar mathematical description. Note that the codynamic landscape of Equation (10.27) implies a recursive evolution law as given in Equation (10.11) if we observe the evolutionary dynamics of the populations $P_1(k)$ and $P_2(k)$ describable by a generation transition function just as given by Equation (10.32). In line with this view and compatible with the external and internal drive in dynamic fitness landscapes considered in Section 10.2, codynamic fitness landscapes have a population–based dynamics.

A considerable number of coevolutionary settings fall into the category for which codynamic fitness landscapes can be models by Equation (10.27). The defining element is that a shared fitness landscape $f(x,y)$ serves as the solution concept. This applies to almost all coevolutionary algorithms used to solve optimization problems by decomposition, which have been called compositional problems [49]. Examples of compositional problems are [20, 22, 48, 50, 51]. For these examples, there are even first promising attempts to measure the resulting codynamic fitness landscapes and draw useful conclusions from it [29]. However, there is also an important group of coevolutionary problems that do not have a solution concept expressible as an equation–like formula known analytically beforehand and generally being knowable without doing numerical experiments. Consider for instance the case where in the course of evaluating an individual, there are multiple interactions with other individuals from the same but also from the other population that have multiple values attached that are aggregated to the individual's fitness. This is frequently the case for so–called test–based problems [11, 16]. For these problems there seems to be no static shared fitness landscape $f(x,y)$ mapping uniquely the set of all possible values x versus all possible values y and equipping every pair with a constant fitness value. To remain with the geographical metaphor of the fitness landscape, we no longer obtain a closed landscape, that is a surface (which may be rugged or smooth), but disjunct fitness islands. All we get is a phenotypic landscape but not a genotypic one. We conjecture that it may generally hold true that compositional problems have a shared fitness landscape, while test–based problems do not.

10.3.3 Examples of Modelling Coevolutionary Processes

A classical and well–known example of a fitness landscape that has the potential to be time–dependent, but also offers the ability to model coevolutionary couplings between different species, are the so–called NK or NKC landscapes introduced by

Kauffman, collaborating with Levin and Johnsen [31, 32, 34], where NK and NKC denote the tunable parameter of the landscape. These landscapes use a genotypic coding with a string of length N over a given alphabet, where in most of the implementations the alphabet of length 2, $\mathbb{A} = \{0,1\}$ is considered. The number N gives the number of components that are required to code for each point of the genotypic space. For the binary alphabet we hence have for each genotype a binary string $x = x_0 x_1 \ldots x_i \ldots x_{N-1}$, where x_i is the i–th bit. The parameter K in the NK fitness landscape describes the degree of epistatic interactions between the N components of each genotype. By tuning K between $0 \leq K \leq N-1$, different degrees of ruggedness (that relate to problem hardness) can be adjusted. The epistatic interaction is modelled by setting K neighbors for each bit x_i via a neighborhood function $n(x_i,K)$ and defining a fitness contribution $f_i(x_i,n(x_i,K))$ for each bit x_i and the K neighbors. In principle, there are two ways to set a neighborhood function $n(x_i,K)$. A first is nearest neighbor interaction, where $\frac{K}{2}$ bits on either side of x_i are considered neighbors. As $\frac{K}{2}$ must be an integer, this imposed a bias for odd K to the right or left hand side, and also requires a periodic boundary condition which says that the last and the first bit in the string are direct neighbors. A second design is random interaction, where the K neighbors are chosen at random (with no repetition or reciprocity) from among the remaining $N-1$ bits. The example considered below demonstrates nearest neighbor interaction with a right hand side bias. The fitness of each genotype x is defined by

$$f(x) = \frac{1}{N} \sum_{i=0}^{N-1} f_i(x_i, n(x_i,K)). \tag{10.28}$$

The contributions $f_i(x_i,n(x_i,K))$ are taken as realizations of a random variable uniformly distributed on the interval $[0,1]$ and depend on the interaction parameter K. As an example, consider the binary alphabet and $N = 4$, which gives $2^4 = 16$ genotypes in the landscape. If the Hamming distance H_D between genotypes next to each other is $H_D = 1$, a location is obtained from genotypic space and neighborhood structure, which results in a metric search space as illustrated in Figure 10.4. Tables 10.1 and 10.2 show the calculation of fitness for $x = 0110$.

In order to model coevolution and coupling between different species (and thereby obtaining a coupled fitness landscape), the NK landscape is modified. We next consider the coupling of two species. However, within the given framework the methodology can be straightforwardly extended to an arbitrary number of species. For two coevolving species, we need to define two genotypic sets, which (as an extension of the setting above) are described by two binary strings of length N, $x = x_0 x_1 \ldots x_i \ldots x_{N-1}$ and $y = y_0 y_1 \ldots y_i \ldots y_{N-1}$. Apart from the K epistatic interactions within each genotype (called internal interactions), there are additionally C epistatic interactions from one genotype to the other (and vice versa), which are called external interactions. This yields the name NKC landscape. As before, we can tune $0 \leq C \leq N-1$. For describing these two types of interactions we first need to set the K internal neighbors for each bit x_i and y_i via (usually identical) internal neighborhood functions $n(x_i,K)$ and $n(y_i,K)$. Additionally, C external neighbors in the other genotype need to be specified by an external

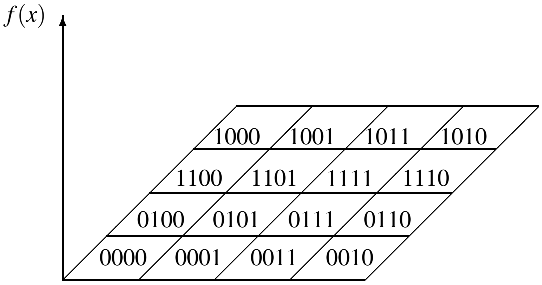


Fig. 10.4 Layout and distancing of the NK landscape defined by Equation (10.28) for $N = 4$ and neighborhood structure with Hamming distance $H_D = 1$

neighborhood function $n(x_i, y_i, C)$. This external neighborhood function is (usually) assumed to be symmetrical, $n(x_i, y_i, C) = n(y_i, x_i, C)$. The effect of internal and external interaction is combined by concatenating both neighborhood functions and results in KC neighborhood functions $n(x_i, K, C) = n(x_i, K)|n(x_i, y_i, C)$ and $n(y_i, K, C) = n(y_i, K)|n(y_i, x_i, C)$. Here $a|b$ means string a concatenated with string b. For these KC neighborhoods fitness contributions $f_i(x_i, n(x_i, K, C))$ and $f_i(y_i, n(y_i, K, C))$ are defined as independent realizations of a uniformly distributed random variable that superpose internal and external fitness contributions. Hence, the fitness of each genotype x and y is given by

$$f(x) = \frac{1}{N} \sum_{i=0}^{N-1} f_i(x_i, n(x_i, K, C)) \tag{10.29}$$

and

$$f(y) = \frac{1}{N} \sum_{i=0}^{N-1} f_i(y_i, n(y_i, K, C)). \tag{10.30}$$

Tables 10.3 and 10.4 give an example of calculating the fitness for $x = 0110$ and $y = 1001$.

So far, neither the NK nor the coupled NKC landscape depends on time. The fitness values do not change during the considered time frame, may that be a random walk on the landscape to calculate some landscape measures or an evolutionary run. With respect to the discussion of coevolutionary processes above, the NKC landscapes defined by the Equations (10.29) and (10.30) do not have timely interactions. Of course, they are coupled via shared fitness contributions f_i but not dynamically. The NK and NKC landscapes can be made dynamic in the same way as the landscapes considered before: by defining that certain landscape features change depending on an external source of dynamics. For the NK landscape this means that

Table 10.1 Example of the NK landscape with $N = 4$ and $0 \leq K \leq 3$ and $x = 0110$. The neighborhood function $n(x_i, K)$ is nearest neighbor interaction with right hand bias, namely $n(x_i, 0) = x_i$, $n(x_i, 1) = x_i x_{i+1}$, $n(x_i, 2) = x_{i-1} x_i x_{i+1}$ and $n(x_i, 3) = x_{i-1} x_i x_{i+1} x_{i+2}$ with the periodic boundary condition $x_N := x_0$. The fitness contributions $f_i(K) := f_i(x_i, n(x_i, K))$ are assigned as realizations of a random variable uniformly distributed on the interval $[0, 1]$, refer to Table 10.2 as an example of a lookup table containing these values. We obtain the fitness function values $f(0110) = 0.4$ for $K = 0$, $f(0110) = 0.5$ for $K = 1$, $f(0110) = 0.475$ for $K = 2$, and $f(0110) = 0.55$ for $K = 3$.

	$K =$ 0					$K =$ 1				
x	0	1	1	0		0	1	1	0	
i	$n(x_i, 0)$				$f_i(0)$	$n(x_i, 1)$				$f_i(1)$
0	0				0.5	01				0.2
1		1			0.1		11			0.4
2			1		0.7			10		0.9
3				0	0.3				00	0.5

	$K =$ 2					$K =$ 3				
x	0	1	1	0		0	1	1	0	
i	$n(x_i, 2)$				$f_i(2)$	$n(x_i, 3)$				$f_i(3)$
0	001				0.1	0011				0.6
1		011			0.5		0110			0.3
2			110		0.6			1100		0.5
3				100	0.7				1001	0.8

Table 10.2 Example of the lookup table for the NK landscape with $N = 4$ and $K = 2$, which is of dimension $N \times 2^{K+1}$

$n(x_i, 2)$	$f_0(2)$	$f_1(2)$	$f_2(2)$	$f_3(2)$
000	0.6	0.3	0.5	0.1
001	0.1	0.5	0.9	0.4
011	0.8	0.5	0.1	0.3
010	0.5	0.8	0.9	0.7
100	0.6	0.3	0.2	0.7
101	0.7	0.9	0.5	0.6
111	0.1	0.7	0.2	0.6
110	0.5	0.3	0.6	0.9

$$f(x, k) = \frac{1}{N} \sum_{i=0}^{N-1} f_i(x_i, n(x_i, K(k)), k). \tag{10.31}$$

Dynamic NKC landscapes can be formulated by introducing time–dependent $K(k)$ and/or $C(k)$ in the models given by Equations (10.29) and (10.30). The landscape in Equation (10.31) also shows that there are two ways for imprinting dynamics on a NK landscape. A first is to change the parameter $K(k)$ (and/or $C(k)$); a second is to change the fitness contributions $f_i(k)$. In some ways the former implies the latter as a changed $K(k)$ and $C(k)$ entails a recasting of the lookup tables and hence changes the fitness contributions as well. The main difference between both variants is that only changing the $f_i(k)$ but retaining the K and C somehow restricts the severity of changes. See for instance [6, 76] for experiments with dynamic NK landscapes. However, it should also be pointed out that dynamic NKC landscapes formulated in a similar fashion as the dynamic NK landscape of Equation (10.31) do not directly model coevolutionary dynamics in the sense of the discussion in Section 10.3.2. Clearly, they are dynamically coupled (and could be interpreted as codynamics), but how the coupling modeled here relates to the coupling via interacting populations is far from being clear. An attempt to clarify such relations are the deformable landscape considered next.

Table 10.3 Example of the NKC landscape with $N = 4$, $K = 1$ and $1 \leq C \leq 2$ and $x = 0110$, $y = 1001$. The KC neighborhood functions $n(x_i,K,C)$, $n(y_i,K,C)$ are symmetric and nearest neighbor interaction, namely $n(x_i,1,1) = n(x_i,1)|n(x_i,y_i,1) = x_i x_{i+1}|y_i$, $n(x_i,1,2) = x_i x_{i+1}|y_i y_{i+1}$, where again the periodic boundary conditions $x_N := x_0$. $y_N := y_0$ are observed. Here $a|b$ means string a concatenated with string b. The fitness contributions $f_i^x(K,C) := f_i(x_i,n(x_i,K,C))$ and $f_i^y(K,C) := f_i(y_i,n(y_i,K,C))$ are assigned as independent realizations of a random variable uniformly distributed on the interval $[0,1]$, refer to Table 10.4 as an example of a lookup table containing these values. We obtain the fitness function values $f^x(0110) = 0.4$, $f^y(1001) = 0.375$ for $K = 1, C = 1$ and $f^x(0110) = 0.4$, $f^y(1001) = 0.5$ for $K = 1, C = 2$.

	K 1	$=C$ 1	$=$								
x	0	1	1	0		y	1	0	0	1	
i	$n(x_i,1,1)$				$f_i^x(1,1)$		$n(y_i,1,1)$				$f_i^y(1,1)$
0	$01\vert 1$				0.3		$10\vert 0$				0.6
1		$11\vert 0$			0.2			$00\vert 1$			0.3
2			$10\vert 0$		0.7				$01\vert 1$		0.1
3				$00\vert 1$	0.4					$11\vert 0$	0.5

	K 1	$=C$ 2	$=$								
x	0	1	1	0		y	1	0	0	1	
i	$n(x_i,1,2)$				$f_i^x(1,2)$		$n(y_i,1,2)$				$f_i^y(1,2)$
0	$01\vert 10$				0.1		$10\vert 01$				0.9
1		$11\vert 00$			0.5			$00\vert 11$			0.5
2			$10\vert 01$		0.3				$01\vert 10$		0.4
3				$00\vert 11$	0.7					$11\vert 00$	0.2

Up to now, we have considered examples where the dynamics of the fitness landscape is either externally defined and subsequently imprinted on the landscape, or is internal due to a mathematical description of the landscape that explicitly depends on time. A fundamentally different way to initiate dynamics in a fitness landscape is

Table 10.4 Example of the lookup table for the NKC landscape with $N = 4$, $K = 1$ and $C = 1$, which is of dimension $N \times 2^{K+C+1}$

$n(x_i,1,1)$	$f_0^x(1,1)$	$f_1^x(1,1)$	$f_2^x(1,1)$	$f_3^x(1,1)$	$n(y_i,1,1)$	$f_0^y(1,1)$	$f_1^y(1,1)$	$f_2^y(1,1)$	$f_3^y(1,1)$
00|0	0.6	0.3	0.5	0.1	00|0	0.6	0.3	0.5	0.1
01|0	0.1	0.5	0.9	0.4	01|0	0.1	0.5	0.9	0.4
11|0	0.8	0.2	0.1	0.3	11|0	0.8	0.5	0.1	0.5
10|0	0.5	0.8	0.7	0.7	10|0	0.6	0.8	0.9	0.7
10|1	0.6	0.3	0.2	0.7	10|1	0.6	0.3	0.2	0.7
11|1	0.7	0.9	0.5	0.6	11|1	0.7	0.9	0.5	0.6
01|1	0.3	0.7	0.2	0.6	01|1	0.1	0.7	0.1	0.6
00|1	0.5	0.3	0.6	0.4	00|1	0.5	0.3	0.6	0.9

by the individuals of a population that inhabits it. This can be done by utilizing the time dependence of the individuals' fitness and/or the individuals' phenotype (that is, a location in the search space), or by using quantities that are derived from fitness and/or location. In doing so, each individual is considered as to represent a phenotypic realization of a genotypic point in the landscape that changes with time. In other words, we use the evolutionary dynamics and let it interact with the environmental dynamics. Instead of individuals, also whole species that belong to a group of species can be considered as far as they are described by a single quantity averaging (or otherwise aggregating) the phenotypic variety of the species. Either way, for describing the evolutionary dynamics of the population (or the group of species) we need a time scale, which usually is discrete generational time. In the following we assume that there is a linear relationship between the generational time and the landscape time k and to simplify even more we set this linear relationship to equality. Nevertheless, the discussion that follows applies likewise for any unique relationship between generational time and landscape time. So, we can use the landscape time to measure generational dynamics. With this time scale we can formulate via a generation transition function ψ (see e.g. [2], p. 64–65) how a population $P(k+1)$ at generation $k+1$ originates from a population $P(k)$ at generation k:

$$P(k+1) = \psi(P(k)), k \geq 0. \tag{10.32}$$

These generational dynamics can now be linked to the fitness landscape and also be considered as to change the topology of the fitness landscape with time. In other words, the population adapts to the fitness landscape and deforms it.

There are some motivating thoughts for such a model. A first is that the individual (or the species) utilizes the abilities and features connected to the phenotypic realization while at the same time interacts and competes with other individuals of the same or other species. This in turn leads to an adaption process in the interacting parties, diminishes the species' fitness and bulges in the fitness landscape for the given phenotypic realization. As a result, fitness can only be maintained by moving in the phenotypic space, which is also called the Red Queen effect. This is a coevolutionary process where the actions of one species interact with the fitness of another species and vice versa. If both species have separated fitness landscapes they are coupled as for instance in the NKC model of the Equations (10.29) and (10.30). If we describe the actions of both species with a single shared fitness landscape, we obtain a dynamically deformable landscape. A very simple way to model such a deformation of the fitness landscapes by the individuals that populate it, is to take a static fitness landscape $f(x)$ and deduct fitness at points related to the position of the population. The most straightforward way to do so is to deform the landscape at the points where the population is in genotypic space. With λ the number of individuals in the population $P(k) = (p_1(k), p_2(k), \ldots, p_\lambda(k))$, we can define a dynamic fitness landscape that is dynamically deformed depending on the individuals' locations as

$$f(x,k) = f(x) - \frac{1}{\lambda} \sum_{i=1}^{\lambda} \alpha_i p_i(x,k), \tag{10.33}$$

where the α_i are some weighting factors. However, such a description has some serious disadvantages. It only deforms the landscape at exactly the points where the individuals are situated from generation to generation but not their vicinity. Also, the deforming process happens at exactly the moment when the individual arrives at the search space point and is not depleted at any later point in time. For these reasons it would be desirable to have some smoothening, both in the spatial and the temporal aspect. These ideas are addressed in the deformable fitness landscapes studied by Ebner and co–workers [17, 19], see also Chapter 12 of this book. These works consider fitness landscapes that are populated by several species, but this type of modelling could also be applied to individuals of a population. For the spatial smoothing, each individual $p_i(k)$ of the population $P(k)$ is surrounded by a negative Gaussian hill $-\exp\left(-(x-P(k))^T A(x-P(k))\right)$, that bulges in the landscape. The positive definite matrix A can be used to adjust the dilation of the spatial deforming. The temporal smoothing is also modeled by a Gaussian function, $\exp\left(\frac{i-k-\tau_{lat}}{2\sigma^2}\right)$, which runs over a temporal counter i and assigns that for the point in landscape time $(k+\tau_{lat})$ the spatial negative Gaussian hill deforms the landscape maximally and this effect phases out for smaller and larger times. The τ_{lat} is a latency to move the effect of the deformation forward in time and σ can be used to tune the timely deforming process. Hence, we have a time–dependent deformable fitness landscape based on a static landscape $f(x)$ as

$$f(x,k) = f(x) - \frac{1}{\alpha} \sum_{i=0}^{i_{end}} \exp\left(-\frac{(i-k-\tau_{lat})^2}{2\sigma^2}\right) \exp\left(-\frac{1}{2}(x-P(k))^T A(x-P(k))\right), \tag{10.34}$$

with $\alpha = \sigma(2\pi)^{\frac{n+1}{2}} \det(A)^{\frac{1}{2}}$ and i_{end} the final run time of the landscape. With Equation (10.34) we have a dynamic fitness landscape whose dynamics is population–based as $f(x,k+1)$ only depends on the values of the population $P(k+1)$ at this generation. This closely resembles the codynamic fitness landscape defined by Equation (10.27).

This population dynamics can follow any law as set out by Equation (10.32) and may include any genetic operator. A convenient way to generate population dynamics for experimenting with the deformable fitness landscape that is simple and computationally inexpensive is a hill climbing process, which was used in [17, 19]. In these works three types of hill climbing dynamics were considered that led to the following updating rules. All these updating rules use the landscape's gradient in order to model the effect that the population tries to escape regions where the landscape's fitness undergoes change. The first rule uses the sign of the landscape's gradient, i.e.

$$P(k+1) = \alpha\, sgn\left(\left. \frac{\partial f(x,k)}{\partial x} \right|_{x=P(k)} \right) \cdot P(k), \tag{10.35}$$

where *sgn* is the element–wise sign operator and α a weighting factor. A second rule sets the movement of the population proportional to the gradient, i.e.

$$P(k+1) = \alpha \left. \frac{\partial f(x,k)}{\partial x} \right|_{x=P(k)} \cdot P(k) \tag{10.36}$$

and a third rule integrates the gradient over time, i.e.

$$P(k+1) = \left(\sum_{i=1}^{k} \alpha_i \left. \frac{\partial f(x,i)}{\partial x} \right|_{x=P(i)} \right) \cdot P(k). \tag{10.37}$$

Detailed experiments with such deformable landscapes are given in Chapter 12 of this book.

10.4 Conclusions

10.4.1 Hierarchy of Fitness Landscapes

In this chapter we have dealt with landscapes whose fitness values change with time. It was shown that such time–dependent landscapes may occur in describing two major evolutionary processes. A first is in treating conditions apart from the genetic makeup of a population that change with time and massively influence the evolutionary outcome. Most prominently, this means that fitness of a phenotypic realization is not constant over the time frame of consideration and creates a dynamic environment in which the evolutionary process takes place. This leads to dynamic fitness landscapes and may imply solving a dynamic optimization problem. A second evolutionary process deals with situations where several populations (or at least several individuals) interact in a cooperative or competitive way and hence mutually influence fitness in a timely fashion. This leads to coevolution and creates fitness landscapes for each population that are codynamic. These landscapes are coupled via dynamic fitness evaluation and repercussions from the interdependent fitness allocation can be seen as to deform the landscapes.

In the last two sections, we have studied such landscapes and have seen that the dynamics may have three sources: internal, external and population–based. The distinction between external, internal and population–based dynamics has, apart from characterizing the source of changes in the fitness landscape, implications for the specification of the dynamics of neighboring points in the search space. For internal dynamics, the landscape's time evolution law of Equation (10.11) states how the fitness of all points in the search space and all their neighboring points change with time. In other words, the changes are explained for every search space point. With external dynamics, we only define changes of selected and characteristic features in the landscape. Here, the changes are explicitly explained for only a discrete subset of search space points. Their neighborhood may change too, but according to the same time regime as the points themselves. Population–based dynamics extends to landscapes for each population involved that mutually influence each other in a codynamic way. This can be interpreted as features of one landscape that drive other landscapes externally. However, there is no separation between that what drives and that what is driven because the process alternates between all involved landscapes and the drive is not directed to selected and pre–defined topological features of the landscapes.

To put these facts into a wider context, there has been an attempt to draw a connection to spatially extended systems and to establish a hierarchy of fitness landscapes [59], which is based on a hierarchy of spatio–temporal dynamics [14, 26] (see Table 10.5). The hierarchy comes from the different combinations of discretizing space and time in the fitness landscape. For the landscape being static, the search space can be either continuous or discrete (binary), which results in problem classes 1 and 2. If the discrete search space consists additionally of a finite number of elements, the corresponding optimization problem is a combinatorial one. Class 3 are discrete fitness functions with discrete dynamics. These dynamics can be external, as for instance the XOR generator [78], see Equation (10.17), or dynamic combinatorial optimization problems such a dynamic knapsack, dynamic royal road or dynamic bit–matching [70]. An example of internal dynamics are fitness landscapes constructed from coupled map lattices (CML). A class 3 problem with population–based drive would be a combinatorial optimization problem solved by a coevolutionary algorithm. Continuous fitness functions with discrete dynamics form class 4. Examples with external dynamics include the moving peak benchmark defined in Equation (10.12), but also other similar problems such a dynamic sphere,

Table 10.5 Hierarchy of fitness landscapes; S: static, D: discrete, C: continuous; Dynamics: N: no dynamics, I: internal, E: external, P: population–based

Class	Space	Time	Possible model	Dynamics
1	*D*	S	Discrete fitness function	N
2	*C*	S	Continuous fitness function	N
3	*D*	*D*	Discrete fitness function with discrete dynamics	I, E, P
4	*C*	*D*	Continuous fitness function with discrete dynamics	E, P
5	*D*	*C*	Lattice of coupled ordinary differential equation (ODE)	I
6	*C*	*C*	Partial differential equation (PDE)	I

dynamic Ackley, dynamic Rosenbrook etc. The coevolutionary problem considered in Section 10.3.2 as well as decomposition problems with a continuous evaluation function (see e.g. [79]) fall into class 4 with population–based dynamics. To define internal dynamics for class 4 problems would mean to have a description as in Equation (10.11) for a non–countable number of points in the fitness landscape, which is impractical for any numerical calculation. Therefore, class 4 problems with internal dynamics do not play a role in the literature. Class 5 and 6 fitness landscapes have continuous time and discrete or continuous search spaces. Possible models for such dynamic fitness landscapes are lattices of ODEs or PDEs. The mathematical formulation involves internal continuous–time dynamics. Such models are suitable for formulating general fitness landscapes and mainly pervade modeling and studying fundamental properties of evolution, but rarely to be found in studies in evolutionary computation. The reason for this might be that, as mentioned before, the discrete population dynamics of an evolutionary algorithm is best linked to fitness landscapes with discrete time characteristics. In addition, both types of mathematical description do, at least not in general, have an analytic solution. Any numerical calculation involves a discretization of time and/or space, for instance in numerical integrating ODEs and PDEs. Hence, both classes can, at least from a numerical point of view, be reduced to class 3 problems. As coevolutionary algorithm drive the codynamic landscapes by interacting populations, the resulting population–based dynamics is always discrete–time. However, codynamics also mean that the landscapes for each population are dynamical slices through an overall landscape. As a consequence, the shared static fitness landscape might, for instance, be of class 2, but the codynamic landscapes for each population are class 4.

10.4.2 Future Research Directions

Research on fitness landscapes has intensified recently and those advances are the topic of this book. Research on fitness landscapes that depend on time is an even younger field and still a considerable step away from maturity. Therefore, it comes as no surprise that some interesting questions remain unanswered yet. However, this is also connected with the hope that the present study might serve as a baseline for further development. Two directions appear to be particularly interesting. A first is linked to one of the major purposes of fitness landscapes: to provide a notion of how difficult it is for an evolutionary search algorithm to solve a given optimization problem. One way to accrue such knowledge is by landscape measures. For static measures theoretical and practical considerations have let to reliable and applicable results, see e.g. [24, 67] but also Chapters 4, 5 and 8 of this book. There are some examples of works on measures for landscapes that depend on time [21, 29, 58, 59]. However, their main focus is on applying measures of static landscapes (that is, measures for topological features) to the dynamic situation. This somehow masks the effect the dynamics has on the landscape and hence on problem hardness. It would be desirable to establish additional measures for dynamic problem hardness. A second direction is studying landscapes of specific problems and by doing so closing the gap between theory and application. Again, this is a path that has been successfully followed for static landscapes [38, 54, 71]. Hence, this approach seems to be also promising for dynamic and codynamic landscapes.

References

[1] Alba, E., Sarasola, B.: Measuring fitness degeneration in dynamic optimization problems. In: Di Chio, C., et al. (eds.) Applications of Evolutionary Computation - EvoApplications 2010, pp. 572–581. Springer, Berlin (2010)

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

[3] Bascompte, J., Jordano, P.: Plant-animal mutualistic networks: the architecture of biodiversity. Annu. Rev. Ecol. Evol. Syst. 38, 567–593 (2007)

[4] Bosman, P.A.N., La Poutré, H.: Computationally intelligent online dynamic vehicle routing by explicit load prediction in an evolutionary algorithm. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 312–321. Springer, Heidelberg (2006)

[5] Bosman, P.A.N.: Learning and anticipation in online dynamic optimization. In: Yang, S., Ong, Y.S., Jin, Y. (eds.) Evolutionary Computation in Dynamic and Uncertain Environments, pp. 129–152. Springer, Heidelberg (2007)

[6] Brabazon, A., Silva, A., de Sousa, T.F., O'Neill, M., Matthews, R., Costa, E.: A particle swarm model of organizational adaptation. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 12–23. Springer, Heidelberg (2004)

[7] Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 1999, pp. 1875–1882. IEEE Press, Piscataway (1999)

[8] Buckling, A., Rainey, P.B.: Antagonistic coevolution between a bacterium and a bacteriophage. Proc. R. Soc. Lond. B 269, 931–936 (2002)

[9] Bull, L.: Coevolutionary species adaptation genetic algorithms: a continuing SAGA on coupled fitness landscapes. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 322–331. Springer, Heidelberg (2005)

[10] Chazottes, J.R., Fernandez, B.: Dynamics of Coupled Map Lattices and of Related Spatially Extended Systems. Springer, Heidelberg (2005)

[11] Chellapilla, K., Fogel, D.B.: Evolving neural networks to play checkers without relying on expert knowledge. IEEE Trans. Neural Netw. 10, 1382–1391 (1999)

[12] Cheng, H., Yang, S.: Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 562–571. Springer, Heidelberg (2010)

[13] Cheng, H., Yang, S.: Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks. Engineering Applications of Artificial Intelligence 23, 806–819 (2010)

[14] Crutchfield, J.P., Kaneko, K.: Phenomenology of spatiotemporal chaos. In: Hao, B. (ed.) Directions in Chaos, vol. 1, pp. 272–353. World Scientific, Singapore (1987)

[15] Cruz, C., Gonzlez, J.R., Pelta, D.A.: Optimization in dynamic environments: a survey on problems, methods and measures. Soft Computing 15, 1427–1448 (2011)

[16] de Jong, E.D., Polack, J.B.: Ideal evaluation from coevolution. Evolutionary Computation 12, 159–192 (2004)

[17] Ebner, M., Watson, R.A., Alexander, J.: Co–evolutionary dynamics on a deformable landscape. In: Zalzala, A., Fonseca, C., Kim, J.H., Smith, A., Yao, X. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2000, pp. 1284–1291. IEEE Press, Piscataway (2000)

[18] Ebner, M.: Coevolution and the red queen effect shape virtual plants. Genetic Programming and Evolvable Machines 7, 103–123 (2006)

[19] Ebner, M., Watson, R.A., Alexander, J.: Coevolutionary dynamics of interacting species. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 1–10. Springer, Heidelberg (2010)

[20] Goh, C.K., Tan, K.C.: A competitive–cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Trans. Evolut. Comp. 13, 103–127 (2009)

[21] Hordijk, W., Kauffman, S.A.: Correlation analysis of coupled fitness landscapes. Complexity 10, 42–49 (2005)

[22] Jansen, T., Wiegand, R.P.: The cooperative coevolutionary (1+1) EA. Evolutionary Computation 12, 405–434 (2004)

[23] Jones, T.: Evolutionary algorithms, fitness landscape and search. PhD thesis, The University of New Mexico, Albuquerque (1995), `http://www.santafe.edu/media/workingpapers/95-05-048.pdf` (retrieved November 11, 2012)

[24] Kallel, L., Naudts, B., Reeves, C.R.: Properties of fitness functions and search landscapes. In: Kallel, L., Naudts, B., Rogers, A. (eds.) Theoretical Aspects of Evolutionary Computing, pp. 177–208. Springer, Heidelberg (2001)

[25] Kaneko, K.: The coupled map lattice. In: Kaneko, K. (ed.) Theory and Application of Coupled Map Lattices, pp. 1–49. John Wiley, Chichester (1993)

[26] Kaneko, K., Tsuda, I.: Complex Systems: Chaos and Beyond. Springer, Heidelberg (2001)

[27] Kardar, M., Parisi, G., Zhang, Y.C.: Dynamic scaling of growing interfaces. Phys. Rev. Lett. 56, 889–892 (1986)

[28] Kashtan, N., Noor, E., Alon, U.: Varying environments can speed up evolution. Proc. Natl. Acad Sci USA (PNAS) 104, 13711–13716 (2007)

[29] Katada, Y., Handa, Y.: Tracking the Red Queen effect by estimating features of competitive co–evolutionary fitness landscapes. In: Fogel, D.B. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2010, pp. 4417–4424. IEEE Press, Piscataway (2010)

[30] Katzav, E., Cugliandolo, L.F.: From coupled map lattices to the stochastic Kardar–Parisi–Zhang equation. Physica A371, 96–99 (2006)

[31] Kauffman, S.A., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. J. Theor. Biol 128, 11–45 (1987)

[32] Kauffman, S.A.: The Origin of Order: Self–Organization and Selection in Evolution. Oxford University Press, New York (1993)

[33] Kauffman, S.A., Weinberger, E.D.: The NK Model of rugged fitness landscapes and its application to the maturation of the immune response. J. Theor. Biol. 141, 211–245 (1989)

[34] Kauffman, S.A., Johnsen, S.: Coevolution to the edge of chaos: Coupled fitness landscapes, poised states, and coevolutionary avalanches. J. Theor. Biol. 149, 467–505 (1991)

[35] Lin, S.C., Goodman, E.D., Punch, W.F.: A genetic algorithm approach to dynamic job shop scheduling problems. In: Bäck, T. (ed.) Proc. Seventh International Conference on Genetic Algorithms, pp. 481–488. Morgan Kaufmann, San Francisco (1997)

[36] Ma, K., Jianga, J., Yanga, C.B.: Scaling behavior of roughness in the two–dimensional Kardar–Parisi–Zhang growth. Physica 378, 194–200 (2007)

[37] Mendes, R., Mohais, A.: DynDE: Differential Evolution for dynamic optimization problems. In: Corne, D. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2005, pp. 2808–2815. IEEE Press, Piscataway (2005)

[38] Merz, P.: Advanced fitness landscape analysis and the performance of memetic algorithms. Evolutionary Computation 12, 303–325 (2004)

[39] Meyers, L.A., Bull, J.J.: Fighting change with change: adaptive variation in an uncertain world. Trends in Ecology & Evolution 17, 551–557 (2002)

[40] Miranda, V.G., Aarão Reis, F.D.A.: Numerical study of the Kardar–Parisi–Zhang equation. Phys. Rev. 77, 031134–1–6 (2008)

[41] Morrison, R.W., De Jong, K.A.: A test problem generator for non–stationary environments. In: Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 1999, pp. 2047–2053. IEEE Press, Piscataway (1999)

[42] Morrison, R.W., De Jong, K.A.: Triggered hypermutation revisited. In: Zalzala, A., et al. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2000, pp. 1025–1032. IEEE Press, Piscataway (2000)

[43] Nguyen, T.T., Yao, X.: Benchmarking and solving dynamic constrained problems. In: Tyrrell, A. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2009, pp. 690–697. IEEE Press, Piscataway (2009)

[44] Nguyen, T.T., Yang, S., Branke, J.: Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation 6, 1–24 (2012)

[45] Oliehoek, F.A., de Jong, E.D., Vlassis, N.A.: The parallel Nash memory for asymmetric games. In: Cattolico, M. (ed.) Proc. Genetic and Evolutionary Computation Conference, GECCO 2006, pp. 337–344. ACM Press, New York (2006)

[46] Panait, L., Luke, S.: Time–dependent collaboration schemes for cooperative coevolutionary algorithms. In: Potter, M.A., Wiegand, R.P. (eds.) 2005 AAAI Fall Symposium on Coevolutionary and Coevolving Systems. AAAI Press, Palo Alto (2005)

[47] Panait, L., Luke, S.: Selecting informative actions improves cooperative multiagent learning. In: Nakashima, H., Wellman, M.P., Weiss, G., Stone, P. (eds.) Proc. Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2006), pp. 760–766. ACM Press, New York (2006)

[48] Panait, L., Luke, S., Harrison, J.: Archive–based cooperative coevolutionary algorithms. In: Cattolico, M. (ed.) Proc. Genetic and Evolutionary Computation Conference, GECCO 2006, pp. 345–352. ACM Press, New York (2006)

[49] Popovici, E., Bucci, A., Wiegand, R.P., de Jong, E.D.: Coevolutionary principles. In: Rozenberg, G., Bäck, T., Kok, J.N. (eds.) Handbook of Natural Computing, pp. 987–1033. Springer, Heidelberg (2010)

[50] Popovici, E., de Jong, K.A.: Understanding competitive co–evolutionary dynamics via fitness landscapes. In: Luke, S. (ed.) 2004 AAAI Fall Symposium on Artificial Multiagent Learning. AAAI Press, Palo Alto (2005)

[51] Popovici, E., de Jong, K.A.: Understanding cooperative co–evolutionary dynamics via simple fitness landscapes. In: Beyer, H.G., O'Reilly, U.M. (eds.) Proc. Genetic and Evolutionary Computation Conference, GECCO 2005, pp. 507–514. Morgan Kaufmann, San Francisco (2005)

[52] Popovici, E., de Jong, K.A.: The dynamics of the best individuals in co–evolution. Natural Computing 5, 229–255 (2006)

[53] Potter, M.A., de Jong, K.A.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation 8, 1–29 (2000)

[54] Prügel–Bennett, A., Tayarani–Najaran, M.H.: Maximum satisfiability: Anatomy of the fitness landscape for a hard combinatorial optimization problem. IEEE Trans. Evolut. Comp. 16, 319–338 (2012)

[55] Richter, H.: Behavior of evolutionary algorithms in chaotically changing fitness landscapes. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 111–120. Springer, Heidelberg (2004)

[56] Richter, H.: A study of dynamic severity in chaotic fitness landscapes. In: Corne, D. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2005, pp. 2824–2831. IEEE Press, Piscataway (2005)

[57] Richter, H.: Evolutionary optimization in spatio–temporal fitness landscapes. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 1–10. Springer, Heidelberg (2006)

[58] Richter, H.: Coupled map lattices as spatio–temporal fitness functions: Landscape measures and evolutionary optimization. Physica 237, 167–186 (2008)

[59] Richter, H.: Evolutionary optimization and dynamic fitness landscapes: From reaction-diffusion systems to chaotic CML. In: Zelinka, I., Celikovsky, S., Richter, H., Chen, G. (eds.) Evolutionary Algorithms and Chaotic Systems. SCI, vol. 267, pp. 409–446. Springer, Heidelberg (2010)

[60] Richter, H.: Memory design for constrained dynamic optimization problems. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 552–561. Springer, Heidelberg (2010)

[61] Richter, H., Dietel, F.: Solving dynamic constrained optimization problems with asynchronous change pattern. In: Di Chio, C., et al. (eds.) EvoApplications 2011, Part I. LNCS, vol. 6624, pp. 334–343. Springer, Heidelberg (2011)

[62] Richter, H., Yang, S.: Memory based on abstraction for dynamic fitness functions. In: Giacobini, M., et al. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 596–605. Springer, Heidelberg (2008)

[63] Richter, H., Yang, S.: Learning behavior in abstract memory schemes for dynamic optimization problems. Soft Computing 13, 1163–1173 (2009)

[64] Rosin, C.D., Belew, R.K.: New methods for competitive coevolution. Evolutionary Computation 5, 1–29 (1997)

[65] Simões, A., Costa, E.: Variable–size memory evolutionary algorithm to deal with dynamic environments. In: Giacobini, M., et al. (eds.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 617–626. Springer, Heidelberg (2007)

[66] Simões, A., Costa, E.: Evolutionary algorithms for dynamic environments: Prediction using linear regression and Markov chains. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN X 2008. LNCS, vol. 5199, pp. 306–315. Springer, Heidelberg (2008)

[67] Smith, T., Husbands, P., Layzell, P., O'Shea, M.: Fitness landscapes and evolvability. Evolutionary Computation 10, 1–34 (2002)

[68] Stadler, P.F., Stephens, C.R.: Landscapes and effective fitness. Comm. Theor. Biol. 8, 389–431 (2003)

[69] Stadler, B.M.R., Stadler, P.F., Wagner, G.P., Fontana, W.: The topology of the possible: Formal spaces underlying patterns of evolutionary change. J. Theor. Biol. 213, 241–274 (2001)

[70] Stanhope, S.A., Daida, J.M.: (1+1) Genetic algorithm fitness dynamics in a changing environment. In: Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 1999, pp. 1851–1858. IEEE Press, Piscataway (1999)

[71] Tavares, J., Pereira, F.B., Costa, E.: Multidimensional knapsack problem: a fitness landscape analysis. IEEE Trans. Sys. Man Cyber. B 38, 604–616 (2008)

[72] Tinós, R., Yang, S.: A self–organizing random immigrants genetic algorithm for dynamic optimization problems. Genetic Programming and Evolvable Machines 8, 255–286 (2007)

[73] van Hemert, J., La Poutré, J.A.H.: Dynamic routing problems with fruitful regions: Models and evolutionary computation. In: Yao, X., et al. (eds.) PPSN VIII 2004. LNCS, vol. 3242, pp. 692–701. Springer, Heidelberg (2004)

[74] Watson, R.A., Pollack, J.B.: Coevolutionary dynamics in a minimal substrate. In: Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E. (eds.) Proc. Genetic and Evolutionary Computation Conference, GECCO 2001, pp. 702–709. Morgan Kaufmann, San Francisco (2001)

[75] Weicker, K.: Performance measures for dynamic environments. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN VII 2002. LNCS, vol. 2439, pp. 64–73. Springer, Heidelberg (2002)

[76] Wilke, C.O., Martinetz, T.: Adaptive walks on time-dependent fitness landscapes. Phys. Rev. E60, 2154–2159 (1999)

[77] Yang, S.: Non–stationary problem optimization using the primal-dual genetic algorithm. In: Sarker, R., Reynolds, R., Abbass, H., Tan, K.C., Essam, D., McKay, R., Gedeon, T. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2003, pp. 2246–2253. IEEE Press, Piscataway (2003)

[78] Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft Computing 9, 815–834 (2005)

[79] Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Information Science 178, 2985–2999 (2008)

Chapter 11
Red Queen Coevolution on Fitness Landscapes

Ricard V. Solé and Josep Sardanyés

Abstract. Species do not merely evolve, they also coevolve with other organisms. Coevolution is a major force driving interacting species to continuously evolve exploring their fitness landscapes. Coevolution involves the coupling of species fitness landscapes, linking species genetic changes with their inter-specific ecological interactions. Here we first introduce the Red Queen hypothesis of evolution commenting on some theoretical aspects and empirical evidences. As an introduction to the fitness landscape concept, we review key issues on evolution on simple and rugged fitness landscapes. Then we present key modeling examples of coevolution on different fitness landscapes at different scales, from RNA viruses to complex ecosystems and macroevolution.

11.1 Introduction: The Red Queen

Coevolution pervades evolutionary change on multiple scales. It is not exaggerated to say, changing a little the classical Dobzhansky statement, that nothing makes sense in biology except in the light of coevolution. Darwin himself recognized this when referring to what he called the entangled bank [9]: "It is interesting to contemplate an entangled bank, clothed with many plants of many kinds, with birds singing on the bushes, with various insects flitting about, and with worms crawling through the damp earth". Indeed, ecosystems need to be seen as collectives of interacting

Ricard V. Solé · Josep Sardanyés
ICREA-Complex Systems Lab, Universitat Pompeu Fabra,
Parc de Recerca Biomèdica de Barcelona (PRBB), Dr. Aiguader 88,
08003 Barcelona, Spain
Institut de Biologia Evolutiva (CSIC-Universitat Pompeu Fabra),
Passeig Marítim de la Barceloneta 37, 08003 Barcelona, Spain
e-mail: {ricard.sole,josep.sardanes}@upf.edu

Ricard V. Solé
Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87501, USA

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_11, © Springer-Verlag Berlin Heidelberg 2014

species whose evolutionary fate is necessarily intermingled in complex ways. Such complex networks pervade ecosystems and their evolutionary dynamics [48].

The rationale of the previous statements is simple. As any species changes over time, it inevitably triggers co-evolutionary responses in those partners directly affected by their interdependencies. A prey running faster than its predator will require changes later to cope with the change. Running fast is one option, hiding in the appropriate place another. If we move ourselves into the microcosmos of host-pathogen interactions, including cells and viruses (or bacteria) as main examples of coevolving players, a faster rate of change in the parasite might need a faster response from the host. Changing attributes might be an unavoidable consequence of entangled ecosystems and not changing might possibly be lethal. This view matches a dynamically unstable scenario where changes keep happening all the time and, as in Lewis Carroll's *Through the looking glass*, species need – as Alice does – to *constantly run to remain in the same place*. Such picture was early supported by Leigh Van Valen's work, and is known as the Red Queen hypothesis.

The Red Queen model was introduced by Leigh Van Valen in 1973 [87]. It was conceived as a theoretical explanation for the observation that the extinction probability of a species is approximately independent of its length of existence [4, 87]. Accordingly with this view, Van Valen observed that the vast majority of taxonomic groups analyzed displayed exponentially decaying survivorship curves. This result implied constancy in the probability of extinction of the taxa, regardless of their previous duration. That is, both data from the fossil record and from extant species suggested that a given species may disappear at any time, irrespective of how long it has already existed. This unexpected phenomenon, termed the *Law of Constant Extinction*, can be formulated, in a simple way, as follows: If $N(t)$ indicates the number of species at a given time and we follow their presence over time (ignoring other events) we would observe an exponential decay law, namely:

$$\frac{dN}{dt} = -\delta(t)N,$$

where $\delta(t)$ indicates a time-dependent extinction rate. If N_0 is the original number, this differential equation is easily solved, and gives:

$$N(t) = N_0 \exp\left[-\int_0^t \delta(t)dt\right].$$

Despite the seemingly obvious assumption that δ depends on t, the surprising observation is that the observed curves fit very well a constant decay rate δ, i.e., a solution:

$$N(t) = N_0 e^{-\delta t},$$

where δ is the extinction probability of a species (per millions of years, Myr). This law is essentially correct on average, despite the fine-scale pattern being much more episodic [56], as depicted in Figure 11.1(a). Here we display the surviving sets of families found in the marine fossil record (the so called *seudocohorts*) through time.

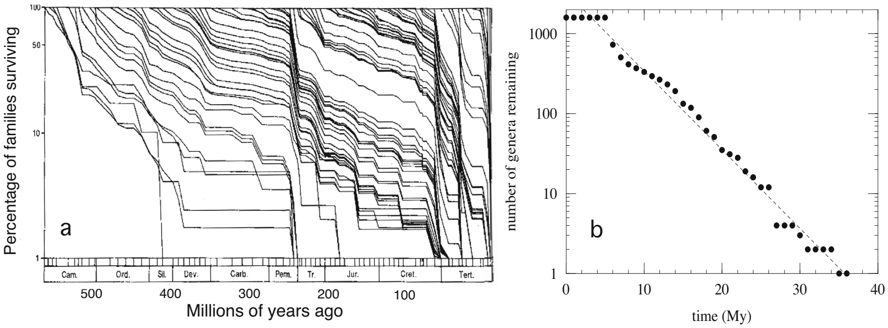


Fig. 11.1 Extinction and the fossil record of life. (a) Successive cohort survivorship curves for 2,316 extinct marine families during the Phanerozoic (redrawn after [57]). Notice that, together with an average exponential tendency to decline (which would give straight lines in a linear-log plot), discrete and punctuated events are found in the dynamics of survivorship. This pattern is well appreciated in (b) where we display the number of ammonoid genera surviving over geologic time.

A roughly exponential decay can be identified, together with sharp extinction events (see also Figure 11.1(b)).

Our intuition, guided by Darwin's theory of natural selection, would have expected species within any group to become longer lived along time: if adaptation improves species progressively through time, a decreasing probability of extinction should be expected. That is, older species should last longer. However, careful analysis revealed that species of modern mammals are likely to become extinct as were their ancestors living 200 Myr ago [4]. If evolution leads to improvement through adaptation, why do modern mammals have the same extinction probabilities as their ancestors? Van Valen's interpretation is simple but counterintuitive: species do not evolve to become any better at avoiding extinction. Van Valen suggested that a constant extinction probability would arise in an always changing biotic community, with species continually adapting to each other's changes. The name for his conjecture alludes to the Red Queen's remark in Lewis Carroll's *Alice Through the Looking Glass*: "here, you see, it takes all the running you can do, to keep in the same place". Van Valen's view of evolution is that species change just to remain in the evolutionary game and extinction occurs when no further changes are possible. Actually, the Red Queen hypothesis is profoundly Darwinian, in that it puts emphasis mostly on biotic interactions rather than on abiotic factors [29]. Van Valen further elaborated this concept in much more detail in subsequent articles, showing that his theory was compatible with the classic population-genetic view of species evolution [85, 86].

To test the plausibility of the Red Queen hypothesis, Maynard Smith and coworkers (see [79] and references therein) developed a theoretical model describing continuous (co-)evolution of species in a constant environment. Such model considered a fixed number of S interacting species, defining some fitness measure ϕ, and a maximum fitness ϕ_i^* was supposed to exist for each species in a given fixed, external biotic environment. At a given time, the fitness ϕ_i and the maximum ϕ_i^* took different values, and each species "tried" to reduce the so called *lag load*, defined as:

$$L_i = \frac{\phi_i - \phi_i^*}{\phi_i}; \quad i = 1, ..., S.$$

If β_{ij} is the change in the lag load L_i due to a change in L_j, then a mean-field equation for the average lag load $\langle L \rangle = \sum_i (L_i)/S$ can be derived. This is done by first separating, for each species, changes due to "microevolution of coexisting species" from those linked with its own microevolution [79]. The entire equation for the lag load variation in a given species is $\delta L_i = \delta_c L_i - \delta_g L_i$, which simply says that it typically increases due to changes in the other species and decreases due to microevolutionary changes in the species under consideration, i.e., $\delta_c L_i$ is the increase in the lag of the ith species caused by evolutionary changes in others, and $\delta_g L_i$ is the reduction in lag caused by changes in species i itself. This can be written in the following way,

$$\delta L_i = \sum_{j=1}^{S} \beta_{ij} \delta_g L_j - \delta_g L_i,$$

where β_{ij} (with $\beta_{ii} = 0$) is the increase in L_i due to a (unit) change in L_j. Assuming that most species are close to their adaptive peaks, any evolutionary change in one species will have a deleterious effect on the other species. The deterministic, time-continuous equivalent model can be formulated with:

$$\frac{dL_i}{dt} = \sum_{j=1}^{S} \beta_{ij} k_j L_j - k_i L_i. \tag{11.1}$$

By taking the average in both sides of Equation (11.1), the evolution of the average lag load is given by:

$$\frac{d\langle L \rangle}{dt} = \frac{1}{S} \sum_{i=1}^{S} \left\{ \sum_{j=1}^{S} \beta_{ij} k_j L_j - k_i L_i \right\}.$$

Assuming now that $k_i = k$ for all $i = 1, ..., S$, the average lag load equation can be written as:

$$\frac{d\langle L \rangle}{dt} = \frac{k}{S} \sum_{j=1}^{S} (\Psi_j - 1) L_j,$$

and it has a steady state solution if $\Psi_j = 1$ for all $j = 1, ..., S$. In other words, if:

$$\Gamma \equiv \sum_{i=1}^{S} \beta_{ij} = 1; \quad \forall\, j.$$

Otherwise, it can be shown that $\langle L \rangle$ will decrease (increase) for $\Gamma < 1$ ($\Gamma > 1$). The previous identity is telling us that the equilibrium state of this system is reached through a balance between the reduction of the individual lag load of each species and the increases due to coevolutionary changes in the remaining partners. The main result of this model is that evolution of species proceeds at an approximately steady rate even in the absence of external or environmental changes [79]. At the end of this chapter we will present a dynamic model of Red Queen dynamics where evolving networks interactions are made explicit.

As we previously discussed, the Red Queen hypothesis provides a plausible explanation of the fossil data record, but it turned to have more implications. For instance, one suggested implication of the Red Queen hypothesis is that coevolving pathogens may facilitate the persistence of outcrossing despite its costs. Specifically, coevolutionary interactions between hosts and pathogens might generate ever-changing conditions and thus favor the long-term maintenance of outcrossing relative to self-fertilization [1] or asexual reproduction [26, 35] (see also [27] for a review). Outcrossing (mating between different individuals) involves the introduction of unrelated genetic material into a breeding line, thus increasing genetic diversity. The previous statements are supported by evidence from nature. For instance, experimental studies on the coevolution of a nematode with a bacterial pathogen [49] revealed that the action of parasites caused an increase of outcrossing in mixed mating populations. Interestingly, these experiments also revealed that coevolution with the pathogen caused extinction in populations without outcrossing, whereas outcrossing populations persisted through reciprocal coevolution. Studies in natural snail populations also revealed that sexual reproduction is more common when parasites are abundant and adapted to infect local host populations [40, 43]. Coevolution and Red Queen dynamics were also identified for the crustacean genus *Daphnia* and its parasites in pond sediments [10].

11.2 Red Queen on a Lattice: A Toy Model

Before we get into the more formal approaches taken to describe and simulate the evolution and coevolution of species on fitness landscapes, consider a simple toy model that illustrates the basic idea behind van Valen's metaphor. Imagine a world where our species can move on the surface of a sphere. To makes things simpler, consider a discretized surface, like a mesh covering the sphere[1].

To simulate such a system we used the so-called cellular automata (CA) models [32]. CA models are a common tool to investigate interacting agents in a physical space, which, for our case, will be a surface. Each point in this mesh is a site, which can be either empty or instead occupied by an individual of a given type. Let us start with a simple "ecosystem" formed by a species exhibiting two phenotypic traits. Let us indicate as $\Sigma = \{0, 1\}$ the two possible "genotypes" which can be understood as two alleles of a given gene.

In our idealized model, 0 and 1 are the only two genotypes, each one associated with a set of parameters defining the underlying phenotype. For simplicity, we

[1] Specifically, we start from a lattice, whose surface has been discretized using a mesh, and then we perform a projection of this mesh on a surface by properly deforming the initial coordinates using a so called icosahedron-based pixelization. For details, see [80].

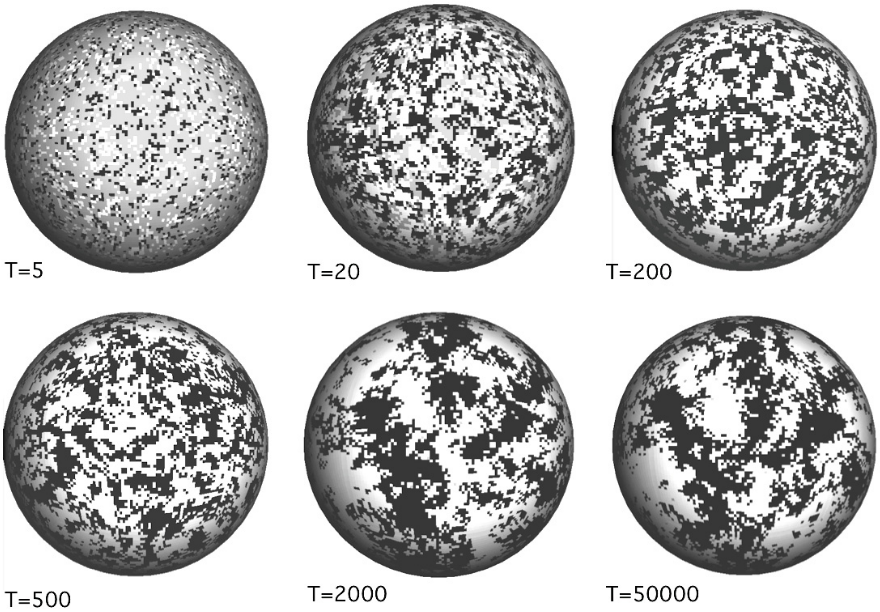


Fig. 11.2 Spatial competition dynamics in a single-species model with two mutants. Starting from a small set of occupied sites with 0 and 1 genotypes (white and black, respectively) scattered on an empty (gray) landscape, we display spatial patterns for different simulation times, T. After a few steps, local populations start to grow, but competition is still weak. Once the two populations occupy enough space, competition starts to be effective but no global exclusion occurs. Instead, the two populations coexist by expanding locally over spatial domains that appear homogeneous. After a long time, the system is rather stable. Although structures keep changing their boundaries, the global picture remains the same, with a spatial landscape displaying large homogeneous patches. Here we used $\mu = 10^{-4}$, $\delta_H = 10^{-3}$ and $r_H = 0.35$.

will consider completely symmetric sets. In other words, we have a neutral change when moving from 0 to 1 and vice versa. These transitions occur proportionally to mutation probability $\mu \in [0,1]$.

We can use the previous system to simulate host-parasite interactions with matching alleles (MA, see Section 11.4 for further details) interactions. Our CA is thus given by a two-dimensional state space, $\Omega(i,j)$, with spatial coordinates (i,j). The states, S, of the automaton at time t are given by $S(i,j;t) \in \Sigma = \{H_k, P_k, E\}$, where H and P denote, respectively, hosts and parasites defined as 1-bit strings (i.e., with $k = 0,1$). E indicates empty sites in the state space Ω.

CA models include dynamics by means of the state-transition rules, which determine the changes of the states according to the current states in a given site together with its neighboring states. For this particular system we used the so-called Moore neighborhood, which considers the eight nearest neighbors. The model rules are given in the following.

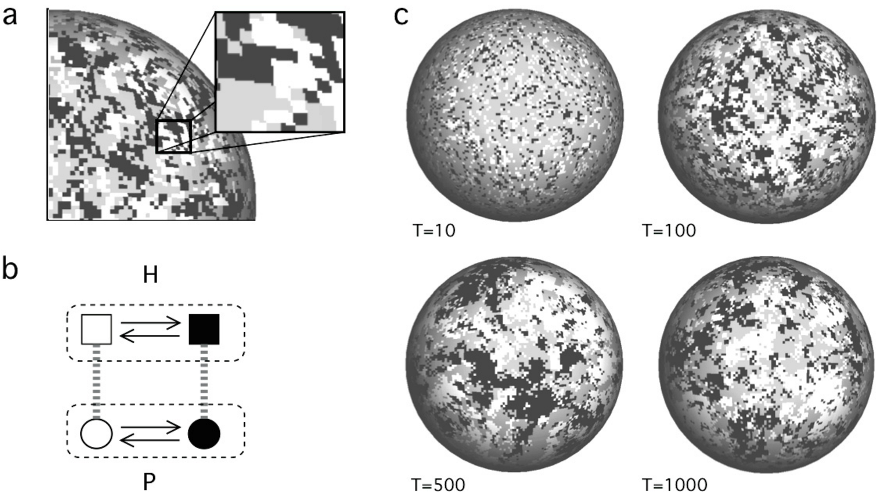


Fig. 11.3 Red Queen dynamics in host-parasite (prey-predator) coevolution. Here we have added to the only-host system displayed in the previous figure an evolving 1-bit parasite. Parasites propagate through space provided they find the same host genotype. Here two levels need to be considered: the spatial framework (a) defined by the local arrangement of individuals and the required genotypic matching. (b) Schematic diagram of the two populations (upper and lower layers) with back and forth mutations (arrows) between genotypes and the requirement of allele matching (vertical lines). The plots in (c) show the host populations (as before) which now keep changing. Similar plots would be observed (although having less dense patches) for parasites. Here we used, as before, $\mu_H = 10^{-4}$, $\delta_H = 10^{-3}$, $r_H = 0.35$; and $\mu_P = 10^{-4}$, $\delta_P = 0.2$, and $r_P = 0.1$.

State transition rules of the CA model for host-parasite matching allele interactions with 1-bit genotypes

(1) **Death:** hosts and parasites decay with probabilities δ_H and δ_P respectively, according to (recall $k = 0, 1$):

$$H_k \xrightarrow{\delta_H} E,$$

$$P_k \xrightarrow{\delta_P} E.$$

(2) **Birth of new hosts:** hosts can replicate with probability r_H, without and with mutations, following the reactions:

$$H_k + E \xrightarrow{r_H(1-\mu_H)} 2H_k.$$

$$H_k \xrightarrow{r_H \mu_H} H_k + H_{1-k},$$

where μ_H is hosts mutation rate.

(3) **Predation:** Predator genotypes predate on hosts same genotype (assuming a perfect MA interaction), with rate r_P. In other words, they can reproduce only under the

presence of the same host genotype in the neighborhood. If such condition is met, the new reactions follow:

$$P_k + H_k \xrightarrow{r_P(1-\mu_P)} 2P_k,$$

$$P_k + H_k \xrightarrow{r_P\mu_P} P_k + P_{1-k},$$

introducing again mutational changes proportionally to parasites mutation rate μ_P. Because of the H-P interaction, hosts experience a parasite-driven mortality.

What is the dynamics of this simple model when no parasites are present? In Figure 11.2 we see an example of the time evolution of this model, starting from an initial condition where we randomly scatter a small population of each genotype over the sphere. The empty, available space is indicated in gray, whereas the two alternative genotypes are shown as black and white squares. After a short transient, where both variants expand with no special constrains, available sites become scarce and the expanding patches grow and develop rugged boundaries. Such pattern stabilizes in the long run, where we observe large domains of each class. This phenomenon is due to the local exclusion of our identical competitors [71] that allows global coexistence to occur.

The previous scenario shows that competing populations end up in a predictable community structure with no further (global) changes. However, when an additional component – parasites (or predators) – is added in the same system, it immediately triggers the emergence of an unstable dynamical state. This is shown in another example in Figure 11.3, where we again represent the host populations, now starting with the same condition as in Figure 11.2 but adding also some randomly distributed predators.

At any step, the spatial distribution changes rapidly and complex waves of expansion and contraction, affecting both genotypes, are observable. Sometimes, the extinction of the parasites returns our system to the previous conditions without parasite (and a spatial segregation pattern). This occurs for example when the mutation rate of the parasite is too small or its death rate too high. Sometimes, the pressure of the parasites is so strong that they cause the extinction of the hosts and the eventual collapse of the whole host-parasite system. Interesting dynamics occur when parasites are able to reliably match their preys and reproduce at a reasonable pace. Similarly, mutation rates need to be high enough to react to depleted host populations and at the same time allow for a conservation of genetic information, thus avoiding undesirable drift (see Section 11.5.1). In other words, intermediate rates of parasite pressure end up in Red Queen evolution.

Here the system constantly changes as a consequence of the hide-and-run effect induced by the parasitic species. Each time a parasite finds a suitable host sharing the same bit, it replicates and will keep expanding provided that it finds additional hosts in the neighborhood. The end result is a spatial pattern of propagating patches and constantly changing distributions of the two available genotypes.

11.3 Fitness Landscapes

Fitness landscapes are a very useful tool and an important concept in evolutionary biology. They are used to map, represent or visualize the relationship between genotypes (or phenotypes) and reproductive success or fitness. Fitness landscapes were first introduced by Sewall Wright and were then extended by other authors (see [38, 54] and references therein). Fitness landscapes assume that each genotype has a well-defined fitness value, which is represented with a height or peak in the landscape (see Figures 11.5 and 11.8). A landscape simply means a single-valued scalar function, $f(x)$, of the state or configuration x of a system. The variable x typically has very many dimensions, and thus may be often written like a multidimensional vector, as a set of N components $x = (x_1, x_2, ..., x_N)$. The term landscape is inspired from the geographic landscapes in which the height h above sea level is a simple function $h = f(x_1, x_2)$, of the two-dimensional location $x = (x_1, x_2)$. In the field of biology, fitness landscapes are generically representing the fitness of a given biological entity as a function of its genotype or phenotype. As biological entity we refer to a given organism or to a particular macromolecule or cell of that individual.

Fitness is a relative measure, since it may depend on the environment and on other interacting organisms [54]. Fitness can be given by several properties, or by a combination of them. For instance, we can use replication or reproduction success as a measure of fitness. Properties like infectivity, migration capacity, ability to cooperate, among others, can also define a fitness which may facilitate survival and adaptation. As Jacob [33, 34] stated, adaptation typically progresses through small changes involving a local search in the space of possibilities (e.g., sequences space). The paradigm is a hill-climbing process via fitter mutants which ”move” towards a global or local optimum in the fitness landscape (see Figure 11.4(A)). The hill-climbing framework was originally proposed by Wright [90, 91], who introduced the concept of space of possible genotypes. Under this framework, each genotype has a given fitness, which belongs to a distribution of fitness values over the space of genotypes in the fitness landscape. Depending upon the distribution of fitness values, the fitness landscape will become more or less mountainous. The behavior of an adapting population will depend on how rugged the fitness landscape is, on the size of the population, and on the mutation rate which moves a population from one genotype to another in sequence space. The motion of a population over a fitness landscape also depends on whether the population reproduces asexually or sexually. The latter reproduction involves mixing of genotypes which can involve reaching more distant points in fitness landscape, compared to asexual reproduction [38].

The fitness landscape concept has been widely used in both evolutionary and coevolutionary biology. Organisms do not merely evolve, they coevolve with other organisms. As a difference from evolution, that can be roughly characterized as an adaptive search on a “potential surface”, in coevolution there may typically be no such potential surface, the process being far more complex [38]. Actually, in coevolutionary processes the fitness landscape of one organism can deform and heave as the other organisms make their own adaptive moves. Under this perspective, one can interpret coevolution as both dynamical and evolutionary processes occurring in coupled fitness landscapes (see Figures 11.3(b), 11.4, 11.5, 11.8 and 11.10(a)).

For the sake of clarity, before accounting for coevolution, we will introduce, in Sections 11.3.1 and 11.3.2 some information about evolution on fitness landscapes. First, we will describe evolutionary phenomena in simple fitness landscapes, and also present a theoretical body to model evolution in these types of landscapes. Then, we will extend our explanations to more complex, rugged fitness landscapes. From Section 11.3.3 onwards (together with the introduction section above) we will strictly focus on coevolutionary phenomena. As an example of coevolution at small scales, Section 11.3.4 includes the view of RNA virus evolution from the perspective of the Red Queen hypothesis. The remaining sections will deal with some examples and models in higher biological scales, from complex ecosystems to macroevolution.

11.3.1 Simple versus Coupled Fitness Landscapes

Models on evolution have considered different theoretical and computational frameworks to characterize several levels of complexity. One of the most successful approaches to address evolutionary phenomena (as well as coevolution as we will discuss later) on fitness landscapes is given by the digital genomes approach [21]. Under this approach, using as an example the evolution of RNA genomes, we can develop a mapping between RNA sequences (defined as a chain of nucleotides involving a four-letter alphabet Ω) and binary sequences, according to:

$$\mathscr{F} : \Omega = \{U,G,A,C\} \longrightarrow \Sigma = \{0,1\}.$$

Alternatively, one can use another Boolean representation using spins instead of bits:

$$\mathscr{F}_{\int} : \Omega = \{U,G,A,C\} \longrightarrow \Sigma = \{+1,-1\}.$$

Both approaches are equivalent because the mapping has the same nature. However, the spins approach exploits some advantages of considering ”up” and ”down” configurations to describe the microscopic dynamics.

Let us define the ith string of the population, $\mathbf{S}_i = (S_{i1},...,S_{i\nu})$, as digital genomes (i.e., sequences of purines and pyrimidines only incorporating the linear information encoded by the string) of length ν. In order to determine the functional relevance of these sequences, we need to map them to a sequence-fitness measure, which can be defined as:

$$f : \mathbf{S}_i \in \Sigma^\nu \longrightarrow f(\mathbf{S}_i).$$

This functional relation can be generically divided into two types: (i) the different bits of a string play an independent role in fitness; or (ii) some bits of the string influence others in a nontrivial way. Case (ii) corresponds to what is known as epistasis. Epistasis occurs when the phenotypic effect of a mutation depends on the presence of other mutations in the genome. Epistasis becomes especially important in highly-compacted genomes that are expected to contain multifunctional proteins or overlapping genes. In this sense, epistasis has been studied and characterized for

RNA viruses, both experimentally and theoretically (see [18] for a review and [64, 65]). In a more general way, epistatic interactions play an important role in evolutionary genetic systems almost whenever multi locus genetics matters and plays a central role in the evolution of genetic systems such as sex and recombination, ploidy, genomic segmentation and modularity, genetic incompatibility and speciation, mechanisms of mutational robustness, mutational load for deleterious mutations through genetic drift, and the rate of adaptive evolution [12].

The digital genomes approach allows us to use an abstract, multidimensional representation of the potential set of states accessible to a ν-bits digital genome. This set or space is given by a sequence space in the form of a hypercube, $\mathscr{H}^{\nu} = \Sigma^{\nu}$, which can provide, at low dimensions, some intuitions about the behavior of strings under selection-mutation pressures (see Figure 11.4). If only small mutation rates are considered, transitions between sequences will take place only involving nearest neighbors in sequence space, thus differing only in one bit.

In general, for a given mutation rate, μ, two sequences $\mathbf{S}$ and $\mathbf{S}'$ will be obtained from each other with a given probability, given by:

$$W_{\mu}(\mathbf{S} \to \mathbf{S}') = \mu^{d_H(\mathbf{S},\mathbf{S}')}(1-\mu)^{\nu - d_H(\mathbf{S},\mathbf{S}')},$$

where $d_H(\mathbf{S},\mathbf{S}')$ is the Hamming distance among the two sequences (i. e., the number of different bits), with:

$$d_H(\mathbf{S},\mathbf{S}') = \sum_{\mathbf{i=1}}^{\nu} \left(\mathbf{1} - \delta_{\mathbf{S_i},\mathbf{S'_i}}\right),$$

where $\delta_{i,j}$ is Kronecker's delta with $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ if $i \neq j$. Here W_{μ} can be interpreted in probabilistic terms: it is the probability of having exactly d_H differences between the two digital genomes. This function allows introduction of the dynamics associated with mutations as transition probabilities. For the spin mapping, the transition probabilities can be expressed as:

$$W_{\mu}(\mathbf{S} \to \mathbf{S}') = \mathscr{N} \exp\left(-\beta \sum_{\mathrm{i}=1}^{\nu} \mathrm{S_i S'_i}\right),$$

where the β term, defined as $\beta = \log(\mu/(1-\mu))/2$, can be interpreted in terms of a temperature, with $\mathscr{N}$ being a normalization constant. Once we define the fitness function associated to each vertex of the hypercube, we can characterize the dynamics.

If $N(\mathbf{S},t)$ indicates the fraction of strings having a given sequence $\mathbf{S} \in \Sigma^{\nu}$ at time t, Eigen's formulation [17] describes the population dynamics as a set of nonlinear differential equations, given by:

$$\frac{dN(\mathbf{S},t)}{dt} = \sum_{\mathbf{S}'} W_{\mu}(\mathbf{S}' \to \mathbf{S}) f(\mathbf{S}') N(\mathbf{S}',t) - \left(\sum_{\mathbf{S}'} f(\mathbf{S}') N(\mathbf{S}',t)\right) N(\mathbf{S},t) \qquad (11.2)$$

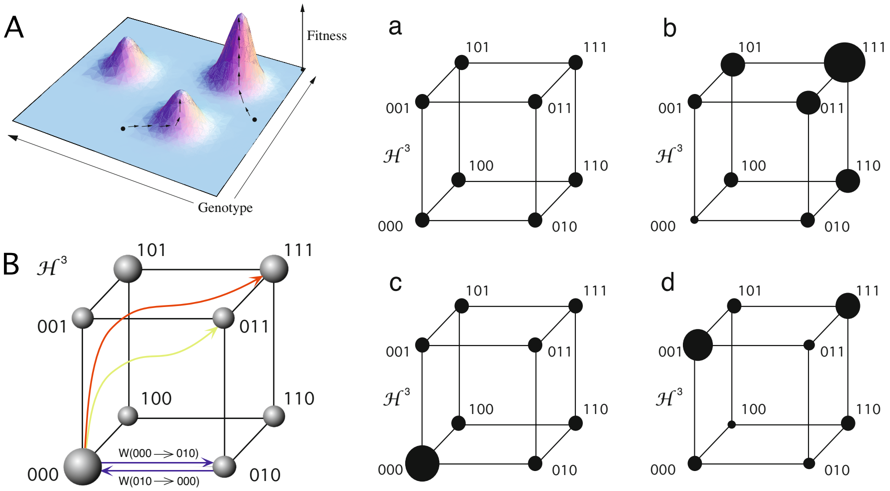


Fig. 11.4 (A) Schematic representation of a fitness landscape with three peaks. Depending on the initial condition in the genotype space, the population will evolve towards different maxima due to mutations which result in motions in the landscape. (B) Fitness landscapes can also be represented in the sequences spaces (here using digital genomes). Such systems allow defining trajectories followed through string evolution. Four standard cases of 3-dimensional sequences space (where the size of the nodes denotes each string fitness value) are also displayed: (a) flat, (b) Fujiyama, (c) Swetina-Schuster (single peak), and (d) rugged fitness landscapes.

The first term on the right-hand site of Equation (11.2) corresponds to positive contributions to the abundance of **S** due to mutation transitions from other strings of the sequence space. The second term includes all the reverse events leaving the node occupied by **S**. Figure 11.4(B) illustrates the information described in Equation (11.2) for 3-bit strings (i.e., $\nu = 3$). The nodes of the hypercube indicate the population size and the three potential transitions from 000 to other strings differing in one, two or all bits are indicated by arrows of different colors (see [18] for the extension of the previous results to discrete dynamical systems).

Simple fitness landscapes can be defined from our previous definitions. Roughly, a sequences space is a discrete space including all sequence combinations of a given sequence, which are connected to neighboring sequences differing in one bit. This space thus results in a set of nodes or vertices (sequences) which are connected by single-point mutations (see Figure 11.4(a-d)). For a given sequence of length ν, the dimension of the sequences space is given by $\mathscr{H}^{\nu}$. For DNA or RNA sequences, the total number of nodes in the sequences space will be 4^{ν}, which results in an astronomic number. As we saw before, this inherent complexity can be reduced using digital genomes defined as bit strings or as "up" and "down" spins. The bit-strings approach allows one to simulate the processes of (co-)evolution and selection under different types of rules or interactions describing different processes in biological systems. For instance, in cancer dynamics [69], in RNA viruses [18, 41, 64, 65, 66, 68, 70], and, in the context of coevolution in matching-alleles dynamics (see Section 11.4), among others. As a simple, illustrative example, Figure 11.4 shows four different simple fitness landscapes. The sequences space in Figure 11.4(a) is a flat fitness landscape, where all the sequences have the same fitness f_0, according to $\mathscr{H}(\mathbf{S}_i) = f_0$. If the sequences space has a least fit sequence (e.g., 0) and then the fitness increases at increasing number of mutations, we have the so-called Fujiyama fitness landscape, shown in Figure 11.4(b), with the fitness given by $\mathscr{H}(\mathbf{S}_i) = f_0 + \sum_{k=1}^{v} S_{ik}$ (see [55] for the application of the Fujiyama landscape to RNA viral populations). Another widely studied case is the so-called single-peak fitness landscape, which is shown in Figure 11.4(c). For this landscape, the fitness can be defined as $\mathscr{H}(\mathbf{S}_i) = f_0 \delta_{\mathbf{S_i},\mathbf{1}} + f_1(1 - \delta_{\mathbf{S_1},\mathbf{1}})$, with $f_0 > f_1$ (see [46, 63, 68] for some examples and applications of the single-peak fitness landscape).

11.3.2 Evolution on Rugged Fitness Landscapes

The sharp, single-peak fitness landscape cited at the end of the previous section defines an extreme in a hierarchy of models introducing different levels of dependencies among genes. A different approximation deals with landscapes in a much more general way, by allowing them to display a given number of local maxima generating a mountainous landscape. Together with the simple fitness landscapes we display a rugged fitness landscape in Figure 11.4(d). For this landscape, where each sequence has a different fitness resulting in as many peaks as sequences, the fitness can be given by $\mathscr{H}(\mathbf{S}_i) = \frac{1}{v} \sum_{k=1}^{v} \xi_{ik}$, with $\xi_{ik} \in k[0,1]$ being a random number. The best known model for the evolution on rugged fitness landscapes is Kauffman's NK model [37, 38], which is also defined on a hypercube. It was originally proposed as a representation of haploid genomes involving two alleles per locus with additive contributions to fitness from different loci.

The NK model is a simple model of random epistatic interactions. In this model N is the number of parts of a system (e.g., genes in a genotype or amino acids in a protein). Each part makes a fitness contribution which depends upon that part as well as upon K other parts among the N. Thus, K reflects how richly cross-coupled the system is indicating how many other genes influence a given gene, i.e., the richness of epistatic interactions among the genes. Such a model, under parameters alteration, generates a family of increasingly rugged multi peaked landscapes. Once again a fitness function is introduced, $\mathbf{f} = f(S_{i1}, ..., S_{iv})$, and changes in the traits are assumed to occur by means of single, one-bit steps (i.e., single-point mutations). These single-chain events are consistent with our assumption of small mutation rates. In this way, a given string obtained by inaccurate replication allows to perform a random *adaptive walk* from a given node towards one of its v nearest neighbors if this leads to an increase in fitness. A direct consequence of this process is that once a local peak is reached, no further changes are allowed to occur. This is completely different from the assumptions made above, which assume the presence of a preferred sequence around which other sequences have a lower fitness value. In the context of NK landscapes, a local peak is very simply defined: if all nearest neighbors in the hypercube are less-fit, we have a fitness local maximum.

How can we construct a system displaying a NK landscape? Kauffman suggested a simple approach using fitness tables: for each element S_{ij}, if it is influenced by K other elements, each element contributes in an additive way to the overall fitness. In other words, if we consider the two-locus model and assume that a given locus i contributes to the global fitness associated to $\mathbf{S}$ by an amount $f_i(\mathbf{S}) \in [0,1]$, the global fitness is given by the average value:

$$f(\mathbf{S}) = \frac{1}{\nu}\sum_{i=1}^{\nu} f_i(\mathbf{S}).$$

As K grows, the ruggedness of the landscape increases, since more complex interactions are allowed to occur.

An interesting feature of the NK model is that, because of its simplicity, it allows the prediction of some evolutionary dynamical patterns. As an example, consider that fitness values are random and uncorrelated, i.e., $f(S_{i1},...,S_{i\nu}) = \xi$, where $\xi \in [0,1]$ is a random number with uniform distribution. This random fitness landscape has many local fitness peaks. This number, M_L, given as

$$M_L(\nu) = \frac{2^\nu}{\nu+1},$$

is very large, and thus our digital sequences can get trapped in a very large number of optima. To see this, consider the number of neighbors of a given node and compute the probability that this node is a local maximum. The chance that it is the fittest among its ν neighbors and itself, given the random choice of values, is simply $P_1 = 1/(\nu+1)$. Since there are 2^ν possible strings, the fraction of those who are local maxima is $M_L(\nu) = 2^\nu P_1$. An extension of this model can easily be introduced by means of the so-called Fujiyama landscape (see the previous section), where a fitness function is defined now as follows:

$$f(S_i) = \frac{1}{z}(1-s)^k,$$

with $k = \nu - \sum_{l=1}^{\nu} S_{il}$, where $z = \sum_{j=1}^{\nu}(1-s)^j$ is a normalization factor. The parameter s weights the steepness of the peak.

Similarly, the presence of epistatic interactions can be introduced using the sequence-dependent fitness:

$$f(\mathbf{S}_i) = \frac{1}{2}\left[\left(\frac{1}{2}\right)^{d_H(\mathbf{S}_i,\mathbf{S}_n)^\xi} + 1\right],$$

where $\xi > 0$ defines the degree and type of epistasis (for $\xi < 1$ we would have antagonistic interactions whereas $\xi > 1$ defines synergistic epistasis (see [18] and references therein)).

11.3.3 Coevolution on Rugged Fitness Landscapes

The previous sections provided the reader with a broad framework that will now be extended to the subject of coevolution. As we already focused on evolution on rugged fitness landscapes, let us start with coevolution on these landscapes (in following sections we will analyze coevolutionary dynamical models in simpler fitness landscapes). In the context of rugged fitness landscapes, the NK model can be modified to analyze evolution between many interacting species, by means of the so-called NKC model [37]. This model introduces a new parameter, named C, which denotes the number of couplings between different species (also represented as strings). Now, the fitness of the NK model needs to be modified in order to introduce the coupling effects where each trait receives inputs from C other traits from different species. These traits are randomly chosen between the S species of the ecosystem. The NKC model includes three main parameters describing: (a) the number of traits required to characterize a given species (N), (b) the number of so-called epistatic interactions among genes in the same species (K), and (c) the number of interactions among traits of different species (C), which introduce coevolution.

Figure 11.5 illustrates this approach for $N = 3$ traits of two interacting species. In the figure, the local peaks are indicated by black circles. Each species is defined by a set of traits, which are coded by bit-strings. Such traits are connected among the different species. In Figure 11.5(a), species 1 is not located in a local peak. As a result of an adaptive walk, it will reach the local peak by mutation. However, as a result of the change in species 1, species 2 is now not located in a fitness peak. The landscape of species 2 has been modified by the adaptive motion of species 1.

The NKC model was analyzed computationally by Kauffman and Johnsen [37], and they showed that this system was dynamically very rich. They identified a chaotic phase, where the ecosystem is always changing and species never end up in a particular configuration (i.e., species do not stop at a given local peak). As we will discuss later, many other different models suggest that chaos can be found in Red Queen dynamics. They also identified a frozen phase, where all species settle down to local peaks. Interestingly, for finite systems at the boundary between the chaotic and the frozen phase in the parameters space, small perturbations generate a coevolutionary avalanche of changes through the system. Typically occurring at critical states, the distribution of such avalanches was shown to follow a power-law. Kauffman and Johnsen mapped these avalanches into extinction events, suggesting that the number of changes in species was proportional to the extinction of less-fit variants. Such a result did not fit the predictions of fossil record extinctions. However, a variation of this model by Newman and Palmer [50], which allowed changes in the parameters, gave an exponent which agreed with fossil record data (see also Section 11.6).

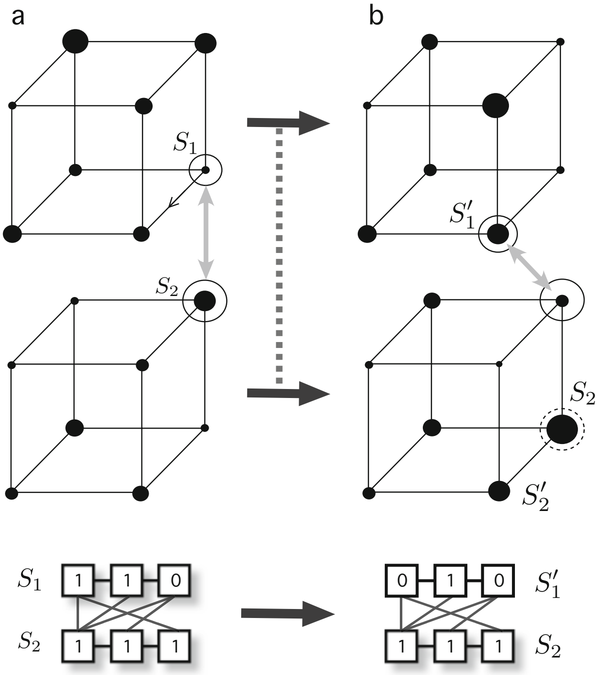


Fig. 11.5 Coevolutionary dynamics in the NKC model. Here only two species (S_1 and S_2) are considered, each one described by $N = 3$ traits, which are represented in cubic sequences spaces. Black circles in the hypercube nodes indicate the fitness associated to each string, and current states are highlighted with open circles in (a). Each bit is assumed to interact with a number of different bits from the other string (genome) as indicated in the lower diagrams. Once the first species changes by climbing to a local optimum, the landscapes of both species get modified. The second species now will be forced to change too, since it is now placed in a low-fit state and will next shift to another local peak (here indicated with a dashed circle in (b)). If no such movement is possible, extinction can take place.

The two phases of the NKC model can be derived from simple theoretical arguments setting $K = N - 1$ [3]. It is known that a given species, in order to reach a local fitness, needs a number of walks L_w, which is on average $L_w \approx \ln(N)$. If we assume that all species are at local peaks and one of them, named a, is perturbed (i.e., is randomly positioned on the fitness landscape), then a will start again to climb some other local peak. If C is large enough (i.e., interactions among species are important), the other species except a can see their landscape modified also starting to change. Following this idea, where each adaptive walk involves a change in a given trait, which can in turn affect other species traits, we can determine a critical condition given by a combination of N and C able to trigger a chain reaction able to percolate through the system. More concretely, the probability that a given trait in a random species depends on species a is C/N. The critical condition is that at least one change in a species occurs. This actually means:

$$L_w \frac{C_c}{N} = 1,$$

with $C_c = N/\ln(N)$. That is, on average, one out of C randomly chased genes is among the L_w changed genes. In other words, when the number of traits is such that $C > C_c$, interactions among different genotypes constantly modify the underlying fitness landscapes, a scenario under which coevolutionary avalanches take place.

11.3.4 Red Queen Dynamics in RNA Virus

First consider a very simple example of coupled fitness landscapes and two identical populations climbing and competing on them. The Red Queen hypothesis of evolution has been widely discussed within the context of RNA viruses [55, 70], where the dynamics of viral populations can be interpreted as a dynamical process of growth, competition and selection taking place in the sequence space. The fitness landscape for a virus is usually defined in terms of replication, infectivity or transmission rates, among others. The landscape appears as a multipeaked surface, where the local maxima represent optimal fitness values which can be reached by mutation. Here, the initial condition plays an important role since depending on where the quasispecies[2] is located in the sequences space, the population will evolve by exploring near genotypes by mutation. Competition experiments between several clonal viral populations [8, 55] provided a good illustration of two basic principles of evolutionary ecology: the Red Queen dynamics and the principle of competitive exclusion.

Experimental results were carried out with two clones of Vesicular stomatitis virus (VSV, see Figure 11.6). Such experiments involved the mixing of two clonal populations of VSV of equal fitness. Passage experiments allowing these populations to grow and compete were performed using standard virus plaque assays. More specifically, genetically marked monoclonal antibody-resistant (MARM) clones of equal fitness to the wild-type VSV were used and their relative frequencies were monitored along passages.

The MARM clones only differed in a single mutation with neutral affects not changing viral fitness. These experiments revealed that both competing populations grew up showing steady increases of fitness, but, at some point, one of the two populations suddenly excluded the other one. The winner of this competition process was not always the same (see Figure 11.6). Although the time scale of the divergence seemed highly predictable. The simultaneous increase of fitness of the two populations and their predictable divergence was suggested to be a product of the Red Queen effect [70]. In the context of RNA viruses, newly arising mutants with higher fitness were able to outcompete lower-fitness ones. At the level of viral genomes or sequences, a favorable mutation within one quasispecies triggers evolutionary responses in the second one, forcing it to evolve. Overlapped with this evolutionary process, and related to the dynamics, the principle of competitive exclusion is also

[2] The term *quasispecies* is used to define the heterogeneous population of viral genomes in RNA viruses.

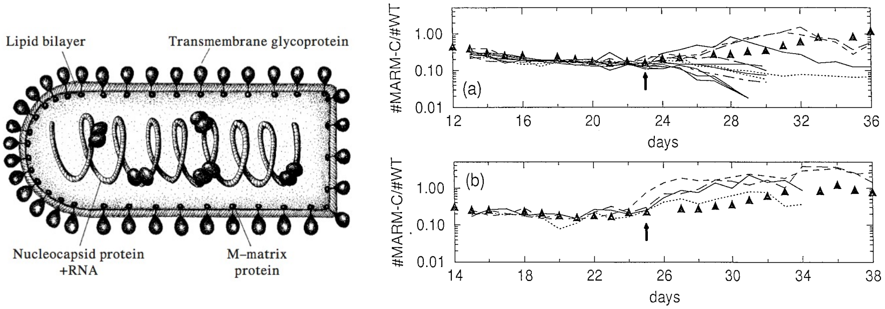


Fig. 11.6 Left: Vesicular stomatitis virus (VSV) virion which contains a negative-sense, single-stranded RNA genome. The bullet shape is characteristic of the *Rhabdoviridae* family (drawing by Ricard V. Solé). Right: Time changes of MARM-C:wt ratio in independent replicas. (a) Shows eighteen replicas started after passage 12. (b) Displays six replicas started from passage 14. In both panels the populations start to diverge after approximately 23 passages (indicated with small arrows). Plots obtained from [55].

at play. This principle states that when two species are strongly competing for the same finite resources, the fitter one asymptotically outcompetes the least fit.

The previous experiments were modeled by Solé and collaborators [70] using different approaches. The simplest one was a bit-string model that considered a population of N bit-strings, named S_i, with sequences: $S_i = S_i^1 S_i^2 ... S_i^\nu$; with $i = 1, 2, ... N$ and $S_i^j \in \{0, 1\}$. At each time generation (passage), the algorithm repeated N times the following rules: a random string, say S_i, of the population was chosen for replication. Replication, proportional to replication probability $r(S_i)$, took place by replacing one of the strings of the population (also randomly chosen), say S_j by a copy of S_i. Replication presented error, at a rate (per bit and replication cycle) μ. So the probability to copy exactly the same bit was $1 - \mu$. The mapping between sequence composition (genotype) and replication rate (phenotype) was done using the Fujiyama fitness landscape (see Figure 11.4(b)), involving the linear relation:

$$r(s_i) = \frac{1}{\nu} \sum_{j=1}^{\nu} S_i^j.$$

As we previously explained, this fitness landscape ignores epistatic interactions.

Computer simulations revealed the same behavior obtained with the experiments with VSV. Figure 11.7 displays the outcome of the model for a population of $N = 3000$ strings of length $\nu = 16$. The populations were initialized in such a way that the initial fitness of both populations was low, also keeping equal their mean replication rates. During the simulations, the strings were competing for the available space (i.e., N available sites). The upper panel of Figure 11.7 shows the total population size of each population, which was maintained roughly constant along time. However, after approximately $t \approx 20$ passages, one of the two populations started outcompeting the other one, that finally disappeared. Once ecological

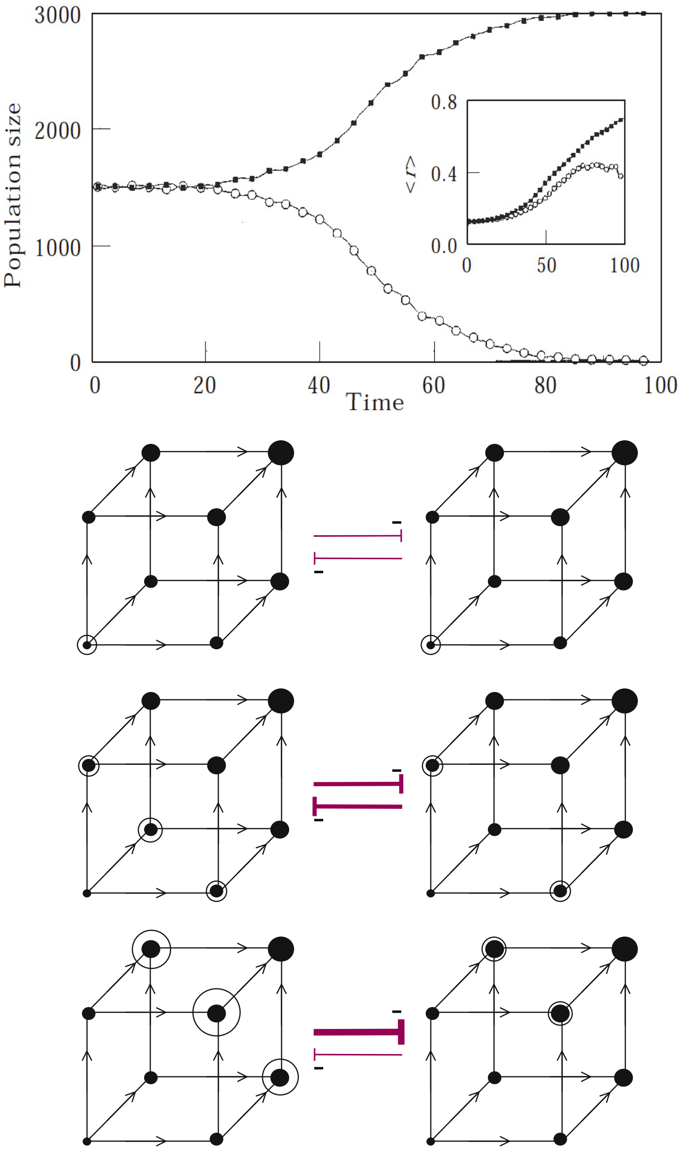


Fig. 11.7 (Upper) Dynamics of the bit strings model using a population with $N = 3000$ strings of length $\nu = 16$ using $\mu = 10^{-3}$. The main plot shows the time dynamics of both populations. The inset shows the mean fitness, $\langle r \rangle$, also for both populations along time (see [70]). The observed changes can be easily interpreted in terms of a parallel climbing of both species on their Fujiyama landscapes together with ongoing competition for resources. Below we illustrate this by means of a small, three-bit landscape. Initially both species (their populations are indicated with open circles) grow slowly and competition is weak. As they climb up and increase their replication rates, competition become strong and symmetry breaking occurs (see text).

competition became tight, selection pressure became stronger and the initial parallel growth in fitness for both populations was no longer sustainable. This dynamical divergence was a direct outcome of a "symmetry breaking" phenomenon which explained the VSV experiments (see [55, 70] for details).

11.4 Gene-for-Gene and Matching Alleles Models of Coevolution

Coevolution is the change of a biological object triggered by the change of a related object. Coevolution can occur at many biological scales: at the molecular level as correlated mutations between amino acids in a protein [92], or at the macroscopic scale as covarying traits between different interacting species in an environment. In coevolution, each entity exerts selective pressures on the other, thereby affecting each other's evolution. This process is schematically illustrated in Figure 11.8(d). Here we show two fitness landscapes for preys and predators. Imagine preys are viruses (or cells infected by viruses) and predators are cytotoxic lymphocytes (cells of the immune system that kill infected cells) that act upon the activation of the adaptive immune response. If the virus, located in the peak with the green dot is able to mutate, visiting the highest peak, the immune system will not be able to recognize and remove it. However, if the virus moves towards the lower peak, which is recognized by dendritic cells or macrophages, able to trigger the immune response, virus populations with this genotype will be impaired in terms of number of particles due to the action of cytotoxic lymphocytes that will remove infected cells. This simple example illustrates how the evolution of one of the partners influences the evolution of the other, and *vice versa*, in a coevolutionary arms race. Broadly speaking, coevolutionary interactions can be antagonistic or mutualistic. The former involve negative interactions such as predation or parasitism. The latter being found when two or more species coevolve by means of cooperation. Coevolution can occur for two interacting species (pairwise coevolution) or can involve a number of different species, which are evolving in responses to another set of species (diffuse coevolution).

There are at least about six proposed forms of coevolution between species, some involving reciprocal adaptation and others a combination of adaptation and speciation [81, 82, 83]. In the context of coevolution between hosts and parasites, some of them had a particular importance in this type of interactions. One is Ehrlich and Ravens [16] hypothesis of how the evolution of defence and counterdefence in host-parasite interactions may lead to the radiation of species through the process of escape-and-radiation coevolution. The genetic basis of infection in real ecosystems has been also represented by two major models: the so-called gene-for-gene (hereafter GFG) and matching alleles (hereafter MA) models. The GFG model is based on data from plant-pathogen interactions, especially in the field of crop plants [20]. Interestingly, the first mathematical model of coevolution was explicitly based on assumptions of a GFG interaction [47]. Later, a multitude of real examples on GFG coevolution were identified between plants and pathogens, mainly between plants

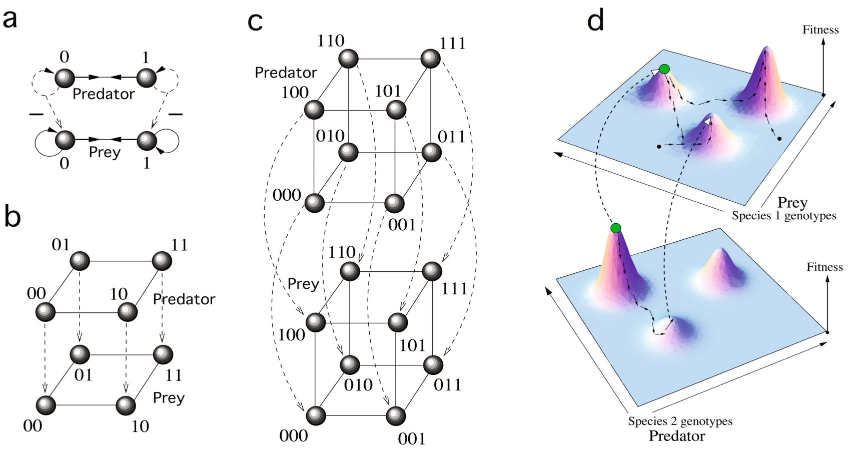


Fig. 11.8 Coevolutionary phenomena can be interpreted as the coupling of fitness landscapes by means of ecological interactions (dashed arrows). We show neutral sequences spaces for predator-prey (host-parasite) species with perfect matching alleles interactions for (a) $\nu = 1$, (b) $\nu = 2$ and (c) $\nu = 3$, being ν the length of the sequences. (d) Prey's evolutionary fate will not only depend on its own and independent exploration of the fitness landscape, but also on predator's evolution (indicated by the small arrows). If a given prey (green circle) moves towards the highest peak, it will escape predator's action increasing its fitness. However, if prey climbs to the lowest peak, and the predator mutates moving to the same peak, host's fitness and reproductive success will diminish.

and fungi, bacteria and viruses (see [84] for a review). The GFG hypothesis states that "for each gene that conditions reaction in the host there is a corresponding gene that conditions pathogenicity in the parasite" [84, 39]. The key feature of this model is that one parasite genotype can infect all host genotypes.

As a difference, in the MA model, favored by invertebrate zoologists [23], an exact genetic match is required for infection (Figure 11.8(a-c)). MA models underlie most of the theory constructed to understand the effects of host-parasite coevolution on sex and recombination [27, 30]. Parker [52] pointed out the importance of MA models for the study of sexual reproduction, suggesting it may hamper the generality of the Red Queen theory for sex. It has been argued that both GFG and MA models are not totally disconnected: they are two ends of a continuum (see [2] for further details).

In the following sections, we will introduce and review recent models and results about coevolving replicators with antagonistic interactions, mainly focusing on MA models. These models, although being suitable to analyze coevolution at small scales (e.g., immune system-viruses) provide good intuitions in larger scales, such as in spatially-extended ecosystems (Section 11.5.1). In Section 11.5.2. we will develop some theory aimed to describe the dynamics arising from MA predator-prey interactions. Finally, we will explore large scale coevolution by means of a complex network model reproducing the extinction pattern found in the fossil record data discussed in the introduction section.

11.5 Minimal Coevolutionary Systems

Coevolutionary phenomena can be studied considering minimal models. Such models can help us to understand fundamental phenomena arising from species interactions and evolution. We notice that coevolution is a highly nonlinear phenomena, since species interactions give place to nonlinear couplings that can result in very rich and complex dynamics. In this section we will first introduce a minimal system of replicators with matching alleles (MA) dynamics moving, replicating and evolving on a surface. Then, we will develop a general mathematical model describing MA interactions for haploid genotypes, assuming well-mixed populations thus ignoring spatial correlations. As the reader will see, the dynamics of such small and simple systems can indeed be very complex.

11.5.1 Spatial Red Queen Dynamics

At the beginning of this chapter we have illustrated the idea of coevolution with a very simple model simulating replicator spatial dynamics with MA interactions. Together with such model, other approaches have focused on the same subject by considering further complexity, such as spatial diffusion of replicators or larger sequences spaces. Recently, Sardanyés and Solé [59] explored a similar system simulating coevolution for host-parasite (prey-predator) replicators also using cellular automata (CA) models.

The authors explored the spatio-temporal dynamics for three different host-parasite systems considering 1-bit, 2-bits and 3-bits strings (the corresponding coupled hypercubes are displayed in Figure 11.8(a-c)). Thus, one of the aims of their work was to analyze the effects of increasing the size of the sequences space in the spatio-temporal dynamics for MA interactions.

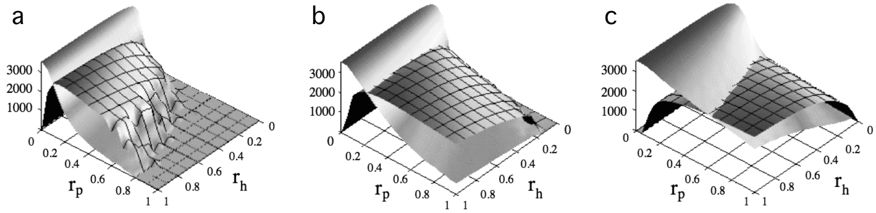


Fig. 11.9 Population equilibria for hosts (plain surfaces) and parasites (gridded surfaces) replicators in the parameters space (r_h, r_p) for the spatially-extended model of coevolution with diffusion using $\mu_h = \mu_p = 10^{-2}$. Three different systems were analyzed, with: (a) $\nu = 1$ bit, (b) $\nu = 2$ bit and (c) $\nu = 3$ bit (see [59] for further details).

The model considered two populations of replicators given by host (prey) strings of size ν: $\mathbf{S}_h^i = (s_h^{i1},...,s_h^{i\nu})$; and by parasite (predator) strings of the same size, $\mathbf{S}_p^i = (s_p^{i1},...,s_p^{i\nu})$ with $S_h^{ij}, S_p^{ij} \in \{0,1\}$ where $i = 1,...,N$, and N is the number of different genotypes ($N = 2^\nu$). Both populations reproduced and evolved on a two-dimensional space with toroidal boundary conditions. For this model we used the so-called Von Neumann neighborhood, which considers interactions with the four nearest neighbors. Specifically, the state-transition rules of the CA considered self-replication with mutation, decay and spatial motion of strings. Rules implemented are given below.

State transition rules of the host-parasite CA model with matching allele interactions [59]

(1) **Self-replication:** If a host and a parasite occupy the same spatial position and have the same sequence of bits (i.e., perfect MA), the parasite eliminates the host and replicates, with probability r_p, to a random neighbor provided it is empty. If only the host lives in the cell, it replicates with probability r_h to a neighbor cell provided it is not occupied by another host string. Replication involves point mutations for host and parasites, with mutation probabilities μ_h and μ_p, respectively. These rules can be represented by the following set of reactions:

$$S_h^i + \vartheta \xrightarrow{r_h(1-\mu_h)^\nu} 2S_h^i, \tag{11.3}$$

$$S_h^i + \vartheta \xrightarrow{r_h W_{ij}^h} S_h^i + S_h^{j\neq i}. \tag{11.4}$$

Reactions (11.3) and (11.4) denote, respectively, error-free and erroneous host replication.

$$S_h^i + S_p^j + \vartheta \xrightarrow{\delta_{ij} r_p (1-\mu_p)^\nu} 2S_p^j, \tag{11.5}$$

$$S_h^i + S_p^j + \vartheta \xrightarrow{\delta_{ij} r_p W_{jl}^p} S_p^j + S_p^{l\neq j}. \tag{11.6}$$

Similarly, reactions (11.5) and (11.6) represent, respectively, error-free and erroneous parasites replication, which is nonlinear due to the density-dependence of the antagonistic interaction. The parameter δ_{ij} is again the Kronecker δ function where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise; and ϑ indicate some available building blocks (i.e. mononucleotides) needed to built new strings. The terms W_{ij}^k, with $k = \{h, p\}$, correspond to the probabilities of erroneous replication for hosts (h) and parasites (p), and are given by:

$$W_{ij}^k = (1-\mu_k)^{\nu - d_H[S_i^k, S_j^k]} \cdot \mu_k^{d_H[S_i^k, S_j^k]},$$

where $d_H[S_i^k, S_j^k]$ is the Hamming distance between two sequences,

$$d_H[S_i^k, S_i^k] = \sum_{i=1}^{\nu} |s_i^k - s_i^k|. \tag{11.7}$$

Equation (11.7) is a function returning the number of different bits when comparing two sequences. Such a function is also used to determine the matching allele interaction between both host (S_i^h) and parasite (S_i^p) sequences, now with

$d_H[S_i^h, S_i^p] = \sum_{i=1}^{\nu} |s_i^h - s_i^p|$, where s_i^h and s_i^p represent the bit value (0 or 1) in the ith position in both strings. A perfect matching allele interaction will occur when $d_H[S_i^h, S_i^p] = 0$.

(2) **Molecular decay:** Host and parasite strings decay with probability δ_h and δ_p, respectively, according to:

$$S_h^i \xrightarrow{\delta_h} \vartheta,$$

$$S_p^i \xrightarrow{\delta_p} \vartheta,$$

(3) **Local diffusion:** Host and parasite strings move, independently and randomly, to empty neighbor cells with diffusion probabilities D_h and D_p, respectively.

To simplify the analysis, all the simulations were run with maximum diffusion constants $D_h = D_p = 1$, also setting $\delta_h = \delta_p = 10^{-2}$. The lattice was randomly inoculated by either host and parasites random sequences.

As we previously mentioned, the rules were implemented for three different systems with different strings' lengths: $\nu = 1$ bit, $\nu = 2$ bit, $\nu = 3$ bit. For all three different values of ν, the system underwent the same three types of asymptotic dynamics: (i) stable coexistence of hosts and parasites with sustained fluctuations; (ii) hosts survival and parasites extinction; and (iii) both hosts and parasites extinction (i.e., coextinction). Populations' trajectories in phase space suggested the presence of chaotic coevolutionary dynamics [59] (see also [60]). In order to characterize the importance of these three possible asymptotic states listed above, we built parameter spaces considering two key evolutionary parameters of hosts and parasites coevolution: self-replication and mutation rates. Figure 11.9 illustrates the outcome of some simulations in the parameter space (r_h, r_p) showing the average population numbers for both global populations[3] after some transient was removed for $\nu = 1$ (Figure 11.9(a)); $\nu = 2$ (Figure 11.9(b)); and $\nu = 3$ (Figure 11.9(c)). A key result was that the increase of the length of the replicators (i.e., increase of the size of the sequences space) promoted stable coevolution, as shown by the reduction of host-parasite coextinctions (Figure 11.9).

Moreover, for each of the genotype lengths we simulated three different scenarios, characterized by different values of hosts and parasites mutation rates. The simulations revealed that asymmetries in mutation rates between hosts and parasites had an important effect in the population dynamics: hosts were only able to escape from parasites (causing parasites extinction) if they mutated much faster. Under this condition, the scenario of host's survival and parasites extinction was found for extremely low values of hosts' self-replication rates. On the contrary, when $\mu_p > \mu_h$, the region of parameters space with host and parasite extinction increased for the three hypercubes analyzed, indicating that when parasites evolved faster than hosts they were more efficient in catching hosts thus increasing coextinction phenomena.

[3] By global populations we mean the sum of all possible genotypes for a given population i.e., hosts or parasites.

11.5.2 Dynamics of Small Replicators with Matching-Allele Interactions

The previous computational models considered antagonistic populations of bit-strings replicating and mutating on a surface. The same system can be investigated with a mathematical model by assuming no spatial correlations (i.e., infinite diffusion). A general model describing predator-prey matching-alleles (MA) interactions can be formulated using a time-continuous deterministic model. Assuming a perfect MA (see Figure 11.8 and Figure 11.10(a)), where each predator genotype can predate only on its homologous prey genotype (i.e., predator genotype i predates on prey genotype i, with $i \in \{0,1\}$), the model is given by the following system:

$$\dot{x_i} = k_i^h x_i \left(1 - \frac{\sum_{j \in \mathscr{H}^\nu} x_j}{\mathscr{K}}\right) - A_i \xi(x_i, y_i) + \\ + \frac{\mu_i^h}{\nu}\left(\sum_{<j>_i} x_j - x_i\right) - \varepsilon_i^h x_i, \qquad (11.8)$$

$$\dot{y_i} = \xi(x_i, y_i) + \frac{\mu_i^p}{\nu}\left(\sum_{<j>_i} y_j - y_i\right) - \varepsilon_i^p y_i, \qquad (11.9)$$

with

$$\xi(x_i, y_i) = \frac{k_i^p y_i x_i}{C_i + x_i}. \qquad (11.10)$$

The state variables x_i and y_i indicate, respectively, the concentration or population numbers of the ith prey and of the ith predator genotype (with $i = 1 \ldots 2^\nu$), which define a ν-dimensional sequences space $\mathscr{H}^\nu$.

Note that prey genotypes have a logistic-like growth constraint in self-replication indicated in the first term in parenthesis of Equation (11.8), where $\sum_{j \in \mathscr{H}^\nu} x_j$, is the total prey population and $\mathscr{K}$ the carrying capacity of the prey genotypes. The logistic term involves exponential growth for small population numbers and saturation as population values approach the carrying capacity. Moreover $1/A_i$ is the yield coefficient of prey genotype i to predator genotype i. Equation (11.10) is a Holling "type II" functional response [6, 28], where predation rate is a saturating function of prey density. C_i and k^p are constants parameterizing the saturating functional response. The constant k^p, which describes the maximal predation rate, can also be interpreted as the predator's maximal self-replication or intrinsic growth rate.

Both terms $\sum_{<j>_i} x_j - x_i$, and $\sum_{<j>_i} y_j - y_i$, denote genetic diffusion by mutation among neighboring genotypes for both prey and predator genotypes, which are proportional to μ^h and μ^p, respectively. Moreover, k_i^h denotes the self-replication constant (intrinsic growth rates) for prey genotypes; ε_i^h and ε_i^p are decay rates which can be interpreted as spontaneous hydrolysis rates as well as density-independent death rates. If only a single genotype is present in each species, Equations (11.8) and (11.9) are close to the well-known Rosenzweig-MacArthur model [89].

As the reader will see, since the dimension of the dynamical system described by the Equations (11.8) and (11.9) depends upon the length of the sequences, one may expect different types of dynamics for different values of ν. As we will discuss in the following sections, where we review results for $\nu = 1$ [61] (Figure 11.10(a)) and $\nu = 3$ [62] (Figure 11.8(c)), this is the case. The fitness landscape for this predator-prey system using $\nu = 1$ is shown in Figure 11.10(a). Actually, this system was studied for two different fitness landscapes, given by a flat or neutral fitness landscape, with $k_i^h \equiv k^h$ and $k_i^p \equiv k^p$, $\forall i$ (i.e., all genotypes share the same fitness values) and for an asymmetric fitness landscape with $k^p < K^p$ or $k^p > K^p$ (i.e., one of the predator's genotypes has a higher fitness in terms of population growth). For this latter case, a predator with a higher fitness actually means that a given genotype is more efficient in catching its preferred prey. For both scenarios we also assumed that mutation rates for both prey and predator genotypes were equal, i.e. $\mu_i^h \equiv \mu^h$ and $\mu_i^p \equiv \mu^p$, $\forall i$ (see [61] for further details).

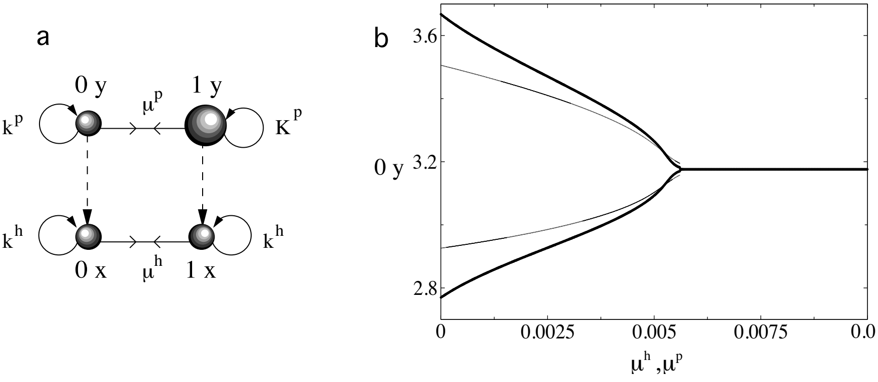


Fig. 11.10 (a) Minimal predator-prey system with matching alleles interaction (vertical dashed lines) modeled with the Equations (11.8) and (11.9) for $\nu = 1$. Upper and lower marbles correspond, respectively, to predator (y) and prey (x) genotypes, which reproduce at rates k_p and k_h (circular arrows), and mutate at rates μ_p and μ_h (straight arrows), respectively. We show an asymmetric fitness landscape for predators, with larger replication rates for predator genotype 1 (i.e., $K_p > k_p$). (b) Bifurcation diagram for predators with genotype 0, using mutation rates (μ^h and μ^p are represented with thick and thin lines, respectively) as control parameters for the neutral fitness landscape (i.e., $K_p = k_p$). At decreasing mutation rates, the system undergoes a Hopf bifurcation resulting in a permanent oscillatory behavior governed by a periodic orbit [61].

The analysis of the qualitative behavior of Equations (11.8) and (11.9) was performed assuming a neutral fitness landscape with $\varepsilon_i^h = \varepsilon_i^p \equiv \varepsilon$. This system was shown to have three fixed points, given by ($x_i^* = 0, y_i^* = 0$), ($x_0^*, x_1^*, 0, 0$), and ($x_0^*, x_1^*, y_0^*, y_1^*$). The first fixed point, if stable, involved predator-prey extinction. The second equilibrium involved prey survival and predator's extinction whereas the third fixed point involved predator-prey coexistence, and thus it was the potential state of Red Queen dynamics. This third fixed point, under symmetry conditions, named $(\mathbf{x}^*, \mathbf{y}^*)$, was given by:

$$x^* = \frac{\varepsilon^p}{k^p - \varepsilon^p},$$

$$y^* = \frac{(1+x^*)(k^h - 2k^h x^* - \varepsilon^h)}{k^p}.$$

To study the stability of this fixed point it was further assumed that $k^h = k^p = 1$, also considering symmetry in decay rates, i.e., $\varepsilon^h = \varepsilon^p = \varepsilon$. Under these conditions, the fixed point reads:

$$x^* = \frac{\varepsilon}{1-\varepsilon},$$

$$y^* = \frac{(1-4\varepsilon+\varepsilon^2)}{(1-\varepsilon)^2}.$$

After some algebra, and after fixing $\mu^h = \mu^p \equiv \mu$, a critical mutation rate causing a Hopf bifurcation was identified at:

$$\mu_c = \frac{\varepsilon(1-4\varepsilon+\varepsilon^2)}{4(1-\varepsilon)}.$$

Such a bifurcation, which involves the emergence of a periodic orbit causing sustained, periodic oscillations, was confirmed by numerical simulations (Figure 11.10(b)). Interestingly, the same bifurcation was also numerically found for the fitness landscape with asymmetries in predator's replication rates. Counterintuitively, the asymmetric fitness landscape revealed that the most efficient predator genotypes achieved lower population equilibria (see [61] for details). Our results identifying periodic Red Queen dynamics were in agreement with other mathematical models on coevolution (see [11, 13]).

11.5.3 Red Queen Chaotic Attractors

So far, we have discussed the dynamical behavior of Equations (11.8)-(11.9) for $\nu = 1$, which can be governed by stable fixed points as well as by a periodic orbit causing sustained and regular oscillations of the predator-prey genotypes populations. The same model, analyzed for $\nu = 3$, revealed much richer dynamics: under some parameter regions, both populations behave chaotically. Hence, similar to what is known as diffusion-induced chaos [53], it was found that the simplest system (with $\nu = 1$), governed by a periodic orbit, could be governed by chaotic attractors at increasing the number of available alleles (more available nodes in sequences space).

It is known that dynamical systems governed by a periodic orbit can become unstabilized to chaos when spatial correlations and diffusion are included. Actually, some decades ago a great deal of attention was paid to self-organization processes in reaction-diffusion systems, and their relevance in chemistry, physics and biology was repeatedly stressed [24, 51, 88]. In this sense, numerical investigations of the spatially-extended Belousov-Zhabotinsky chemical reaction showed the presence of chaotically oscillating structures. Moreover, diffusion-induced chaos has also been discussed in the context of spatial ecological dynamics [53]. We notice that MA chaotic dynamics can also be interpreted from the perspective of dynamical unstabilization due to diffusion in space. That is, each sequence of the sequences space can be interpreted as a patch, and populations can diffuse between patches because of mutation (i.e., diffusion in sequences space). Under this view, the oscillatory behavior of variables x_i and y_i in Equations (11.8) and (11.9) for $\nu = 1$, become unstabilized to chaos for $\nu = 3$.

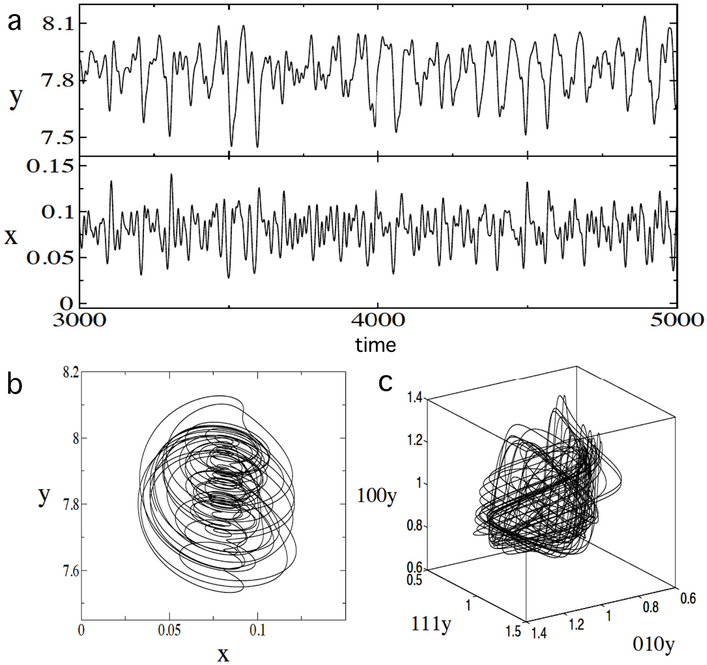


Fig. 11.11 Red Queen chaotic dynamics for the Equations (11.8) and (11.9) using $\nu = 3$. (a) Global population time dynamics of parasites (y) and host (x). (b) Strange attractor governing host-parasite global dynamics. In (c) we show the chaotic attractor in the parasites three-dimensional phase space for genotypes $(010y, 111y, 100y)$ (see [62]).

Figure 11.11 shows the chaotic coevolutionary dynamics for the 3-bits sequences modeled with Equations (11.8) and (11.9). The time series in Figure 11.11(a) are represented for the global populations of predators (y) and preys (x). In Figure 11.11(a) and (b) we show, respectively, the chaotic attractors for global host-parasite populations (represented in the phase space (x,y)) as well as the attractor for parasite genotypes 010, 111, and 100 (see [62] for further details).

Our previous results suggested that Red Queen dynamics can be chaotic even for small haploid replicator systems with MA interactions. It was previously shown that large networks with host-parasitoid replicators can also behave chaotically. More specifically, Kaneko and Ikegami [31, 36] characterized the so-called homeochaos in multi-species models with antagonistic interactions and evolution. They suggested that chaos, more than a destabilizing behavior [5], could involve stability in multi-species ecosystems through a weak, chaotic state arising in high-dimensional dynamical systems. Roughly, homeochaos was suggested to suppress strong chaos causing large fluctuations that could near populations to extinction. Homeochaos is characterized by many positive, but close to zero Lyapunov exponents (i.e., a type of hyperchaos). Such a property of the spectrum of Lyapunov exponents involves narrow chaotic fluctuations with small amplitude, which are able to keep population numbers far away from attractors involving extinction. The concept of homeochaos was later extended to low-dimensional systems, and its role was discussed in both deterministic and stochastic host-parasitoid models with discrete time generations [60].

Chaotic evolutionary dynamics have been found in other theoretical studies of genetic polymorphisms under frequency-dependent selection (see for example [19, 45, 67]). Moreover, Dercole and colleagues [11] recently showed that predator-prey coevolutionary models governed by periodic fluctuations became chaotic when the system is embedded in a three-species food chain model by the addition of a superpredator able to coevolve. These authors argued that over space, genetically-driven chaos may cause evolutionary divergence of local metapopulations, even under the absence of environmental change, thus promoting genetic diversity among ecological communities over long evolutionary time.

11.6 Large Scale Coevolution on Complex Networks

Our last example deals with large-scale evolutionary dynamics. The study of large scale coevolution was also performed in multi-species models using complex networks theory. A very simple model of large-scale evolution involving a set of N interacting species can be easily defined [72, 73, 74, 75]. In this model, species interactions are introduced by means of a $N \times N$ connectivity matrix $\mathbf{W} = (W_{ij})$. Evolution for this system is introduced through changes in its elements. Similar to some of the models previously presented, here the "state" of each species is described by a binary variable $S_i \in \{0,1\}$ ($i = 1,2,...,N$), for the i-th species, with $S_i = 1$ or $S_i = 0$ if the species is alive or extinct, respectively. So the whole ecosystem is described in terms of a simple directed graph where the connections are initially set to random values. Each species receives either positive or negative inputs from other species. These signs indicate that the given species is favored or harmed by the species which send the input. For instance, a negative input would correspond to the interaction with a predator or with a parasite. Alternatively, a positive input would correspond to mutualism or symbiosis. Such a model, in its simplest form, can be formulated in terms of a set of rules.

Rules of the network coevolution model (illustrated in Figure 11.12(a-d))

(1) **Random changes in the connectivity matrix**. At each generation, we select one input connection for each species and assign it to a new, random value without regard for the previous state of the connection. This rule introduces changes into the web, which can be due to evolutionary responses or to environmental changes of some sort. In other words, changes derived from coevolution among two species, innovation at the species level and/or environmental-driven changes are lumped together within this rule.

(2) **Extinction**. Changes in the connectivity will eventually lead to extinctions. Extinction events are decided by computing the total sum of the inputs for each species. This sum, if negative will involve the extinction of the species ($S_i = 0$), and all its connections are removed. Otherwise, nothing happens ($S_i = 1$). Hence, the state of the i-th species is updated following the following dynamical equation:

$$S_i(t+1) = \Phi\left[\sum_{j=1}^{N} W(i,j)\right]$$

where $\Phi(z) = 1$ for positive z and zero otherwise.

(3) **Diversification**. A number of species can disappear due to the extinction rule, leaving empty sites. These sites will be refilled by diversification: each extinct species is replaced by a randomly chosen survivor. The replacement is made by simply copying the connections of the survivor into the empty site.

This model shows a strongly nonlinear behavior with avalanches of extinction as well as the correct power law distribution of extinction sizes [44, 72, 74, 78] (although with an exponent typically close but higher than $\alpha = 2$, see also [15]). The outcome of the model was that both small and very large events were generated *by the same* dynamical rules. Most of the times, the extinction of a given species had no consequences for the other species. But from time to time, a given (keystone) species with positive inputs to others disappeared. The removal of this species was suggested to have a destabilizing effect on others, able to cause further propagation leading to mass extinction events.

What is the origin of such extinction patterns? We first need to see how a given species can shift towards a negative sum of inputs. The reason is easily understood from rule 1 above.

Since changes of links among species are random and the new values are chosen from a uniform distribution, we can expect that, in the long term, the sum of inputs will decay to zero. If we look at the sign of the links, so that the probability of finding positive links, $P(W^+) = P[W_{ij} > 0]$, and the probability of finding negative links $P(W^-) = P[W_{ij} < 0]$. The time evolution of the positive links can be described in terms of a master equation, given by

$$\frac{dP(W^+,t)}{dt} = w(W^- \to W^+)P(W^+) - w(W^+ \to W^-)P(W^-),$$

where $P(W^+)+P(W^-)=1$, starting from an initial condition $P(W^+,0)=P_0$. Since, from the first rule, we have $w(W^- \to W^+) = w(W^+ \to W^-) = 1/2N$, the master equation reads:

$$\frac{dP(W^+,t)}{dt} = \frac{1}{2N}\left[1-P(W^+)\right],$$

and an exponential decay is obtained:

$$P(W^+,t) = \frac{1}{2}\left[1+(2P_0-1)e^{-t/N}\right].$$

As a consequence, the sum of inputs $\mathscr{F}_i = \sum_j W_{ij}$ will also decay exponentially: $\mathscr{F}_i(t) \sim e^{-t/N}$, predicting an exponential decay in the probabilities of survival, as expected from the Red Queen hypothesis (see the introduction section). This rule actually introduces the basic ingredients of Van Valen's theory. All species in the system keep changing all the time (either due to biological or environmental causes) eventually reaching extinction. The ultimate fate of all species is to become extinct, and

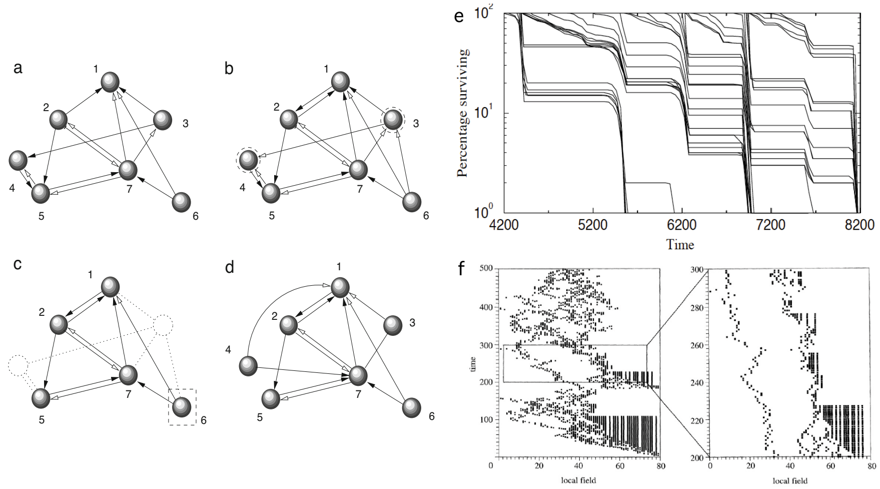


Fig. 11.12 (a-d) Rules of the evolution model exemplified with a network with $N=7$ nodes and a given initial connectivity (a). Negative and positive interactions are indicated with black and white arrows, respectively. The first step of the model is to modify the links (b): here different pairs of interactions are found, including mutualism, parasitism, predation and competition. For this particular case, species 3 and 4 become nonviable, dying in the next step (c). Species 6 is selected (among the survivors) and copied into the two empty sites (Rule 3). Such copies carry the links of their parent species, which will be modified again by the first rule in an iterative process. This set of rules generates a very complex dynamical pattern of species evolution. In (e) we display an example of the survival curves obtained from our model (compare with Figure 11.1). In (f) we also show the evolution of the local fields over time. A mixture of slow and rapid changes occur, in a punctuated fashion (see [73] for further details).

so an exponential decay in the survival probability will be observed. Here, however, there is no intrinsic, species-level variability (in terms of a genotype) and the fate of a given species will be dominated by network responses and chance. Ecological-driven phenomena are the key forces in the long run, although small-scale events are taking place all the time. The nature of the decay turns to be exponential on average but episodic when looking at pseudocohorts (see Figure 11.1), consistently with Raup's analysis commented on in the introduction of this chapter (compare Figures 11.1 and 11.12).

An analytic study of the previous model is rather difficult because of the random nature of the interaction matrix. The model can, however, be simplified by mapping the set of rules into a linear model [44, 78], giving place to a mean field approach to the network model.

Consider a set of N species, characterized by a single integer quantity ϕ_i ($i = 1,2,...,N$). This quantity will play the role of the internal field. Each species is now represented by this single (integer) number $\phi_i \in \{-N,-N+1,...,-1,0,1,...,N-1,N\}$, which represents the sum of inputs from other species. The dynamics consist of three steps: (a) with probability $P = 1/2$, $\phi_i \to \phi_i - 1$, otherwise no change occurs (this is equivalent to the randomization rule in the network model); (b) all species with $\phi_i < \phi_c$ (below a given threshold) are extinct. Here we use $\phi_c = 0$ but other choices give the same results. The number of extinct species, $0 < E < N$, gives the size of the extinction event. All E extinct species are replaced by survivors. Specifically, for each extinct site (i. e. when $\phi_j < \phi_c$) we choose one of the $N-E$ survivors ϕ_k and update ϕ_j to $\phi_j = \phi_k$; (c) after an extinction event, a wide reorganization of the web structure occurs [75]. In this simplified model this is introduced as a coherent shock. Each of the survivors are updated as $\phi_k = \phi_k + q(E)$, where $q(E)$ is a random integer between $-E$ and $+E$. This mean-field approach defines a three-step process. If $N(\phi)$ indicates the frequency of species having a local field ϕ, we have:

$$N(\phi,t+1/3) = \frac{1}{2}N(\phi,t) + \frac{1}{2}N(\phi+1,t),$$

$$N(\phi,t+2/3) = N(\phi,t+1/3) + N(\phi,t+1/3)\sum_m \frac{m}{N-m}P(m),$$

if $\phi > 0$ and zero otherwise. Finally,

$$N(\phi,t+1) = N(\phi,t+2/3) - N(\phi,t+1/3) + \sum_{q>-\phi} N(\phi+q,t+1/3)P(q).$$

From these equations, the full master equation for the dynamics reads:

$$N(\phi,t+1) = \frac{1}{2}\sum_{q=-\infty}^{+\infty}\sum_m \frac{P(m)}{2m+1}\theta(m-|q|)\left[N(\phi+q,t) - N(\phi+q+1,t)\right]$$

$$+\frac{1}{2}[N(\phi,t)+N(\phi+1,t)]\sum_m \frac{mP(m)}{N-m}.$$

where two basic statistical distributions, which are self-consistently related, have been used. These are:

$$P^*(q) = \sum_m \frac{P_e(m)}{2m+1}\theta(m-|q|),$$

which is an exact equation giving the probability of having a shock of size q. The second is $P_e(m)$, the extinction probability for an event of size m. We have a mean-field approximation relating both distributions:

$$P_e(m) = \sum_q P^*(q)\delta\left[\sum_{\phi=1}^{q-1} N(\phi) - m\right].$$

The last equation introduces the average profile $N(\phi)$, i. e. the time-averaged distribution of ϕ-values. For the mesoscopic regime $1 \gg q \gg N$, by applying a Taylor expansion to the master equation, i. e.

$$N(\phi) = \frac{1}{2}\sum_{q=-\infty}^{+\infty}\sum_m \frac{P(m)}{2m+1}\left\{2N(\phi+q) + \left.\frac{\partial N}{\partial \phi}\right|_{\phi+q} + \frac{1}{2}\left.\frac{\partial^2 N}{\partial \phi^2}\right|_{\phi+q} + \ldots\right\} +$$

$$+\frac{1}{2}\left\{2N(\phi) + \left.\frac{\partial N}{\partial \phi}\right|_{\phi} + \frac{1}{2}\left.\frac{\partial^2 N}{\partial \phi^2}\right|_{\phi} + \ldots\right\}\sum_m \frac{mP(m)}{N-m},$$

and using a continuous approximation, it is easy to see that the previous equation reads:

$$\frac{1}{2}\int dm \int_{-m}^{m} \frac{P(m)}{2m}\left\{2\left[\frac{N(\phi+q)}{N(\phi)} - 1\right] + \left.\frac{\partial LnN}{\partial \phi}\right|_{\phi+q} + \ldots\right\} +$$

$$+\frac{1}{2}\left\{2 + \left.\frac{\partial LnN}{\partial \phi}\right|_{\phi} + \ldots\right\}\int \frac{mP(m)}{N-m}dm = 0.$$

Assuming that $N(\phi)$ decays exponentially, i. e. $N(\phi) = \exp(-c\phi/N)$, we can integrate each part of the last equation, using $N(\phi+q)/N(\phi) = \exp(-cq/N) \approx 1 - cq/N$. The first term cancels exactly, the second gives $-2c/N$ and the third scales as $(1 - O(1/N))N^{1-\tau}$. So the previous equation leads to

$$-\frac{2c}{N} + N^{1-\tau}G\left[1 - O\left(\frac{1}{N}\right)\right] = 0,$$

in order to satisfy this equality, we have $\tau = 2$, which gives us the scaling exponent for the extinction distribution. Hence, in agreement with Burlando's analysis, the taxonomy that emerges from this model also displays fractal behavior (with an exponent $\alpha_b \approx 2$).

These models, able to reproduce observed patterns, can have important implications for evolutionary theory. An intense debate over the last decades has concerned the basic mechanisms operating at different temporal scales. Some authors (specially in the field of population genetics) suggested that the rules operating at small scales (i.e., microevolutionary events) can be directly translated into the process of macroevolution [14, 25, 42]. However, other authors like Stephen Jay Gould, claimed that different processes are at work in evolution at different scales [22], although no well-defined mechanism for such decoupling was proposed. We notice that the network organization of ecologies, changing in a coevolving landscape, suggests a possible source of decoupling the micro- and macro scales. Moreover, these models can also help understanding the complex dynamical behavior of large extinctions and their aftermath [7, 58, 76, 77].

11.7 Conclusions

Coevolutionary dynamics introduces an additional layer beyond single-species evolution. Coevolution pervades biology on multiple scales but its role and impact – as illustrated by our previous examples – is rather different at each scale. In microorganisms, the changes that couple diverse species (as hosts and pathogens) within a given ecosystem are associated with many different molecular events related to membrane receptors, production and pumping of toxins, development of aggregates or resistance to antibiotics, to cite just a few. Arm races are known to occur and take place on short time scales.

Coevolution occurs in other systems, including technological ones. Coevolution between predators and prey predate at least part of the evolutionary events that triggered the emergence of complex animals at the base of the Cambrian explosion. It is likely that the so called Ediacaran fauna, dominated by simple, filtering organisms with small developmental complexity became replaced by the well known, Burgess-Shale pattern as a consequence of predator-prey arm races. Many challenges lie ahead in our understanding of how coevolution shaped biological complexity and how to properly approach it from a theoretical perspective. Among other questions, we still need to understand how to connect ecological networks and coevolving landscapes, how to place these landscapes in the middle of the multidimensional space involving development, ecology and the environment, and what universal trends are to be found in their structure and dynamics. The previous examples only provide a glimpse of the richness and complexity arising in (co-)evolution, but they also illustrate the power of simple models to be able to address relevant questions.

Acknowledgements. We want to thank Sergi Valverde for help in model implementation and the other members of the Complex Systems Lab for useful discussions. This work has been funded by the Botín Foundation, by the James S. McDonnell foundation and by the Santa Fe Institute.

References

[1] Agrawal, A.F., Lively, C.M.: Parasites and the evolution of self-fertilization. Evolution 55(5), 869–879 (2001)

[2] Agrawal, A.F., Lively, C.M.: Infection genetics: gene-for-gene versus matching-alleles models and all points in between. Evol. Ecol. Res. 4, 79–90 (2002)

[3] Bak, P., Flyvjerg, H., Lautrup, B.: Coevolution in a rugged fitness landscape. Phys. Rev. A 46, 6724–6730 (1992)

[4] Benton, M.J.: Red Queen hypothesis. In: Briggs, D.E.G., Growther, P.R. (eds.) Paleobiology, Blackwell, Oxford (1995)

[5] Berryman, A.A., Millstein, J.A.: Are ecological systems chaotic - and if not, why not? Trends Ecol. Evol. 4, 17–28 (1986)

[6] Case, T.J.: An Illustrated Guide to Theoretical Ecology. Oxford University Press, Oxford (2000)

[7] Chen, Z.Q., Benton, M.J.: The timing and pattern of biotic recovery following the end-Permian mass extinction. Nat. Geosci. 5, 375–383 (2012)

[8] Clarke, D.K., Duarte, E.A., Elena, S.F., Moya, A., Domingo, E., Holland, J.J.: The Red Queen reigns in the kingdom of RNA viruses. Proc. Natl. Acad. Sci. U.S.A. 91, 4821–4824 (1994)

[9] Darwin, C.: On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. John Murray, London (1859)

[10] Decaestecker, E., Gaba, S., Raeymaekers, J.A.M., Stoks, R., van Kerckhoven, L., Ebert, D., de Meester, L.: Host-parasite 'Red Queen' dynamics archived in pond sediment. Nature 450, 870–873 (2007)

[11] Dercole, F., Ferriere, R., Rinaldi, S.: Chaotic Red Queen coevolution in a three species food chain. Proc. Roy. Soc. B 277, 2321–2330 (2012)

[12] de Visser, J.A.G.M., Elena, S.F.: The evolution of sex: empirical insights into the roles of epistasis and drift. Nat. Rev. Genetics 8, 139–149 (2007)

[13] Dieckmann, U., Marrow, P., Law, R.: Evolutionary cycling in predator-prey interactions: Population dynamics and the Red Queen. J. Theor. Biol. 176, 91–92 (1995)

[14] Dobzhansky, T.: Genetics and the Origin of Species, 3rd edn. Columbia University Press, New York (1951)

[15] Drossel, B.: Biological evolution and statistical physics. Adv. Phys. 50, 209–295 (2001)

[16] Ehrlich, P.R., Raven, P.H.: Butterflies and plants: A study in coevolution. Evolution 18, 586–608 (1964)

[17] Eigen, M.: Selforganization of matter and evolution of biological macromolecules. Naturwiss 58, 465–523 (1971)

[18] Elena, S.F., Solé, R.V., Sardanyés, J.: Simple genomes, complex interactions: Epistasis in RNA virus. Chaos 20, 26106 (2010)

[19] Ferrière, R., Fox, G.A.: Chaos and evolution. Trends Ecol. E 10, 480–485 (1995)

[20] Flor, H.H.: The complementary genetic systems in flax and flax rust. Adv. Genetics 8, 29–54 (1956)

[21] Freund, H., Wolter, R.: Evolution of bit strings: some preliminary results. Complex Systems 5, 279–298 (1991)

[22] Gould, S.J.: The Structure of Evolutionary Theory. Harvard University Press, Cambridge (2003)

[23] Grosberg, R.K., Hart, M.W.: Mate selection and the evolution of highly polymorphic self/nonself recognition genes. Science 289, 2111–2114 (2000)

[24] Haken, H.: Advanced Synergetics. Springer Series in Synergetics. Springer, New York (1983)
[25] Haldane, J.B.S.: The Causes of Evolution. Longmans and Green, London (1932)
[26] Hamilton, W.D.: Sex vs. non-sex vs. parasite. Oikos 35, 282–290 (1980)
[27] Hamilton, W.D., Axelrod, A., Tanese, R.: Sexual reproduction as an adaptation to resist parasites (a review). Proc. Natl. Acad. Sci. U.S.A. 87, 3566–3573 (1990)
[28] Hastings, A., Powell, T.: Chaos in a three-species food chain. Ecology 72(3), 896–903 (1991)
[29] Hoffman, A.: Testing the Red Queen hypothesis. J. Evol. Biol. 4, 1–7 (1991)
[30] Howard, R.S., Lively, C.M.: Parasitism, mutation accumulation and the maintenance of sex. Nature 367, 554–557 (1994)
[31] Ikegami, T., Kaneko, K.: Evolution of host-parasitoid network through homeochaotic dynamics. Chaos 2, 397–407 (1992)
[32] Ilachinsky, A.: Cellular Automata. A Discrete Universe. World Scientific, Singapore (2000)
[33] Jacob, F.: Evolution and tinkering. Science 196, 1161 (1977)
[34] Jacob, F.: Molecular tinkering in evolution. In: Rondall, D.S. (ed.) Evolution from Molecules to Men. Cambridge University Press, Cambridge (1983)
[35] Jaenike, J.: An hypothesis to account for the maintenance of sex in populations. Evol. Theor. 3, 191–194 (1978)
[36] Kaneko, K., Ikegami, T.: Homeochaos: dynamic stability of a symbiotic network with population dynamics and evolving mutation rates. Physica D 56, 406–429 (1992)
[37] Kauffman, S.A., Johnsen, J.: Coevolution on the edge of chaos: Coupled fitness landscapes, poised states and coevolutionary avalanches. J. Theor. Biol. 149, 467–505 (1991)
[38] Kauffman, S.A.: The Origins of Order. Oxford University Press, New York (1993)
[39] Kerr, A.: The impact of molecular genetics of plant pathology. Annu. Rev. Phytopathol. 25, 87–110 (1987)
[40] King, K.C., Delph, L.F., Jokela, J., Lively, C.M.: The geographic mosaic of sex and the Red Queen. Curr. Biol. 19, 1438–1441 (2009)
[41] Lafforgue, G., Martínez, F., Sardanyés, J., de la Iglesia, F., Shi-Shun, L., Qi-Wen, N., Solé, R.V., Chua, N.H., Darós, J.-A., Elena, S.F.: Tempo and mode of plant RNA virus escape from RNA interference-mediated resistance. J. Virol. 85(19), 9686–9695 (2011)
[42] Li, W.H., Graur, D.: Fundamentals of Molecular Evolution. Sinauer Associates, Sunderland (1991)
[43] Lively, C.M.: Evidence from a New Zealand snail for the maintenance of sex by parasitism. Nature 328, 519–521 (1987)
[44] Manrubia, S.C., Paczuski, M.: A simple model of large-scale organization in Evolution. Int. J. Mod. Phys. C 9, 1025–1032 (1998)
[45] May, R.M., Anderson, R.M.: Epidemiology and genetics in the coevolution of parasites and hosts. Proc. R. Soc. Lond. B 219, 281–313 (1983)
[46] McCaskill, J.S., Altemeyer, S.: Error threshold for spatially resolved evolution in the quasispecies model. Phys. Rev. Lett. 86, 5819 (2001)
[47] Mode, D.J.: A mathematical model for the co-evolution of obligate parasites and their hosts. Evolution 12, 158–165 (1958)
[48] Montoya, J.M., Pim, S., Solé, R.V.: Ecological networks and their fragility. Nature 442, 259–264 (2006)

[49] Morran, L.T., Schmidt, O.G., Gelarden, I.A., Parrish II, R.C., Lively, C.M.: Running with the Red Queen: Host-parasite coevolution selects for biparental sex. Science 333, 216–218 (2011)

[50] Newman, M.E.J., Palmer, R.G.: Modeling Extinction. Oxford University Press, New York (2003)

[51] Nicolis, G., Prigogine, I.: Self-Organization in Non-Equilibrium Systems. Wiley-Interscience, New York (1977)

[52] Parker, M.A.: Pathogens and sex in plants. Evol. Ecol. 8, 560–584 (1994)

[53] Pascual, M.: Diffusion-induced chaos in a spatial predator-prey system. Proc. Roy. Soc. London B 251, 1–7 (1993)

[54] Molecular evolution on rugged landscapes: proteins, RNA and the immune system. In: Perelson, A.S., Kauffman, S. (eds.) SFI Studies in the Sciences of Complexity, vol. IX. Addison-Wesley, Redwood (1991)

[55] Quer, J., Huerta, R., Novella, I.S., Tsimring, L., Domingo, E., Holland, J.J.: Reproducible nonlinear population dynamics and critical points during replicative competitions of RNA virus quasispecies. J. Mol. Biol. 264, 465–471 (1996)

[56] Raup, D.M.: Biological extinction and Earth history. Science 231, 1528–1533 (1986)

[57] Raup, D.M.: A kill curve for phanerozoic marine species. Paleobiology 17, 37–48 (1991)

[58] Roopnarine, P.D.: Extinction cascades and catastrophe in ancient food webs. Paleobiology 32(1), 1 (2006)

[59] Sardanyés, J., Solé, R.V.: Chaotic stability in spatially-resovled host-parasite replicators: The Red Queen on a lattice. Int. J. Bif. and Chaos 17(2), 589–606 (2007)

[60] Sardanyés, J.: Low dimensional homeochaos in coevolving host-parasotoid dimorphic populations: Extinction thresholds under local noise. Commun. Nonlinear Sci. Numer. Simul. 16, 3896–3903 (2011)

[61] Sardanyés, J., Solé, R.V.: Matching allele dynamics and coevolution in a minimal predator-prey replicator model. Phys. Lett. A 372, 341–350 (2008)

[62] Sardanyés, J., Solé, R.V.: Red Queen strange attractors in host-parasite replicator gene-for-gene coevolution. Chaos, Solitons and Fractals 32(5), 1666–1678 (2007)

[63] Sardanyés, J., Elena, S.F.: Error threshold in RNA quasispecies models with complementation. J. Theor. Biol. 265, 278–286 (2010)

[64] Sardanyés, J., Elena, S.F.: Quasispecies spatial models for RNA viruses with different replication modes and infection strategies. PLoS One 6(9), e24884 (2011)

[65] Sardanyés, J., Solé, R.V., Elena, S.F.: Replication mode and landscape topology differentially affect RNA virus mutational load and robustness. J. Virol. 83(23), 12579–12589 (2009)

[66] Sardanyés, J., Elena, S.F., Solé, R.V.: Simple quasispecies models for the survival-of-the-flattest effect: The role of space. J. Theor. Biol. 250, 560–568 (2008)

[67] Seger, J.: Evolution of exploiter-victim relationships. In: Crawley, M.J. (ed.) Natural Enemies: the Population Biology of Predators, Parasites and Diseases, pp. 3–25. Blackwell, Oxford (1992)

[68] Solé, R.V., Sardanyés, J., Díez, J., Mas, A.: Information catastrophe in RNA viruses through replication thresholds. J. Theor. Biol. 240, 353–359 (2006)

[69] Solé, R.V.: Phase transitions in unstable cancer cell populations. Europ. Phys. Journal 35(1), 117–124 (2003)

[70] Solé, R.V., Ferrer, R., González-García, Q.J., Domingo, E.: Red Queen dynamics, competition and critical points in a model of RNA virus quasispecies. J. Theor. Biol. 198, 47–59 (1999)

[71] Solé, R.V., Bascopmte, J.: Self-organization in Complex Ecosystems. Princeton University Press, Princeton (2006)

[72] Solé, R.V.: On macroevolution, extinctions and critical phenomena. Complexity 1, 40–46 (1996)

[73] Solé, R.V., Bascompte, J., Manrubia, S.C.: Extinction: bad genes or weak chaos? Proc. Roy. Soc. B 263, 161–168 (1996)

[74] Solé, R.V., Manrubia, S.C.: Extinction and self-organized criticality in a model of large-scale evolution. Phys. Rev. E 51, 6250–6253 (1996)

[75] Solé, R.V., Manrubia, S.C.: Criticality and unpredictability in macroevolution. Phys. Rev. E 55, 4500–4508 (1997)

[76] Solé, R.V., Montoya, J., Erwin, D.H.: Recovery after mass extinction: evolutionary assembly in large-scale biosphere dynamics. Phil. Trans. Roy. Soc. B-Biol. Sci. 357, 697–707 (2002)

[77] Solé, R.V., Saldaña, J., Montoya, J.M., Erwin, D.H.: Simple model of recovery dynamics after mass extinction. J. Theor. Biol. 267, 193–200 (2010)

[78] Solé, R.V., Manrubia, S.C., Mercader, J.P., Benton, M., Bak, P.: Long-range correlations in the fossil record and the fractal nature of macroevolution. Adv. Compl. Syst. 1, 255–266 (1998)

[79] Stenseth, N.C., Maynard Smith, J.: Coevolution in ecosystems: Red Queen evolution or stasis? Evolution 38, 870–880 (1984)

[80] Tegmark, M.: An icosahedron-based method for pixelizing the celestial sphere. The Astrophys. J. 470, L81–L84 (1996)

[81] Thompson, J.N.: Interaction and coevolution. Wiley, New York (1982)

[82] Thompson, J.N.: Concepts of coevolution. Trends Ecol. Evol. 4, 179–183 (1989)

[83] Thompson, J.N.: Coevolution and the evolutionary genetics of interactions among plants and insects and pathogens. In: Burdon, J.J., Leather, S.R. (eds.) Pests, Pathogens and Plant Communities, pp. 249–271. Blackwell, Oxford (1990)

[84] Thompson, J.N., Burdon, J.J.: Gene-for-gene coevolution between plants and parasites. Nature 360, 121–125 (1992)

[85] Van Valen, L.: Energy and evolution. Evol. Theory 1, 179–229 (1976)

[86] Van Valen, L.: Evolution as a zero-sum game for energy. Evol. Theory 4, 129–142 (1980)

[87] Van Valen, L.: A new evolutionary law. Evol. Theory 1, 1–30 (1973)

[88] Vidal, C., Pacault, A. (eds.): Non-Equilibrium Dynamics in Chemical Systems. Springer Series in Synergetics. Springer, New York (1984)

[89] Weltz, J.S., Levin, S.: Size and scaling of predator-prey dynamics. Ecol. Lett. 9, 548–557 (2006)

[90] Wright, S.: Evolution in Mendelian populations. Genetics 16, 97 (1931)

[91] Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution. Proceedings of the Sixth International Congress on Genetics 1, 356 (1932)

[92] Yip, K.Y., Patel, P., Kim, P.M., Engelman, D.M., McDermott, D., Gerstein, M.: An integrated system for studying residue coevolution in proteins. Bioinformatics 24, 290–292 (2008)

Chapter 12
Eco–Evolutionary Dynamics on Deformable Fitness Landscapes

Richard A. Watson and Marc Ebner

Abstract. Conventional approaches to modelling ecological dynamics often do not include evolutionary changes in the genetic makeup of component species and, conversely, conventional approaches to modelling evolutionary changes in the genetic makeup of a population often do not include ecological dynamics. But recently there has been considerable interest in understanding the interaction of evolutionary and ecological dynamics as coupled processes. However, in the context of complex multi-species ecosytems, especially where ecological and evolutionary timescales are similar, it is difficult to identify general organising principles that help us understand the structure and behaviour of complex ecosystems. Here we introduce a simple abstraction of coevolutionary interactions in a multi-species ecosystem. We model non-trophic ecological interactions based on a continuous but low-dimensional trait/niche space, where the location of each species in trait space affects the overlap of its resource utilisation with that of other species. The local depletion of available resources creates, in effect, a deformable fitness landscape that governs how the evolution of one species affects the selective pressures on other species. This enables us to study the coevolution of ecological interactions in an intuitive and easily visualisable manner. We observe that this model can exhibit either of the two behavioural modes discussed in the literature; namely, evolutionary stasis or Red Queen dynamics, i.e., continued evolutionary change. We find that which of these modes is observed depends on the lag or latency between the movement of a

Richard A. Watson
University of Southampton, School of Electronics and Computer Science,
Institute for Life Sciences, Institute for Complex Systems Simulation,
Agents Interaction and Complexity, Highfield, Southampton, SO17 1BJ, UK
e-mail: raw@ecs.soton.ac.uk

Marc Ebner
Ernst Moritz Arndt Universität Greifswald,
Institut für Mathematik und Informatik,
Walther-Rathenau-Strasse 47, 17487 Greifswald, Germany
e-mail: marc.ebner@uni-greifswald.de

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_12, © Springer-Verlag Berlin Heidelberg 2014

species in trait space and its effect on available resources. Specifically, if ecological change is nearly instantaneous compared to evolutionary change, stasis results; but conversely, if evolutionary timescales are closer to ecological timescales, such that resource depletion is not instantaneous on evolutionary timescales, then Red Queen dynamics result. We also observe that in the stasis mode, the overall utilisation of resources by the ecosystem is relatively efficient, with diverse species utilising different niches, whereas in the Red Queen mode the organisation of the ecosystem is such that species tend to clump together competing for overlapping resources. These models thereby suggest some basic conditions that influence the organisation of inter-species interactions and the balance of individual and collective adaptation in ecosystems, and likewise they also suggest factors that might be useful in engineering artificial coevolution.

12.1 Introduction

Conventional population genetic models of evolution generally address selection acting on genetic changes within a single population without regard for changes to the ecological context of that selection, and conversely, conventional ecological models (e.g., Lotka-Volterra models) generally address changes in the abundance of each species without regard for genetic change within each species [28, 37, 39, 42]. However, it is clear that changes in the genetic composition of a species can affect its fitness dependencies with other species (either directly, as in trophic interactions, or by changing the overlap of resources utilised, as in non-trophic interactions) and hence alter the ecological dynamics of an ecosystem. Reflexively, the selective pressures acting on a population can be greatly influenced by its biotic environment and thus ecological dynamics also shape evolutionary changes. Recently, there has been considerable interest in the interaction of ecological and evolutionary dynamics in an attempt to understand them as coupled 'eco-evo' processes [25, 28, 29]. Adaptive dynamics models, for example, take account of the fact that the selective pressures on a genetic variant are sensitive to ecological conditions and, conversely, that genetic changes can alter ecological equilibria [25]. These models provide a simple way to link ecological and evolutionary dynamics when the relevant timescales are almost separated (i.e., genetic changes are assumed to occur at ecological equilibrium). But when ecological and evolutionary timescales are strongly overlapping, it can still be difficult to identify general principles that help us understand their interaction [28, 29]. For example, when multiple species compete for a set of shared resources, under what conditions will competitive feedback cause them to diversify onto separate resources? And conversely, under what conditions will multiple species evolve to compete for the same resources causing continued conflict and inefficient use of resources?

Meanwhile, conventional evolutionary computation methods utilise a single population and address optimisation problems corresponding to static fitness landscapes. Cooperative coevolution [30] approaches to function optimisation utilise multiple populations each contributing a part of a combined solution. The aim of this approach is a form of problem decomposition where each species addresses a relatively small semi-independent sub-problem and remaining dependencies between sub-problems can be optimised by virtue of evaluation in a ‘shared domain model’ [30]. Competitive coevolution approaches to optimisation problems generally use one population to represent solutions and another to represent problem instances that need to be solved. For example, Hillis [17] evolved sorting networks in one population and data sets to be sorted in the other; Juille and Pollack [19] evolved cellular automata rule sets against parameters of the initial conditions; Reynolds [33], Miller and Cliff [3, 4, 26], Floreano and Nolfi [12] and Floreano et al. [13] evolved pursuers against evaders either in simulation or using real robots. In some cases, competitive coevolution can lead to an arms race where each population continually challenges the other to improve, with the potential to keep the problem population in the zone of proximal development for the solution population [43] and also perhaps provide an open-ended adaptive pressure [44]. However, both uses of coevolution can sometimes fail to deliver these ideals. In cooperative coevolution the main problem is finding a way to automatically decompose the problem into suitable sub-problems such that different populations find diverse semi-independent sub-problems [45]. In competitive coevolution, species may ‘disengage’, breaking the mutual selective pressure, or evolve to exploit each others specific weaknesses rather than evolve general solutions, or chase each other around in endless cycles of relative improvement that fail to yield any improvement in absolute terms [44]. As is the case with natural eco-evolutionary dynamics, it can be difficult to identify general principles that help us understand when co-evolutionary dynamics will produce one type of dynamics rather than another, and, in particular, the conditions that lead to effective co-adaptation.

In natural systems there are, of course, many specific contingencies that may affect the nature of eco-evolutionary dynamics. Likewise, in any given optimisation problem, there are many domain specific, and implementation specific, contingencies that may affect the success of a coevolutionary approach. With the aim of keeping a model as simple as possible but not more so, here we introduce a very simple model of eco-evolutionary interactions that avoids case-specific details, but includes a rich space of possible inter-species fitness interactions and coevolutionary dynamics in a multi-species ecosystem. Specifically, we model the mean phenotype of each species as a point in a continuous low-dimensional (quantitative) trait space. Each point in trait space confers the ability to utilise a particular combination of resources in a continuous multi-dimensional resource or niche space [18, 22]. For example, a particular size and shape of bill confers an ability to utilise a particular size of seed, and/or a tolerance to a particular temperature or humidity enables occupation of corresponding habitats. When a species occupies a particular point in this trait space, depletion of the resources in that corresponding niche creates competition with species of similar genotypes. This provides a simple abstract representation of non-trophic ecological interactions based on niche overlap or species packing [23, 24] where the evolution and coevolution of species alters their relative location in trait space and hence their competitive interaction coefficients. This creates a simulation that captures the notion of multiple populations coevolving on a 'rubber sheet' fitness landscape – each species deforming the fitness landscape of the other – in a quite literal manner. The deforming implies that the fitness landscape is dynamic. Although other abstract approaches to modelling coupled fitness landscapes have been proposed, (e.g. [20]), by examining the mean phenotype of populations in a low-dimensional niche space (one or two dimensions), rather than in a high dimensional genotype space, this approach has the distinct advantage that it is straightforward to visualise the state of the entire ecosystem at a point in time. It also allows us to investigate some specific research questions in a simple and straightforward manner.

In particular, in the following experiments we investigate two related issues: the type of dynamical behaviour exhibited by evolutionary change and the efficiency with which an ecosystem collectively utilises available resources. For example, in this model, what are the conditions under which ecological interactions produce an ever-changing selective pressure that maintains species in a state of perpetual evolutionary change, or conversely, conditions where species equilibrate and stabilise, extinguishing evolutionary change? Also when multiple populations experience the same set of available resources, multiple species might compete for the same high quality resources, or conversely, resource competition might produce a diversifying effect causing species to spread-out and utilise different complementary resources; what factors influence the balance of these behaviours?

The first of these issues is a classic question in ecology and evolution relating to the 'Red Queen' hypothesis [39, 41] as discussed in the following section. The second issue is relevant to the balance of individualistic and collective adaptation typified by artificial competitive and cooperative coevolution, respectively. That is, although in both competitive and cooperative coevolution selection is applied at the individual level not at the collective level, in cooperative coevolution our interest as engineers is nonetheless on the collective welfare of the species in the system (the reason we do not explicitly select on collectives, biological unrealism aside, is that to do so would forfeit the potential for problem decomposition). The desirable dynamics of these two scenarios are therefore quite different. In competitive coevolution only one species represents solutions, and improvements in the other species are only desirable in so much as they motivate improvement in these solutions. In this case, we aim for each population to keep the other population under continued selective pressure, and this implies that improvement in one species confers decline in the other (a 'zero-sum' evolutionary game). Conversely, in cooperative coevolution a solution is represented by a whole set of species and we are interested in essentially the opposite dynamic, where diverse species find ways to be simultaneously good at different aspects of the problem. If they are successful in diversifying appropriately, this implies an increase in collective welfare (informally, 'everyone is better off' and the game is therefore not zero-sum), but it also implies that in successfully minimising competition, species find a way to decrease the mutual conflict and likewise the pressure for continued evolutionary change that they exert on one another. These opposites of individualistic and collective adaptation thus suggest contrasting behaviors with respect to convergence and diversification of species, and also a correspondence with continued evolutionary change and evolutionary stasis, respectively. Thus the type of dynamical behaviour and the efficiency with which an ecosystem collectively utilises resources seem to have interesting interactions that we will investigate. Our goal is to isolate important factors which influence the balance of these different outcomes. We find that both possibilities are possible without changing the nature of the underlying game, or the level at which selection is applied, but merely by altering the coupling between ecological and evolutionary timescales.

Here we utilise our previous work with one- [9] and two-dimensional fitness landscapes [10]. The basic mechanism of the coevolutionary interactions operates as follows. In Figure 12.1 species *A* has already reached a local optimum while Species *B* climbs towards the same local optimum due to Darwinian selection, Figure 12.1(b & c). When two species occupy the same position, they are located in the same niche, it is assumed that they are in direct competition with each other and hence the fitness of both of these species is decreased - which effects a depression of the fitness landscape.

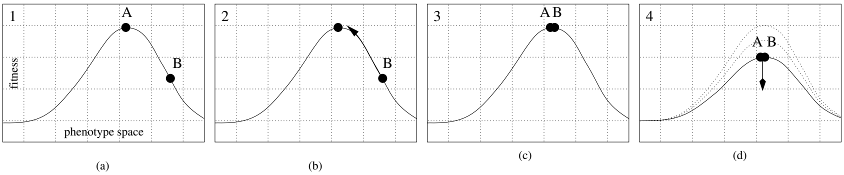


Fig. 12.1 Two species are located on the same fitness landscape. (a) Species A has reached a local fitness peak. (b) Species B is adapting to the same local optimum, i.e. is adapting to the same niche. (c) Species A and B are co-located at the same ecological niche. (d) The presence of species B in the same niche has an impact on the fitness of species A. The fitness of both species is reduced.

Before describing how we apply this basic mechanism in a multi-species model, we will briefly discuss how a two-species system might result in an arms race or Red Queen dynamics.

12.2 The Red Queen Hypothesis

A classic example of coevolutionary selective pressures is found in the predator-prey scenario. Individuals from the predator population need to catch prey in order to survive, and prey need to escape from predators. Such a scenario may lead to an arms race where both sides try to out-compete their opponent [5] in terms of, say running ability or maneuverability. Note that, usually, predator-prey models assume instantaneous interaction between the predator and prey population. A more accurate model might involve some delay between the adaptation of one population and the effect of this evolution on the other population [11].

The idea that a particular trait may be continually changing as a result of selective pressures even though the fitness of both coevolving species remains constant over

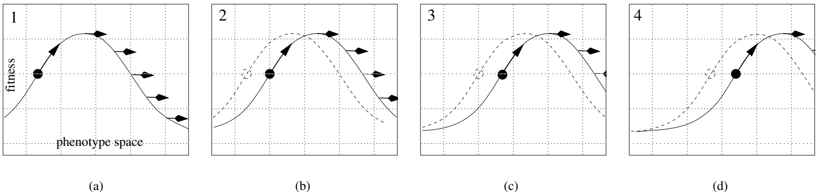


Fig. 12.2 According to the Red Queen hypothesis, the fitness of coevolving species may remain at the same level over evolutionary time even though some particular trait is evolving in response to a selective pressure. Here, it is assumed that as the species adapts to its local optimum, the fitness landscape is deformed by the presence of that species. (a)-(c) The local maximum in the fitness landscape is effectively shifted to the right by this deformation. Thus even though the species has been continually adapting to its local optimum it still has the same fitness over time. In other words, the species is moving through phenotype space even though its fitness remains constant.

time has been called the Red Queen hypothesis after a figure from Lewis Carroll's novel *Through the Looking Glass* [8, 34, 36, 41]. In the novel Alice and the Red Queen have to run but apparently do not get anywhere since the ground moves backward underneath them at an equal rate. The Red Queen explains: "Now here, you see, it takes all the running you can do, to keep in the same place" [2].

Figure 12.2 visualizes the Red Queen effect. A species is assumed to be driven towards a local optimum by the selective pressure. Initially, it is located on one side of a local optimum. As the species climbs towards the top, the landscape changes. Due to this change of the landscape, it appears as if the species has not succeeded in climbing the hill despite being driven to follow the local adaptive gradient. Measuring progress in coevolutionary scenarios can therefore be problematic [3, 21]. In open-ended evolutionary systems, e.g. self-reproducing programs [1, 7, 31, 32], we might hope that continued evolutionary change results in continued progress or improvement. But, it is not guaranteed that coevolutionary interactions that lead to continued change, i.e., Red Queen dynamics, will necessarily result in continued improvement, i.e., as implied in the term 'arms race'. Over-specialisation and intransitive relationships may result in continual cycling through trait space that fails to produce improvement in any absolute sense [44].

The conditions that lead to either Red Queen dynamics or conversely conditions that produce a stable attractor where no further change (let alone progress) is possible, are therefore of great interest. Van Valen [41] originally suggested that species may exhibit Red Queen dynamics, producing continuous evolutionary change, without any extrinsic changes in environment. Maynard Smith [38] pointed out, however, that Van Valen's model depends on the assumption of a zero-sum game where an evolutionary change in one species that improves its fitness necessarily results in an equal decrease in fitness in total over all other species. Stenseth and Maynard Smith [39] argue that Van Valen's assumption of a zero-sum game does not necessarily follow from the assumption of a fixed amount of total resources. Their approach separates the notions of a fitness interaction between two species from the notion of how a change in one species affects the ‘lag load’ of another species. The lag load is a measure of how far a species is from a local adaptive peak - and the presence of a non-zero lag load implies continued evolutionary change. Crucially, they argue that a genetic change conferring a fitness increase in one species may produce either a net increase or a net decrease in the lag-load of all species taken together. In other words, even with constant total resources, there are ways to utilise resources efficiently and ways to utilise them inefficiently, and this means that the underlying game is not zero-sum. Then, for non-zero-sum games there remains the possibility that, for linear interactions, the resulting dynamics may be either contractive leading to stable coexistence with each species at a local peak, or divergent where some species may lag increasingly far behind the local peak, possibly leading to extinctions. However, in the case where the relationship between lag load and change in lag load is non-linear then, as before, a Red-Queen dynamics of stable change is possible. Intuitively, this would be the case if, when lags are small and species are near local optima, most evolutionary changes in one species produce large increases in the lags of others species (this follows from geometric arguments - i.e., from a point near a local optima, most directions lead down) whereas, when lags are large most evolutionary changes in one species produce relatively little increase in the lags of other species. In this case, a stable but non-zero amount of total lag is expected, thereby conferring continued evolutionary change.

As Stenseth and Maynard Smith suggest, whether a coevolutionary scenario has the necessary conditions for Red Queen dynamics or for stasis is ultimately an empirical matter. But here we introduce a relatively simple mechanistic model of inter-species interactions to investigate contributing factors. In this model, different arrangements of species utilise resources with different degrees of efficiency; e.g., in an efficient arrangement, each species utilises a niche that has as little overlap as possible with other species, whereas if all species attempt to utilise the same resource this is relatively inefficient. The underlying evolutionary game implicit in this model therefore has non-zero-sum properties. Our model also has the potential to exhibit the non-linear relationship between lag load and change in lag load described by the geometric intuition above. It therefore seems plausible that Red Queen dynamics might occur in accord with the arguments of Stenseth and Maynard Smith. However, we observe that under some conditions our model nonetheless results in evolutionary stasis - or at least, exhibits qualitatively different modes of behaviour with very different amounts of evolutionary change. We investigate which parameters of the model determine these distinct modes of behaviour and we find that which of these modes is observed depends on the latency between the movement of a species in trait space and its effect on available resources. Specifically, if ecological change is nearly instantaneous compared to evolutionary change, stasis results; but conversely, if evolutionary timescales are closer to ecological timescales, such that ecological resource depletion is not instantaneous on evolutionary timescales, then Red Queen dynamics result. We also observe that in the stasis mode, the overall utilisation of resources by the ecosystem is relatively efficient, with diverse species utilising different niches, whereas in the Red Queen mode the organisation of the ecosystem is such that species tend to clump together competing for overlapping resources. These models thereby suggest a link between the issues of change and stasis discussed by Stenseth and Maynard Smith and matters of ecological diversity and organisation. In the same way, these models also suggest some basic conditions that influence the organisation of inter-species interactions and the balance of individual and collective adaptation in ecosystems, loosely corresponding to the opposites of cooperative and competitive approaches to artificial coevolution.

In the next section, we describe the details of our modelling approach, and the various conditions and parameters that we investigate.

12.3 A Dynamically Deforming Fitness Landscape

We model the mean phenotype of a species as a point in an n-dimensional continuous trait space. An n-dimensional vector is used to represent the position of each population. For example, $n = 1$, describes a one-dimensional fitness landscape [9] and each species is represented using a single scalar value or quantitative trait. Natural selection moves the population mean in the direction that climbs the local gradient in the fitness landscape. Here this is modeled abstractly as a gradient ascent or hill climbing process (see Figure 12.3) on a fitness landscape [46].

This approach simplifies the evolutionary dynamics of each species making it easier to focus on the evolutionary interaction between many species and the interaction between evolutionary and ecological dynamics in a complex ecosystem. This

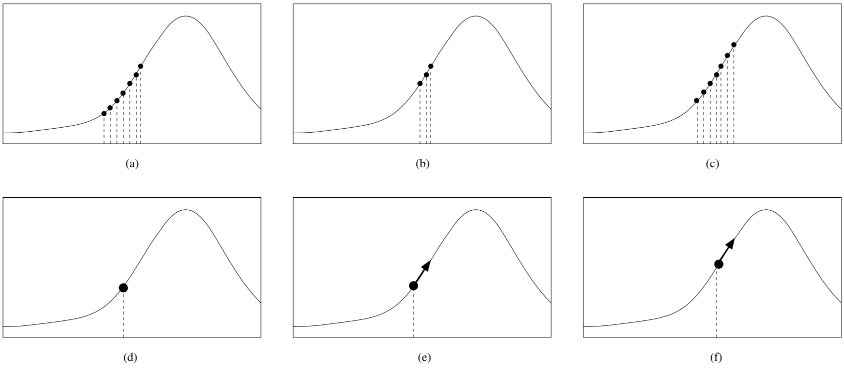


Fig. 12.3 Modeling the evolutionary process. (a), (b), and (c) show a population climbing towards a local optimum on a one-dimensional fitness landscape. (a) Initially, a population of individuals is located on the side of an incline. (b) only highly fit individuals survive (c) the remaining individuals then produce offspring and the new population is now located at a higher position in fitness space. (d), (e), and (f) show the same population represented as the population average. (d) a single point represents the population of individuals (e) gradient of the fitness landscape (f) the population average is moved up the incline depending on the measured gradient. The population average is now located at a higher position in fitness space.

approach is justified on the assumption that the genetic diversity within any one species is insignificant (at least, compared to the genetic diversity between species), and that the distribution of genetic variation remains unimodal (speciation is excluded), as for example, under 'strong selection weak mutation' assumptions [14]. Intuitively, we might imagine that the exact relationship between the local fitness gradient and the rate of adaptation of the species could be important in affecting the type of dynamics we observe in the ecosystem of multiple interacting species. Below we define three different methods that we investigated for updating the position of a species. In all cases the direction of change is determined by the direction of increasing fitness in each dimension of the fitness landscape - but the different methods affect the rate of evolution: 1) constant rate evolution, 2) rate of evolution linearly proportional to fitness gradient, 3) rate of evolution determined by rate of change of fitness gradient.

The first of these, constant rate, is appropriate where evolution is mutation limited. That is, regardless of how steep the fitness gradient is, the maximum rate of evolution is limited by the availability of variation that can respond to it, and the generation of genetic variation is unaffected by the selective pressure. Suppose that a species is located at position $x(t)$ at time step t. If $f(x,t)$ denotes the height of the fitness landscape at position x at time step t, then the simple update rule would be

$$\dot{x}(t) = \begin{cases} -1 & \text{if } \frac{\partial}{\partial x} f(x,t) < 0, \\ 0 & \text{if } \frac{\partial}{\partial x} f(x,t) = 0, \\ 1 & \text{if } \frac{\partial}{\partial x} f(x,t) > 0 \end{cases} \tag{12.1}$$

where $\dot{x}(t)$ is the velocity of the species. Thus, using update rule (12.1), fitness only controls the direction of movement.

In more general conditions the rate of evolution in a population will be sensitive to the magnitude of the selective coefficients. Our second, linearly proportional, update method thus sets the velocity of the species proportional to the gradient of the fitness landscape in Equation (12.2). This is consistent with a conventional population genetic model where the rate of change is proportional to fitness variance.

$$\dot{x}(t) = \alpha \frac{\partial}{\partial x} f(x,t) \tag{12.2}$$

Here α is the factor of proportionality.

The third update model is a logical extension. Specifically, Equation (12.3) integrates the fitness gradient over time such that the rate of evolution is a function of the second differential of fitness rather than the first differential or a constant. Hence the population responds to the gradient of the current position as well as to the gradient of the previous time step.

$$\dot{x}(t) = \alpha \left(\frac{\partial}{\partial x} f(x,t) \right) + \beta \dot{x}(t-1) \tag{12.3}$$

A momentum term such as this is known to have interesting effects on many kinds of dynamical systems. In an evolving population this type of dynamic could result if recent selection altered the ability of the population to respond to subsequent selection. For example, Pavlicev et al. [27] show that the action of past selection can alter the evolvability of the population by increasing genetic variation in the direction of selection.

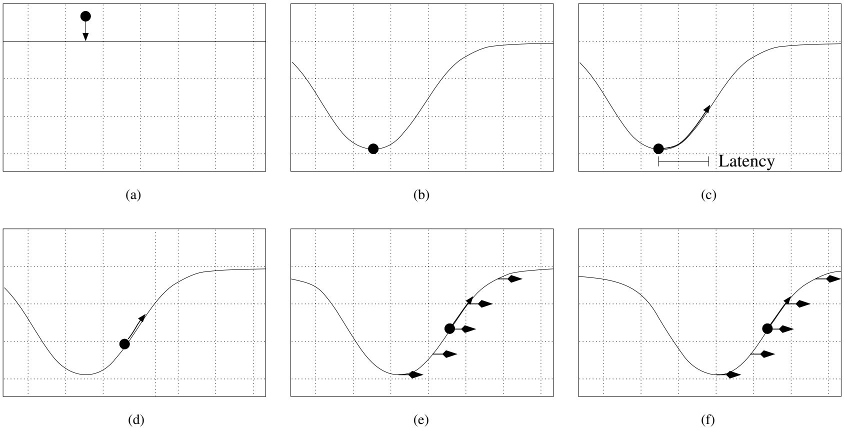


Fig. 12.4 Deformation of the fitness landscape. (a) Initially, a flat fitness landscape represents a uniform distribution of resources. (b) Each species placed on this landscape deforms the landscape in its vicinity as if by depleting available resources. This is similar to the deformation of a rubber sheet by a point mass. (c) Natural selection moves each species up the local fitness gradient. (d) In general, the ecological depletion of resources caused by the species is not necessarily immediate. (e) The deformation of the landscape caused by the species thus follows the position of the species with a latency period of several time steps. (f) As the position of the species moves away from the depression in the fitness landscape, and the deformation of the landscape responds to the position of the species, this leads to a condition of stable evolutionary change (Red Queen dynamics).

In our model all species coexist in a shared fitness landscape. The fitness landscape represents, abstractly, the availability of resources in a continuous niche space and the height of each point in the landscape is thus modified dynamically in response to the location of species. Each species thus has an impact on the shape of the fitness landscape in its local vicinity. Intuitively, each species deforms the landscape much like a point mass placed on a rubber sheet. In principle, the shape of this deformation reflects the distribution of phenotypes around the population mean. Here we assume a Gaussian population distribution and hence a corresponding Gaussian deformation. More generally, our model could be extended to investigate a positive effect on the fitness landscape, but here we assume only negative effects as if by competition for resources. (This negative impact on fitness is similar to fitness sharing or crowding methods which are used in evolutionary algorithms to promote diversity [15].) We assume that the shape of the population distributions and the resource depletion that they confer is equal for all species, but the more species that occupy a certain point on the fitness landscape the larger the deformation.

If ecological dynamics are very rapid compared to evolutionary change then the effect of a species on its environment is effectively immediate. This assumes, in effect, that resources are always at equilibrium before the next evolutionary change occurs. This might be appropriate when modelling physical resources such as space or light. More generally, the ecological response of a niche to the intrusion of a new species may not be immediate. A latent effect on local fitness might be appropriate when modelling biotic resources (species of the lower trophic level) that are themselves subject to nonlinear growth and decay. Rather than model the dynamic behaviour of resources explicitly, here we simply investigate the effect of a delay or latency term in the effect that each species has on the depression of the fitness landscape. A latent effect takes some time steps before it becomes apparent. Hence, in our model, we basically have two different modes. The deformation is either placed at the same position as the species (immediate effect) or it is placed at the position where the species was located some time steps ago (latent effect). For simplicity, we assume the same latency value for all species/locations in trait space.

In the latent as well as the non-latent model each species climbs the local fitness gradient. However, in the latent model, it takes a certain time before a local optimum is depressed by the presence of the species. Once this happens, the fitness of the species is no longer optimal and it needs to adapt to a new optimum. Figure 12.5 illustrates what happens if two species climb towards the same local optimum.

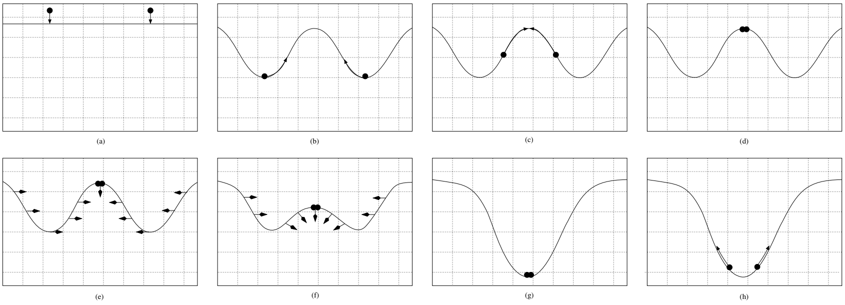


Fig. 12.5 (a) Two species are placed next to each other. (b) Initially, they deform the landscape surrounding them which causes a local optimum which is located in between the two species. (c) Both species are climbing towards the same local optimum. (d) If the latency is rather long, then both species are able to reach this optimum before the landscape deforms. Once they have reached this optimum, they have to stay put. (e-f) After a while the deformation caused by the exploitation of local resources follows them. This significantly reduces their fitness. (g-h) From the bottom of the valley, they may climb up again to either side of the valley they created.

12.4 Experimental Results

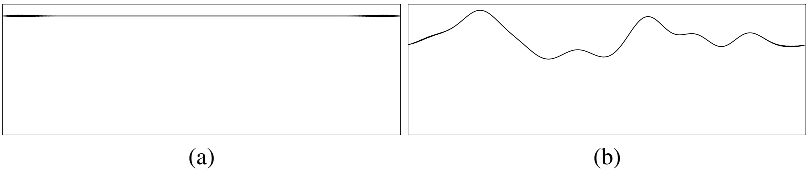
Experiments were performed on a one-dimensional and on a two-dimensional fitness landscape with circular boundary conditions. We will see below that interesting behavior emerges as we switch from a one-dimensional to a two-dimensional world. The source code for these experiments can be downloaded from the second author's web page[1]. The experiments are also available as MPEG as well as AVI movies.

12.4.1 Experiments on a One-Dimensional Landscape

First we investigated the dynamics of competitive coevolution on a one-dimensional landscape. We have varied the update rule, the latency period and the type of environment used. The parameters for these experiments are shown in Table 12.1. The different settings illustrate interesting qualitative features of the dynamics. In particular, there are three different dynamic regimes that these experiments exhibited - static, cyclic, and races - as we shall discuss.

Table 12.1 Nine different experiments were carried out on a one-dimensional fitness landscape. We have varied the update rule, the latency period and the type of environment. The most important parameter determining the behavior of the species is the latency period.

Experiment	Species	Update Rule	Latency	Hills	Observed Behavior	Figure
1	10	Equation (12.1)	0	0	cyclic (stasis)	12.7
2	10	Equation (12.1)	50	0	clumped shift	12.8
3	10	Equation (12.2)	0	0	stasis	12.9
4	10	Equation (12.2)	50	0	cyclic or arms race	12.10 & 12.11
5	10	Equation (12.3)	0	0	stasis	same as 12.9
6	10	Equation (12.3)	3	0	cyclic	12.12
7	10	Equation (12.3)	50	0	arms race	12.13
8	10	Equation (12.3)	0	25	stasis	12.14
9	10	Equation (12.3)	50	25	arms race	12.15


(a)

(b)

Fig. 12.6 (a) Flat fitness landscape. (b) Non-flat fitness landscape with random variations.

[1] `http://stubber.math-inf.uni-greifswald.de/~ebner/`

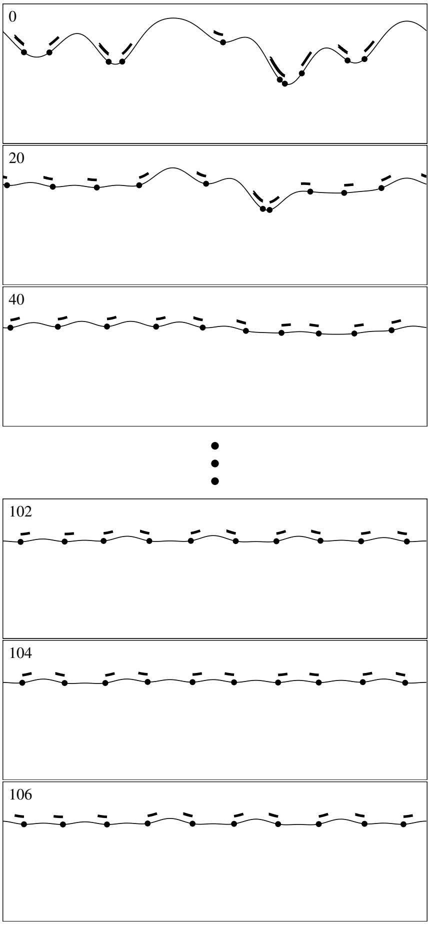


Fig. 12.7 Experiment 1: The velocity is determined by the sign of the environment's gradient (constant evolution rate). The parameters (shown in Table 12.1) lead to small cyclic behavior. The species keep oscillate back and forth. The species' current velocity is illustrated by the lines directly above the species. The number in the upper left corner shows the current time step.

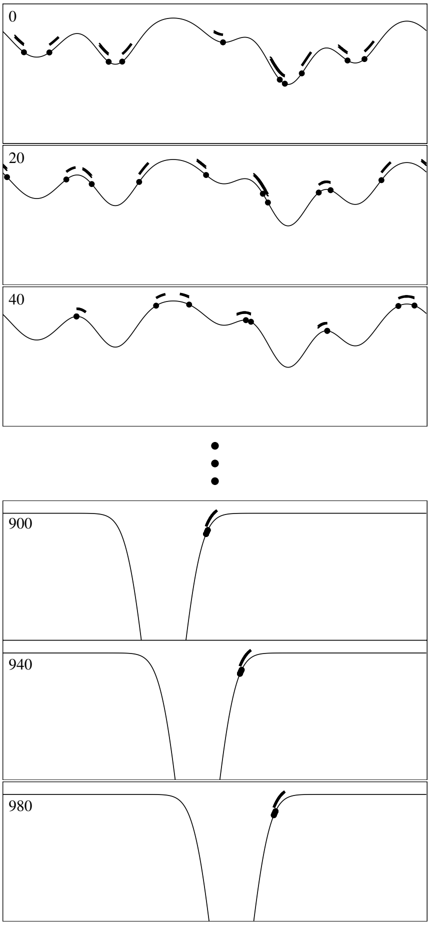


Fig. 12.8 Experiment 2: Velocity is set according to the sign of the landscape's gradient (proportional rate). The parameters (shown in Table 12.1) lead to a 'clumped shifting' behavior.

We have used 10 species throughout our experiments on the one-dimensional fitness landscape. Experiments were carried out using different environments, specifically, a uniform distribution of resources creating an initially completely flat landscape (Figure 12.6(a)) and a non-uniform distribution of resources where some Gaussian 'hills' were distributed randomly over the landscape (Figure 12.6(b)).

We experiment with all three update rules and vary the latency period. We observe that the latency period is the crucial parameter in determining the different modes of behaviour that the ecosystem exhibits. Specifically, an arms race only results if we have a non-zero latency period.

For Experiment 1, update rule (12.1) was used to update the direction of the species' evolution (with constant rate). A completely flat environment and no latency was used, i.e. the depletion of resources happens immediately. The resulting behavior is shown in Figure 12.7. The species spread over the landscape and keep moving back and forth because the best place to be is the point that is most distant from other species at the current time. However, as soon as a species moves in the direction the best place to be is behind it in the other direction. Basically, evolution has come to a halt. The species no longer move over the landscape.

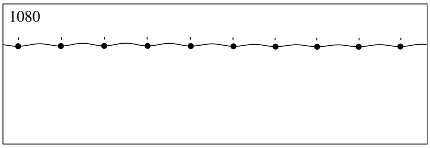


Fig. 12.9 Experiment 3: For this experiment, we use a velocity update rule where velocity is directly proportional to the gradient of the environment. As we can see from the small dots located directly above each species, the species are almost stationary. As a result, no further evolutionary change or improvement is possible and evolutionary space is only partially explored.

When we set the latency to 50 (Experiment 2), we observe the clumped shifting behavior as shown in Figure 12.8. First, the species climb towards a local optimum. Then the deformation increases. Eventually several or even all of the species end up very close together. The clump of species causes a large depression on the fitness landscape and continually moves in one direction. The entire evolutionary landscape is explored. Note that at any point in time there are many areas of the fitness landscape where resources are not being utilised at all (i.e., there are no species in those locations and no depression of the landscape).

For Experiment 3 we used the update rule (12.2) to update the velocity of the species. The parameter α was set to 10. No latency, i.e. the depletion of resources is immediate, and a completely flat environment was used. As a result, the species spread over the entire landscape utilising all areas of niche space equally. This leads to an almost stationary state with little movement as shown in Figure 12.9. Once the species are spread over the entire fitness landscape, no further evolution or improvement is possible. This can also be viewed as each species having found a niche where they do not interfere very much with the other species.

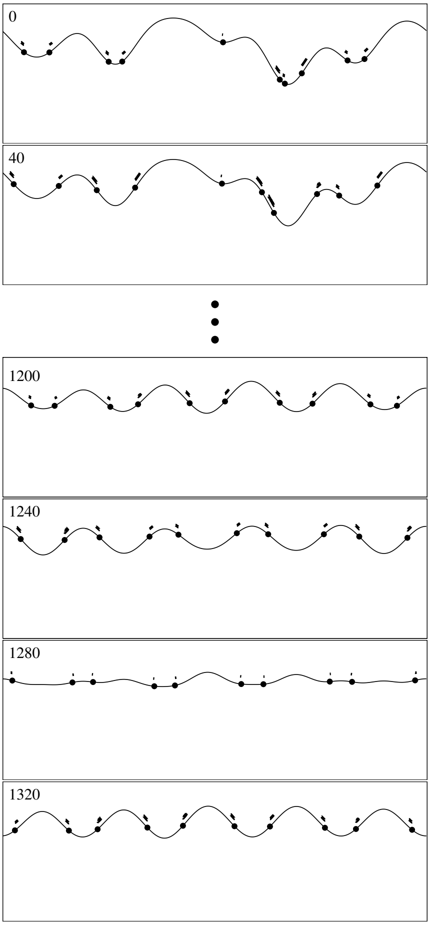


Fig. 12.10 Experiment 4(a): Here, we use a velocity update rule where velocity is always equal to the gradient of the environment. The parameters of this experiment (as shown in Table 12.1) may lead to a cyclic behavior as is shown here.

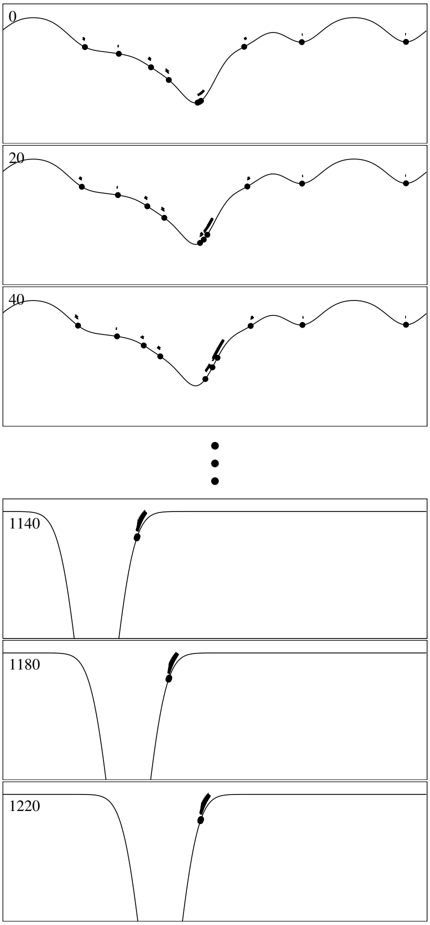


Fig. 12.11 Experiment 4(b): The parameters of Experiment 4 may also lead to an arms race

For Experiment 4 we used update rule (12.2). The latency parameter was set to 50, i.e. the depletion of resources does not happen immediately. We have used a completely flat landscape/uniform environment for this experiment. Two qualitatively different behaviors were observed. One outcome of this experiment, a cyclic behavior, is shown in Figure 12.10. First, the species spread out over the landscape. Local optima are created in between two species due to the delayed impact on the fitness landscape. The species try to climb towards these optima. However, once they have reached them, the depletion of resources sets in and the species are moved to a lower position on the fitness landscape. New local optima have formed and the species again have to climb towards these optima. This process repeats indefinitely. Evolution essentially has come to a halt. Depending on how the species are distributed initially over the fitness landscape, we also observed an arms race (see Figure 12.11). This happens if several species are located very close to each other. As they sweep over the landscape, they collect more and more species in their arms race until eventually all species are included.

Update rule (12.2) as well as update rule (12.1) result in a rather slow movement of the species. Update rule (12.3) results in a much faster movement of the species. Experiment 5 is the first to use update rule (12.3) with parameters $\alpha = 9$ and $\beta = 0.9$. No latency and an initially flat landscape was used. We again observe the behavior which was observed in Experiment 3 (see Figure 12.9). The species spread-out over the landscape and remain almost stationary. Evolution has come to a halt.

For Experiment 6 we only changed one parameter slightly. Specifically, here a latency of 3 is introduced. This leads to a cyclic behavior as is shown in Figure 12.12. Initially, the species spread out over the entire landscape. We then observe groups of two species which are located next to each other. Both of them try to climb towards the local optimum located in between them. Due to the non-zero latency parameter, the deformation follows the species with a little delay. This causes the depression of the optimum to which they have climbed and the creation of new local optima at the exact same position where they started out. This leads to cyclic behavior because the species now try to climb these optima and they end up in the exact same position where they had started. Once this state has been reached, no further improvement is possible and the evolutionary space is only partially explored.

Experiment 7 examines a latency of 50. Here we observe an arms race between the species. Figure 12.13 shows the results obtained for a typical run. From their initially random distribution, two species that happen to lie on the same side of a deformation try to climb towards the same local optimum. After a while the deformation follows them. As they try to escape from the local valley, more and more species are caught by this deformation. Eventually an arms race results in which all species are involved.

We also experimented with a non-uniform distribution of resources creating an initial fitness landscape that is not flat. For this environment, 50 Gaussian peaks are distributed over the fitness landscape and summed (Figure 12.6(b)). As before, the landscape is then deformed by the positions of the species. This landscape was used for Experiments 8 and 9.

Experiment 8 uses the same parameters as Experiment 5 except that the initial fitness landscape is not flat. The species again spread out over the entire landscape. Due to the fact that the latency factor was set to 0, the influence of a species on their environment happens immediately. Hence, the non-flat landscape is deformed slightly by the species. The species nevertheless adapt to their environment and an almost stationary state as shown in Figure 12.14 results. Once this attractor has been reached, evolution comes to a halt. Note that the non-uniform distribution of species matches the availability of resources in the initial fitness landscape resulting in an ecosystem organisation where no resources are left unused.

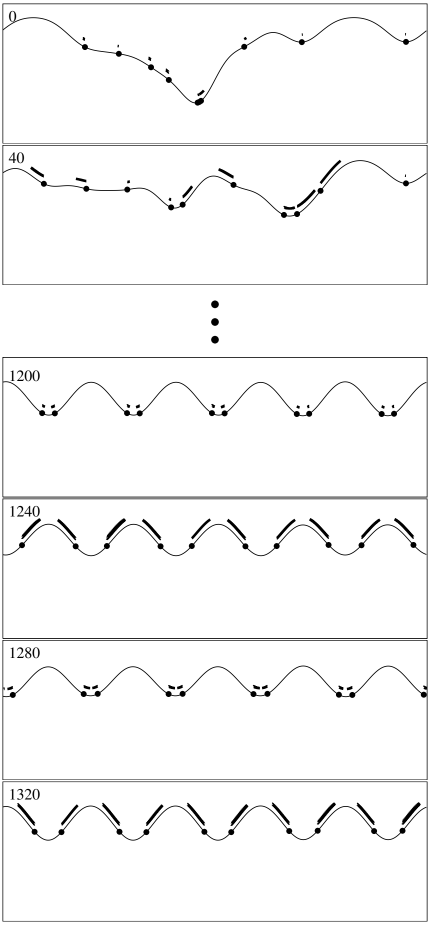


Fig. 12.12 Experiment 6: The cyclic behavior shown here is the attractor of Experiment 6. The species spread out over the landscape. Groups of two species try to climb towards the local optimum which is located in between them. Due to the latency, the depletion follows the species after a delay. The species then find themselves with low fitness and new fitness gradients, but this returns them to a position they have been to before. Once this attractor is reached, no further improvement is possible.

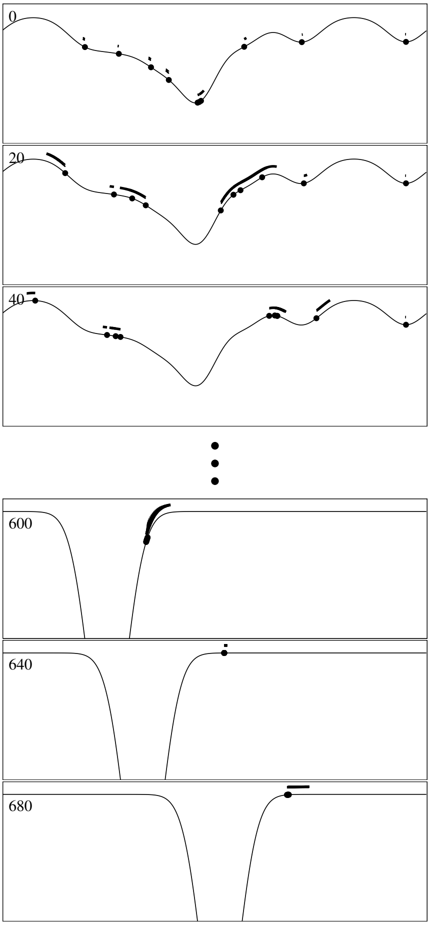


Fig. 12.13 Experiment 7: The parameters of Experiment 7 (as shown in Table 12.1) lead to an arms race

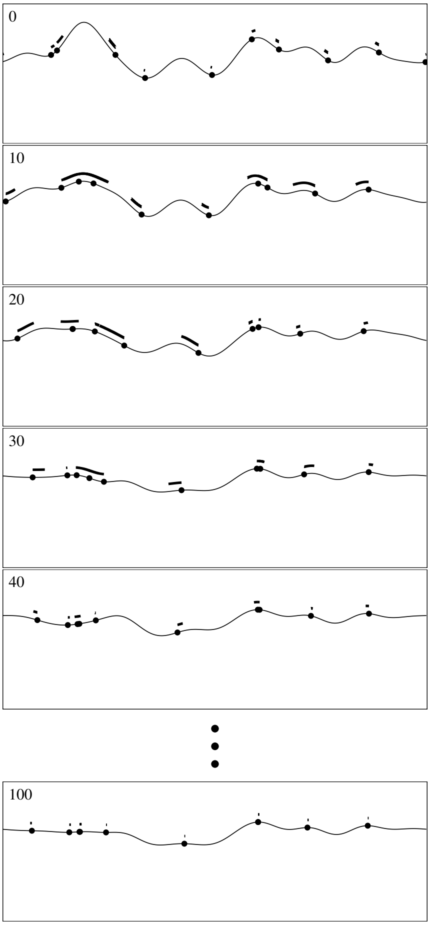


Fig. 12.14 Experiment 8: A non-flat fitness landscape was used. The shape of this fitness landscape can be seen in Figure 12.6(b). Initially, the species spread out over the fitness landscape. They try to climb towards a local optimum avoiding the negative influence of other species. After some time a stable attractor is reached. All species become stationary. Once this has happened, no further improvement is possible.

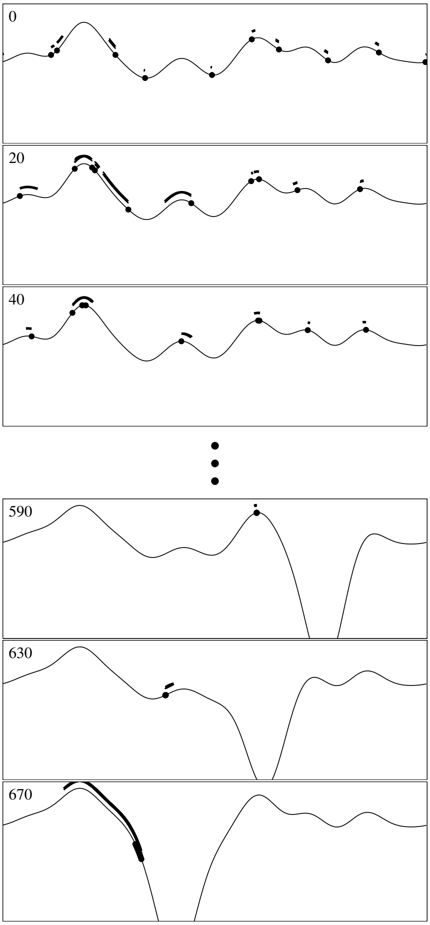


Fig. 12.15 Experiment 9: This experiment uses the same parameters as Experiment 7 except that a non-flat fitness landscape was used. Due to the large latency, the species clump together setting up an arms race. The species explore the entire fitness landscape despite the non-uniform distribution of resources, but at any one point in time, many niches are left unused.

For Experiment 9 we have increased the latency to 50. All other parameters were exactly the same as for Experiment 7. We observe that the species first adapt to their environment by climbing towards a local optimum. However, once this optimum has been reached, depletion of the resources kicks in and the species are no longer located on the optimum. They have to adapt and climb towards a new optimum. An arms race sets in where the entire evolutionary space is explored. The species move over the entire landscape, i.e. all local optima are explored (Figure 12.15).

We conclude this set of experiments by noting that with latency an arms race is possible. Without latency an arms race does not happen for any of the conditions we tested. This is robust for all update rules, i.e. models of evolutionary rates.

12.4.2 Experiments on a Two-Dimensional Landscape

Conceivably, the results obtained with a one-dimensional landscape, although simple to simulate, might introduce special symmetries that are not representative of trait spaces with dimensionality higher than one. Intuitively, one might put it like this - pushing a ball up an incline with a pointed stick is much easier in a pipe than on a plane. That is, when one species moves away from another species in a

Table 12.2 Parameter settings which were used for the experiments on the two-dimensional landscape.

Experiment	Species	Update Rule	Latency	Hills	Observed Behavior	Figure
10	30	Equation (12.1)	0	0	shift	12.17
11	30	Equation (12.1)	50	0	continued motion	12.18
12	30	Equation (12.2)	0	0	stasis with small drift	12.19
13	30	Equation (12.2)	50	0	cyclic with drift	12.20
14	30	Equation (12.3)	0	0	stasis with drift	12.21
15	30	Equation (12.3)	3	0	cyclic with drift	12.22
16	30	Equation (12.3)	50	0	arms race	12.23
17	30	Equation (12.3)	0	100	stasis	12.25
18	30	Equation (12.3)	50	100	arms race	12.26

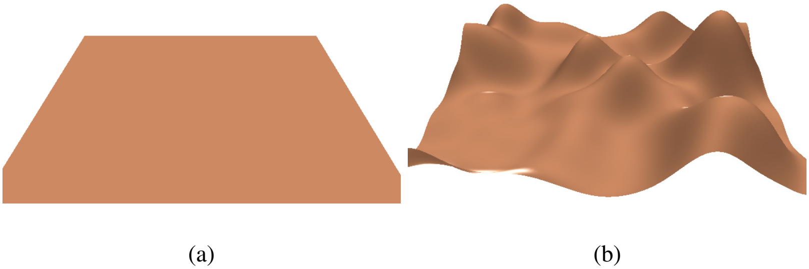


Fig. 12.16 (a) Flat 2D fitness landscape. (b) Non-flat 2D fitness landscape with random variations.

one-dimensional space, the 'evader' has no option but to move in exactly the same direction as the 'pursuer'. This might contribute to a stability in the arms race that is somewhat artificial. In contrast, in a two-dimensional space, the direction of movement of two species may diverge and one species may move 'past' another.

We therefore perform the same experiments on a two-dimensional landscape. In the two-dimensional experiments it is even more difficult to convey the dynamics of the experiments using static figures than it is in previous figures using a one-dimensional environment. The interested reader is referred to the movies or the programs which are available for download[2]. Table 12.2 summarizes the parameters that we have used for our experiments in two dimensions. Since the two-dimensional landscape is bigger than the one dimensional landscape, we have increased the number of species to 30. Figure 12.17 shows the results for Experiment 10. Experiments 10 through 16 use a flat two-dimensional fitness landscape shown in Figure 12.16(a). The results for the experiments on the two-dimensional landscape are qualitatively similar to the results which were obtained for the one dimensional case. However, in some cases more complex behavior results due to the fact that additional directions are available in which a species can move. For example, in Experiment 2, on the one-dimensional fitness landscape, we experienced a clumped shifting behavior of all species. For the two-dimensional case (Experiment 11) we experience continued motion because the species are not constrained to move along a single line. The results of Experiment 11 are shown in Figure 12.18. The results of Experiments 12 and 13 are shown in Figures 12.19 and 12.20 respectively. For Experiment 12, we observe a resulting state of stasis as in the one-dimensional case. Experiment 13 results in a mix between cyclic behavior and continued motion.

Experiments 14 through 18 produced the most interesting behaviors. Experiment 14 resulted in a state of stasis with small drift. A snapshot of the resulting state is shown in Figure 12.21. For Experiment 15 the latency was increased to 3. With a latency of 3 we again observed cyclic behavior with drift (Figure 12.22). Because of the two dimensions, the individuals also move through space. When we use a latency of 50 (Experiment 16) we again obtain an arms race similar to the one obtained in Experiment 7 on the one-dimensional fitness landscape. In comparing the two corresponding experiments (Experiment 7 and 16) in one- and two- dimensional spaces we can see how they differ. In both environments, the species climb towards a local optimum, and the landscape responds to the presence of these species, the local optimum has turned into a valley. On a one-dimensional environment the species have only two possible directions in which they can leave this local depression – left or right. However on a two-dimensional landscape, the individuals have an infinite number of directions they can move in. In climbing out of the local depression, the species spread-out across the rim Figure 12.24. If the separation between the species is of sufficient extent, then two separate depressions may be created each of which may result in its own arms race further increasing the separation between them. We expect that this possibility would be further increased in trait-spaces with higher dimensionality than two. It is interesting to note that our model does not involve island

[2] http://stubber.math-inf.uni-greifswald.de/~{}ebner/

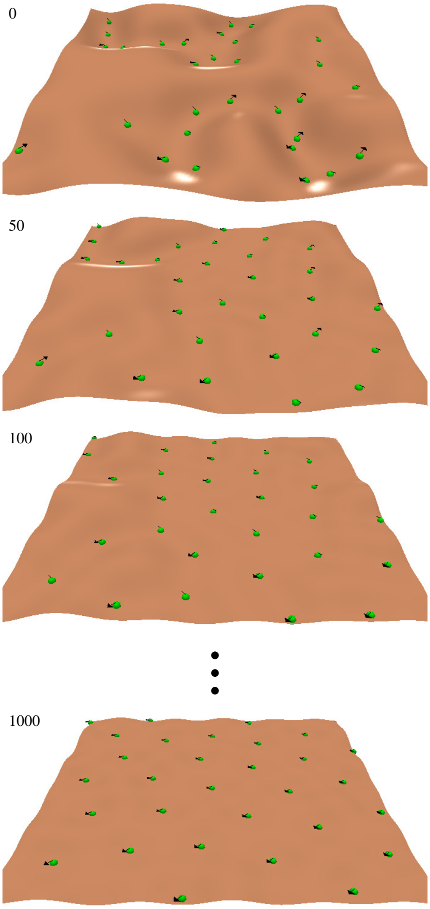


Fig. 12.17 Experiment 10: As in the one-dimensional case one can observe a slow synchronous shift of the species

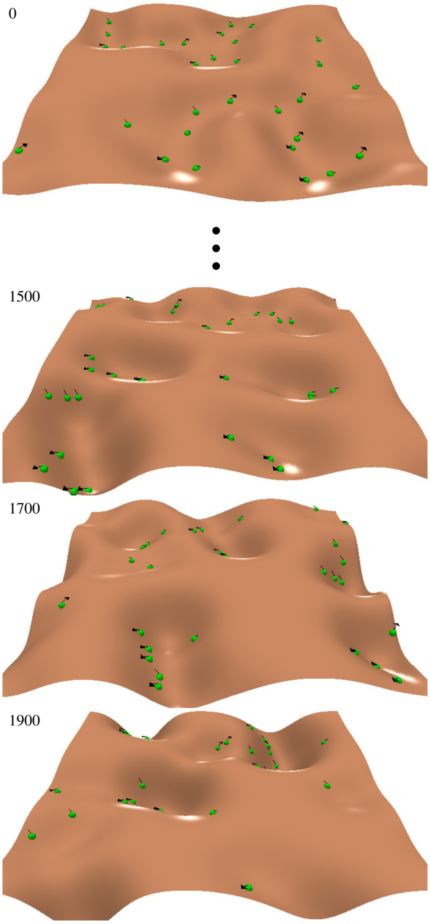


Fig. 12.18 Experiment 11: We observe continued asynchronous motion. The results are again similar to the one-dimensional case. The asynchronous motion results from the increase in the number of directions a species can move.

models or other means of artificially segregating the species, i.e. reproductive isolation [16] or mate preference due to marker traits [6, 35]. Here, sympatric speciation [40], speciation without geographic of physical isolation, is a result of the coupled dynamics of the species [25]. However, this is nonetheless an impoverished form of ecological diversity compared to the results of experiments without latency.

Experiment 17 and 18 use a non-flat fitness landscape shown in Figure 12.16(b) with 100 Gaussian hills distributed randomly over the landscape. A latency of 0 is used for Experiment 17. We obtain a state of stasis with oscillations which is shown

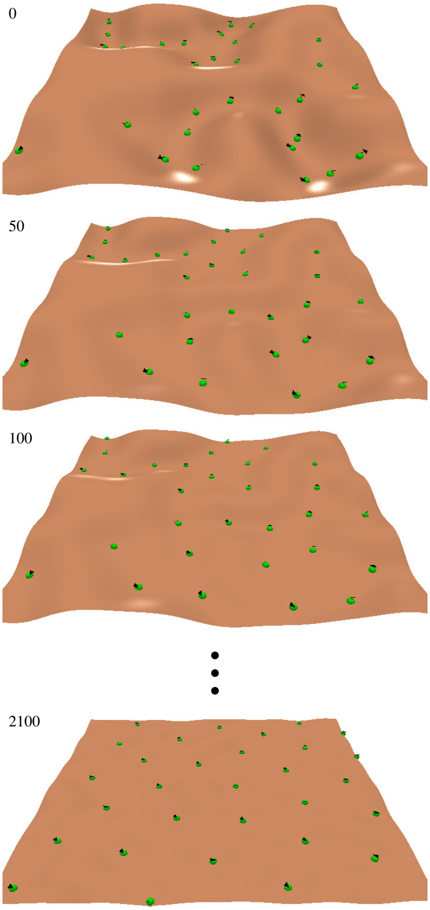


Fig. 12.19 Experiment 12: We again observe stasis with drift as in the one-dimensional case

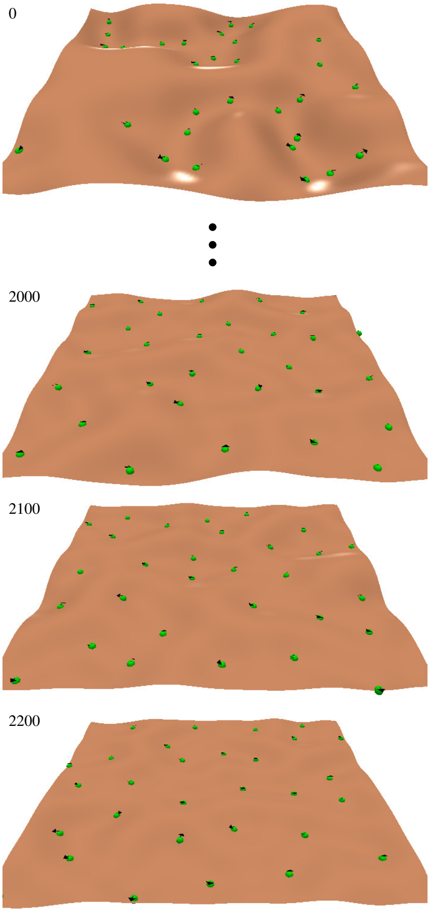


Fig. 12.20 Experiment 13: We observe a mixture between cyclic behavior and continued motion

in Figure 12.25. The species distribute over the entire landscape climbing hills as long as it is advantageous to them and in so doing they equalize the landscape. The ecosystem therefore arrives at an organisation of species which uses the available resources efficiently. In contrast, when we use a latency of 50, we again observe Red Queen dynamics as shown in Figure 12.26. Latency thus causes an ecosystem organisation where resources are not used efficiently - the same resources are being used by many species whilst others are not being used.

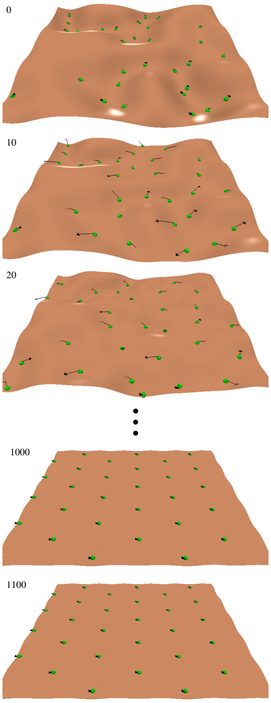


Fig. 12.21 Experiment 14: With a latency of 0 evolution eventually stops and a state of stasis is reached

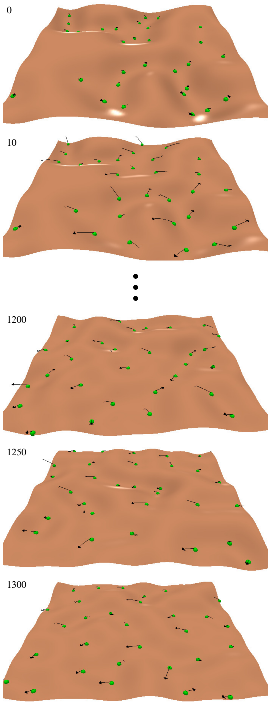


Fig. 12.22 Experiment 15: A latency of 3 produces cyclic behavior

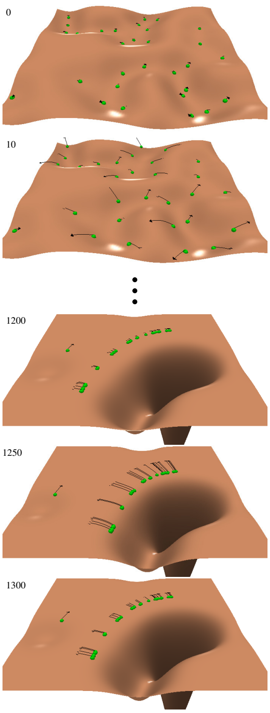


Fig. 12.23 Experiment 16: With a latency of 50 we obtain an arms race between different species

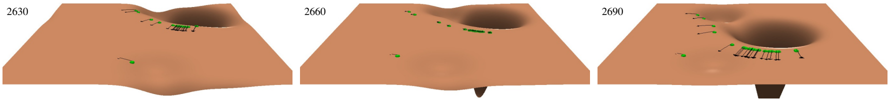


Fig. 12.24 Spontaneous speciation occurs during Experiment 16. The species spread along the rim of the depression. This causes the creation of two separate sub-sets of species engaged in their own arms races.

12.5 Discussion and Conclusions

12.5.1 Stasis, Change and Improvement

The investigations above have examined a model of an ecosystem where multiple species compete for shared resources. We studied the conditions in this model that produce Red Queen dynamics. The reasoning of Stenseth and Maynard Smith suggests that such a model, having a non-zero-sum game and where the relationship between the lag load (distance from local peaks) and the rate of increase of lag load

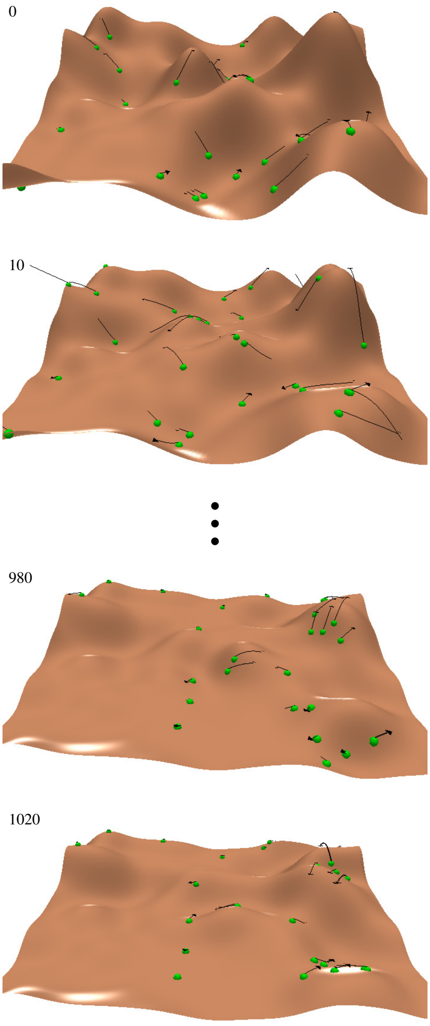


Fig. 12.25 Experiment 17: A non-flat environment and a latency of 0 was used. The species again spread over the landscape to avoid the negative influence of other species and to exploit and fitness advantages present in the landscape.

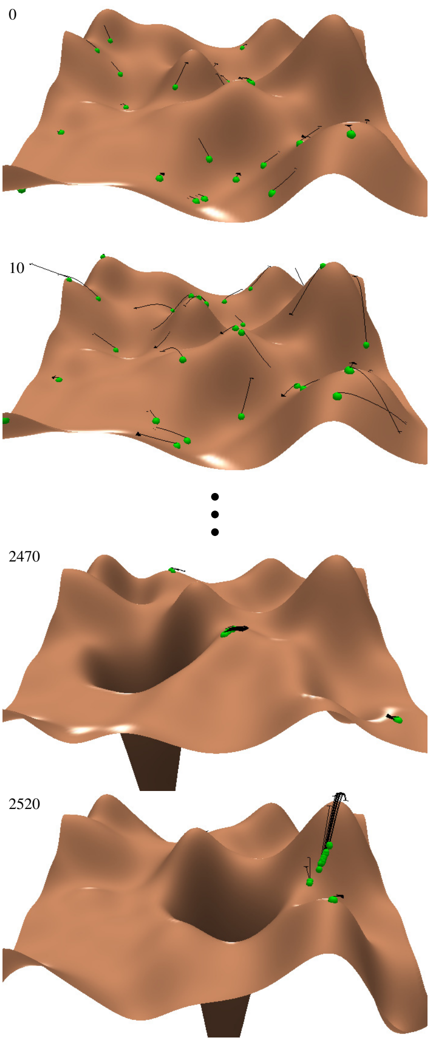


Fig. 12.26 Experiment 18: An arms race occurs. The species eventually sweep over all local optima, but at any one point in time their use of available resources is inefficient.

is non-linear in this manner, can readily produce stable Red Queen dynamics. Our model provides a specific mechanistic illustration of the scenario they describe and our observations indicate that Red Queen dynamics can be readily exhibited. However, we also observe that within these conditions the same model can also produce evolutionary stasis. We find that the critical parameter in determining this distinction is the latency with which evolutionary changes affect ecological resources and thus affect other species. Specifically, if ecological changes are much more rapid than evolutionary changes, such that the ecological response to an evolutionary change is effectively instantaneous, then an evolutionary change in one species has (via its effect on the fitness landscape) an immediate effect on the selective pressure acting on other species and evolutionary stasis is possible. However, if evolutionary timescales are closer to ecological timescales, such that the ecological response to an evolutionary change is not immediate, then we often observe species engaging in Red Queen dynamics.

We investigated several other factors that may affect such dynamics; the relationship between rate of evolution and the slope of the fitness landscape, the dimensionality of the trait/niche space, and whether the intrinsic distribution of resources is uniform or non-uniform. We find that the relationship between latency and the Red Queen dynamics is quite robust to these factors. However, there were some differences between one- and two-dimensional spaces and it seems plausible that higher dimensional spaces would amplify these differences.

In the case of a uniform distribution of resources, we cannot really say that the presence of continued evolutionary change corresponds to improvement – all parts of the trait space have intrinsically equal value. However, it is notable that the continued evolutionary change causes species to explore the entire space. In the case of non-uniform resource distributions, some areas of trait space are intrinsically more valuable than others. In this case, the Red Queen dynamics are capable of pushing species off local optima and forcing them to explore other peaks in the landscape. There is however, no guarantee that any subsequent peak is an intrinsic improvement over the previous peak – it is only a relative improvement at this point in time because of the transient resource depletion. The relationship between continued evolutionary *change* and an arms race that produces continued *improvement* therefore remains problematic.

12.5.2 Diversity and Efficiency

Moving beyond the questions of change and stasis as discussed by Stenseth and Maynard Smith, we find that there is a strong link between these different dynamical outcomes and features of ecological diversity and efficiency. The natural result of ecological competition is to produce a selective pressure to diversify – to utilise resources that others are not using. But conversely, when species evolve without competition on the same landscape the tendency is for them to follow the same selective gradients and therefore converge on the same high fitness regions. We observe that when ecological dynamics respond rapidly to evolutionary change (latency= 0), species diversify onto different resources as expected. But when ecological responses are delayed (latency> 0), species clump together competing for resources in the same or strongly overlapping niches. This is not because competition has been turned off or reduced – species still use the same amount of resource in any one timestep and the total resource available is constant – but the organisation of that competition is different. We can make some intuitive sense of this observation. In an ecosystem of species where competition is latent the organisation of the species with respect to one another is based on out-of-date information – based on the locations that species occupied some evolutionary timesteps in the past. Thus, such an ecosystem is less able to organise itself to utilise resources efficiently than one where the information about the location of other species is up-to-date. In the case of a uniform distribution of resources, the non-latent ecosystem simply spreads out evenly over niche space. In the case of a non-uniform distribution of resources we observe that the distribution of species is correspondingly non-uniform tending to approximately equalise the landscape. In contrast, in the latent dynamics (given either uniform or non-uniform resources), species positions are clumped together leaving many resources under-utilised at any particular point in time. The non-latent dynamics are thereby relatively efficient in their collective use of resources compared to the latent dynamics. Future work could plausibly quantify the efficiency of the resultant resource utilisation and the relationship of these qualitative regimes to the amount of latency.

It also makes intuitive sense that the two modes of diverse-efficient organisation and clumped-inefficient organisation correspond to the two dynamical modes of stasis and Red Queen dynamics, respectively. In order for a species to stop evolving, selective gradients in all directions need to be exhausted. That can be achieved when species are utilising all resources in a diverse manner, but when species are clumped together that naturally leaves some resources under-utilised with non-zero fitness gradients that promote further evolutionary change. This suggests a systematic relationship between ecological organisation and diversity and evolutionary stasis that deserves further attention.

In artificial coevolution, these two modes of behaviour correspond loosely to the two types of coevolutionary set-up – cooperative and competitive. In the diverse organisation, the ecosystem as a whole collectively solves the problem of utilising all available resources. This effectively decomposes the overall problem, dynamically dividing it up into semi-independent sub-problems and avoiding a scenario where multiple species attempt to solve the same part of the problem. But, at the same time, this scenario fails to produce an arms race where continued evolutionary change is observed. Conversely, in the clumped organisation, the species do engage in arms races – each species continually pushing other species to evolve to new areas of the trait space. This means that all species tend to cover all areas of the space, including the highest peaks in the underlying fitness landscape. But, at the same time, in this dynamic the ecosystem as a whole fails to utilise all resources collectively. Accordingly, we can view the non-latent dynamics as producing a cooperative coevolution scenario and the latent dynamics as producing a competitive coevolution scenario. The former effectively solves the problem-decomposition issue of cooperative coevolution (although the nature of the decomposition problem here is not difficult) whereas the latter effectively solves the arms-race conditions required for competitive coevolution (although continued improvement is a different matter). However, our observations also suggest that finding a balance of the two is non-trivial and perhaps even intrinsically opposed. In the current models the trait space of a species and its fitness dependencies on other species have a direct relationship (i.e., based on distance). For engineering purposes, where the task being performed (collectively or individually) is complex, the relationship between the trait space of an individual and its frequency dependent fitness effects on others will be less straightforward. Nonetheless, further investigation could address the change-over from the diverse-stable mode to the converged-dynamic mode as a function of latency and whether or not the trade-off of competition and cooperation can be usefully controlled.

In summary, our simple evolutionary model allows us to explore the dynamics of coevolution under various conditions. The model is simple and easy to understand and the behaviour of the species can be observed in real time. Our investigations using these models illustrate some important factors which influence the balance of evolutionary stasis and Red Queen dynamics. We find that both outcomes are possible without changing the nature of the underlying game, or the level at which selection is applied, but merely by altering the coupling between ecological and evolutionary timescales. Specifically, if ecological change is nearly instantaneous compared to evolutionary change, stasis results; but conversely, if evolutionary timescales are closer to ecological timescales, such that resource depletion is not instantaneous on evolutionary timescales, then Red Queen dynamics result. We also observe that in the stasis mode, the overall utilisation of resources by the ecosystem is relatively efficient whereas in the Red-Queen mode, the organisation of the ecosystem is inefficient as species tend to clump together competing in overlapping niches. These models thereby suggest a link between the issues of change and stasis discussed by Stenseth and Maynard Smith and matters of ecological diversity and organisation. In the same way, these models also suggest some basic conditions that influence the organisation of inter-species interactions and the balance of individual and collective adaptation in ecosystems.

References

[1] Adami, C.: Learning and complexity in genetic auto-adaptive systems. Physica D 80, 154–170 (1995)

[2] Carrol, L.: The Complete Works. CRW Publishing Limited, London (2005)

[3] Cliff, D., Miller, G.F.: Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. In: Morán, F., Moreno, A., Merelo, J.J., Chacón, P. (eds.) Third European Conference on Artificial Life, pp. 200–218. Springer, Berlin (1995)

[4] Cliff, D., Miller, G.F.: Co-evolution of pursuit and evasion II: Simulation methods and results. In: Maes, P., Mataric, M.J., Meyer, J.A., Pollack, J., Wilson, S.W. (eds.) From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, pp. 506–515. The MIT Press, Cambridge (1996)

[5] Dawkins, R., Krebs, J.R.: Arms races between and within species. Proc. R. Soc. Lond. B 205, 489–511 (1979)

[6] Dieckmann, U., Doebeli, M.: On the origin of species by sympatric speciation. Nature 400, 354–357 (1999)

[7] Ebner, M.: A three-dimensional environment for self-reproducing programs. In: Kelemen, J., Sosík, P. (eds.) ECAL 2001. LNCS (LNAI), vol. 2159, pp. 306–315. Springer, Heidelberg (2001)

[8] Ebner, M.: Coevolution and the red queen effect shape virtual plants. Genetic Programming and Evolvable Machines 7(1), 103–123 (2006)

[9] Ebner, M., Watson, R.A., Alexander, J.: Co-evolutionary dynamics on a deformable landscape. In: Zalzala, A., Fonseca, C., Kim, J.H., Smith, A., Yao, X. (eds.) Proceedings of the 2000 Congress on Evolutionary Computation, vol. 2, pp. 1284–1291. IEEE Press, San Diego (2000)

[10] Ebner, M., Watson, R.A., Alexander, J.: Coevolutionary dynamics of interacting species. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 1–10. Springer, Heidelberg (2010)

[11] Faro, J., Velasco, S.: An approximation for prey-predator models with time delay. Physica D 110, 313–322 (1997)

[12] Floreano, D., Nolfi, S.: God save the red queen! Competition in co-evolutionary robotics. In: Koza, J.R., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M., Iba, H., Riolo, R.L. (eds.) Genetic Programming 1997: Proceedings of the Second International Conference on Genetic Programming, July 13-16, pp. 398–406. Morgan Kaufmann Publishers, San Francisco (1997)

[13] Floreano, D., Nolfi, S., Mondada, F.: Competitive co-evolutionary robotics: From theory to practice. In: Pfeifer, R., Blumberg, B., Meyer, J.A., Wilson, S.W. (eds.) From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior, pp. 515–524. The MIT Press, Cambridge (1998)

[14] Gillespie, J.H.: Molecular evolution over the mutational landscape. Evolution 38, 1116–1129 (1984)

[15] Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, Reading (1989)

[16] Higgs, P.G., Derrida, B.: Genetic distance and species formation in evolving populations. Journal of Molecular Evolution 35, 454–465 (1992)

[17] Hillis, W.D.: Co-evolving parasites improve simulated evolution as an optimization procedure. In: Langton, C.G., Taylor, C., Farmer, J.D., Rasmussen, S. (eds.) Artificial Life II, SFI Studies in the Sciences of Complexity, pp. 313–324. Addison-Wesley (1991)

[18] Hutchinson, G.E.: The niche: an abstractly inhabited hypervolume. In: Hutchinson, G.E. (ed.) The Ecological Theater and the Evolutionary Play, Yale University Press, New Haven (1965)

[19] Juille, H., Pollack, J.B.: Coevolving the ideal trainer: Application to the discovery of cellular automata rules. In: Koza, J.R. (ed.) Proceedings of the Third Annual Conference on Genetic Programming, University of Wisconsin, Madison, WI, pp. 519–527. Morgan Kaufmann, San Francisco (1999)

[20] Kauffman, S.A.: The Origins of Order. Self-Organization and Selection in Evolution. Oxford University Press, Oxford (1993)

[21] Luke, S., Wiegand, R.P.: Guaranteeing coevolutionary objective measures. In: De Jong, K.A., Poli, R., Rowe, J.E. (eds.) Foundations of Genetic Algorithms VII, pp. 237–251. Morgan Kaufman, San Francisco (2002)

[22] MacArthur, R.H.: The theory of the niche. In: Lewontin, R.C. (ed.) Population Biology and Evolution, Syracuse University Press, Syracuse (1968)

[23] MacArthur, R.H., Levins, R.: Competition, habitat selection and character displacement in a patchy environment. Proc. Nat. Acad. Sci. USA 51, 1207–1210 (1964)

[24] May, R.M., MacArthur, R.H.: Niche overlap as a function of environmental variability. Proc. Nat. Acad. Sci. USA 69, 1109–1113 (1972)

[25] Metz, J.A.J., Geritz, S.A.H., Meszena, G., Jacobs, F.J.A., van Heerwaarden, J.S.: Adaptive dynamics: a geometrical study of the consequences of nearly faithful reproduction. In: Strien, S.J.V., Lunel, S.M.V. (eds.) Stochastic and Spatial Structures of Dynamical Systems, North Holland, Amsterdam, The Netherlands, pp. 183–231 (1996)

[26] Miller, G.F., Cliff, D.: Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In: Cliff, D., Husbands, P., Meyer, J., Wilson, S.W. (eds.) From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior, pp. 411–420. The MIT Press, Cambridge (1994)

[27] Pavlicev, M., Cheverud, J.M., Wagner, G.P.: Evolution of adaptive phenotypic variation patterns by direct selection for evolvability. Proceedings of the Royal Society B: Biological Sciences 278, 1903–1912 (2011)

[28] Pelletier, F., Garant, D., Hendry, A.P.: Eco-evolutionary dynamics. Phil. Trans. R. Soc. B 364, 1483–1489 (2009)

[29] Post, D.M., Palkovacs, E.P.: Eco-evolutionary feedbacks in community and ecosystem ecology: interactions between the ecological theatre and the evolutionary play. Phil. Trans. R. Soc. B 364, 1629–1640 (2009)

[30] Potter, M.A., Couldrey, C.: A cooperative coevolutionary approach to function optimization. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 374–383. Springer, Heidelberg (2010)

[31] Ray, T.S.: Is it alive or is it GA? In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, University of California, San Diego, pp. 527–534. Morgan Kaufmann Publishers, San Mateo (1991)

[32] Ray, T.S.: Evolution, complexity, entropy and artificial reality. Physica D 75, 239–263 (1994)

[33] Reynolds, C.W.: Competition, coevolution and the game of tag. In: Brooks, R.A., Maes, P. (eds.) Artificial Life IV, July 6-8, pp. 59–69. The MIT Press, Cambridge (1994)

[34] Ridley, M.: The Red Queen: Sex and the Evolution of Human Nature. Penguin Books, New York (1994)

[35] Riolo, R.L., Cohen, M.D., Axelrod, R.: Evolution of cooperation without reciprocity. Nature 414, 441–443 (2001)

[36] Rosenzweig, M.L., Brown, J.S., Vincent, T.L.: Red queens and ESS: The coevolution of evolutionary rates. Evolutionary Ecology 1, 59–94 (1987)

[37] Roughgarden, J.: Coevolution in ecological systems: results from "loop analysis" for purely density-dependent evolution. In: Christiansen, F.B., Fenchel, T.M. (eds.) Measuring Selection in Natural Populations, pp. 499–517. Springer, Berlin (1977)

[38] Maynard Smith, J.: A comment on the red queen. Amer. Natur. 110, 325–330 (1976)

[39] Stenseth, N.C., Maynard Smith, J.: Coevolution in ecosystems: Red queen evolution or stasis? Evolution 38(4), 870–880 (1984)

[40] Tregenza, T., Butlin, R.K.: Speciation without isolation. Nature 400, 311–312 (1999)

[41] Van Valen, L.: A new evolutionary law. Evolutionary Theory 1, 1–30 (1973)

[42] Valentine, J.W.: Conceptual models of ecosystem evolution. In: Schopf, T.J.M. (ed.) Models in Paleobiology, pp. 192–215. Freeman Cooper, San Francisco (1972)

[43] Vygotsky, L.S.: Mind and Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge (1978)

[44] Watson, R.A., Pollack, J.B.: Coevolutionary dynamics in a minimal substrate. In: Spector, L. (ed.) Proceedings of the Genetic and Evolutionary Computation Conference, pp. 702–709 (2001)

[45] Wiegand, R.P.: An Analysis of Cooperative Coevolutionary Algorithms. George Mason University Fairfax, VA (2004)

[46] Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones, D.F. (ed.) Proceedings of the Sixth International Congress on Genetics, Ithaca, NY, pp. 356–366 (1932)

Chapter 13
Correlation Analysis of Coupled Fitness Landscapes

Wim Hordijk

Abstract. In this chapter we present an overview of a statistical analysis to measure and express the correlation structure of fitness landscapes. This correlation analysis is then applied to both static and coupled fitness landscapes as generated by the NK-model and the NKC-model, respectively. An overview of the main results is provided, which shows that this correlation analysis can indeed be applied in a meaningful way to coupled fitness landscapes. This can provide a direct and useful link to the actual search performance of evolutionary algorithms that use a coevolutionary approach.

13.1 Introduction

Fitness landscapes are a useful metaphor in the context of evolutionary computation to envision a population of candidate solutions as moving around on a more or less mountainous landscape, in search of the highest peaks. This concept originated in population genetics, but has turned into a full theory in the area of evolutionary computation, trying to relate the structure of fitness landscapes to the performance of (evolutionary) search algorithms.

To better understand this relation, it is important to have a quantitative way of measuring and expressing the structure of fitness landscapes. Traditionally, this is done by using a measure of the landscape's correlation length, i.e., the largest distance at which there is still some correlation between the fitness values of two (arbitrary) points in the landscape. However, this is a rather crude summary of a landscape's overall structure. In this chapter, we review a more detailed measure of fitness landscapes that identifies and estimates a full statistical model to express a landscape's correlation structure.

Wim Hordijk
SmartAnalytiX.com, Lausanne, Switzerland
e-mail: wim@WorldWideWanderings.net

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_13, © Springer-Verlag Berlin Heidelberg 2014

In the context of evolutionary computation, where the goal is to optimize a given objective function, the associated fitness landscape is usually fixed. However, in nature, different species do not evolve in isolation but coevolve together, thus creating changing environments for each other. This idea of coevolution has also been adopted in evolutionary computation, where for example candidate solutions coevolve with test cases. In this situation, the resulting fitness landscape is not static, but the landscapes of two or more coevolving populations are coupled. A move of one population on its own landscape changes the shape of the landscape of the other population. In this chapter, we show how the correlation analysis mentioned above can be extended and applied to such coupled fitness landscapes as well.

The remainder of the chapter is structured as follows. The next section provides a brief review of the concept and definition of fitness landscapes, and of a useful model for studying the structure of fitness landscapes known as the NK-model, and also the NKC-model, its extension to coupled landscapes. Section 13.3 then presents an overview of a statistical analysis to measure and express the correlation structure of fitness landscapes. It is also explained how this correlation analysis can be applied to coupled fitness landscapes. The results of applying this correlation analysis to both static and coupled fitness landscapes generated by the NK(C)-model are presented in Section 13.4. Finally, Section 13.5 summarizes the main conclusions.

13.2 Fitness Landscapes

13.2.1 The Concept of Fitness Landscapes

The concept of a *fitness landscape* was originally introduced in population genetics as a metaphor to envision the flow of populations on a landscape with hills, valleys, and mountain peaks [31]. The height of each point in the landscape is associated with the fitness of an individual (or sub-population) occupying that point in genotype space. An evolving population can now be seen as moving through this space under the forces of genetic mutation and natural selection, trying to "find" the highest peaks in the landscape.

This landscape metaphor was later adopted in evolutionary computation [1, 3, 24] to envision (combinatorial) optimization as a search through a mountainous landscape, also looking for the highest peaks. In this case, the height of a point in the landscape is the value of some objective function as evaluated on the candidate solution represented by that point, where the objective function is defined by a given optimization problem. However, the fitness landscape concept has gone far beyond being a mere metaphor, and has developed into a full theory that tries to relate the structure of a fitness landscape to the performance of search and optimization algorithms (see e.g. [9, 12, 13, 15, 19, 21, 25, 26, 27]).

13.2.2 The Definition of Fitness Landscapes

Generally, a fitness landscape is defined by three components:

(1) A *representation space* V, where each point $v \in V$ represents, in some suitable encoding, a phenotype or a candidate solution to some optimization problem.
(2) A *neighborhood relation* $\mathscr{N} : V \rightarrow 2^V$ which induces a metric, or notion of "closeness", on the representation space.
(3) A *fitness function* $f : V \rightarrow \mathbb{R}$ which associates a fitness value f with each point v in the representation space.

The representation space V can, for example, consist of all possible DNA sequences, i.e., strings over the alphabet $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, where each DNA string (or genotype) represents a possible phenotype. Or, the representation space could consist of the possible bit strings of a given length n, i.e., $V = \{0,1\}^n$. These bit strings could then represent candidate solutions to some optimization problem. For example, given a graph G with n nodes for which a subgraph $G' \subset G$ with some optimal property needs to be found, each possible bit string of length n naturally represents a possible subgraph G' (i.e., subset of nodes).

The neighborhood relation $\mathscr{N}$ indicates which other elements of V are direct "neighbors" of a given point $v \in V$. Thus, $\mathscr{N}$ maps a point in the landscape to a subset of all possible points, i.e., those that are its neighbors. The notation 2^V indicates the set of all possible subsets of the elements of a set V. In the context of evolutionary algorithms, the neighborhood relation is usually defined by the "move operator" of interest, i.e, the operator that is used to search through the representation space V. In other words, the neighborhood relation $\mathscr{N}(v)$ indicates which points in V can be reached in one move (of the search algorithm) from a given point v. In the case of bit strings as representation and point mutation as the search operator, $\mathscr{N}(v)$ consists of all bit strings that differ in exactly one bit from v, for example $\mathscr{N}(000) = \{001, 010, 100\}$. This automatically induces the Hamming distance as a metric on V.

Obviously, one can define different fitness landscapes on the same representation space V by using different search operators, as expressed in the "one operator, one landscape" claim [13, 14]. Also note that there can be probabilities associated with the neighborhood relation. In the point mutation example above, each one of the three neighbors of a given point v has an equal probability of being produced by the move operator. However, if we consider a mutation operator where each bit has a probability p of being mutated, then the neighborhood of a given point v is the entire space V (if $p > 0$, there is always a small but non-zero probability that each bit is mutated). But in this case the probabilities of different neighbors being produced are not equal. In particular, the probability that a given neighbor w at Hamming distance h is produced, is equal to $p^h(1-p)^{n-h}$ for bit strings of length n. In one proposed variant of the fitness landscape definition above, these probabilities are included as well [13].

Finally, the fitness function f can be anything from a simple mathematical function that calculates a fitness value $f(v)$ directly from a given representation v, to some complicated function that first transforms a given representation v into an associated candidate solution, and then computes a fitness value based on how well this candidate solution solves the given optimization problem (which itself could be a complicated computation or even simulation). Unless stated otherwise, here we assume optimization to mean maximization, i.e., a higher fitness value is better. In case of a minimization problem, we can simply take $-f(v)$ (or perhaps $1/f(v)$) and still consider it as a maximization problem.

13.2.3 A Model of Fitness Landscapes

In this chapter, we use a particular model of fitness landscapes known as the *NK-model* [18, 19, 21], which was introduced to have a tunable and problem-independent model for constructing fitness landscapes. The main parameters of the model are N, the number of "genes" in the genotype, i.e. the length of the strings that constitute the points in the representation space V, and K, the number of other genes that epistatically influence a given gene, i.e., the fitness contribution of each gene is determined by the gene itself plus K other genes. We consider an alphabet of size two, i.e., the representation space is $V = \{0,1\}^N$, or all bit strings of length N.

The fitness of a bit string $v \in V$ is defined as follows. Assume every bit v_i $(i = 1,\ldots,N)$ in a bit string v is assigned a fitness contribution f_i. However, this fitness contribution f_i does not only depend on the value (0 or 1) of the bit itself, but also on the values of K other bits v_j in the bit string $(0 \leq K \leq N-1)$. These dependencies are called *epistatic interactions*. So, the fitness contribution of one bit depends on the value of $K+1$ bits in total: the bit itself and K others. This gives rise to a total of 2^{K+1} possibilities, called neighborhood configurations. Each of these neighborhood configurations is assigned a random value drawn from a uniform probability distribution between 0.0 and 1.0. The fitness contribution f_i of bit v_i can thus be specified by a list of 2^{K+1} random values between 0.0 and 1.0. This assignment of fitness contributions is repeated for each bit v_i, $i = 1,\ldots,N$. So, there are N lists (one for each bit), each with 2^{K+1} entries.

Having thus assigned the possible fitness contributions for every bit in the string, the fitness of the entire bit string is now defined as the average of the fitness contributions f_i of all the bits, given its particular configuration of bits:

$$f(v) = \frac{1}{N}\sum_{i=1}^{N} f_i.$$

Table 13.1 shows an example, taken from [19], for $N = 3$ and $K = 2$ (i.e., the fitness contribution of each bit depends on all other bits).

One final aspect of the NK-model determines how the K epistatic interactions for each bit are chosen. Generally, this is done in one of two ways. The first way is by choosing them at random (without repetition or reciprocity) from among the other $N-1$ bits. This is called *random interactions*. The second way is by choosing them as the K neighboring bits. Thus, the $K/2$ bits on each side of a bit will influence the

Table 13.1 An example of an instance of the NK-model for $N = 3$ and $K = 2$. The fitness contributions f_i for the three bits for each of the $2^{K+1} = 8$ neighborhood configurations are assigned at random. The fitness of the entire string is the average of the fitness contributions of the three bits.

bit value			fitness contribution			total fitness
v_1	v_2	v_3	f_1	f_2	f_3	$f(v)$
0	0	0	0.6	0.3	0.5	0.47
0	0	1	0.1	0.5	0.9	0.50
0	1	0	0.4	0.8	0.1	0.43
0	1	1	0.3	0.5	0.8	0.53
1	0	0	0.9	0.9	0.7	0.83
1	0	1	0.7	0.2	0.3	0.40
1	1	0	0.6	0.7	0.6	0.63
1	1	1	0.7	0.9	0.5	0.70

fitness of this bit. This is called *nearest neighbor interactions*. To make this possible, periodic boundary conditions are taken into account, i.e., the bit string is considered circular, so the first and the last bit are each others neighbors. Note that for $K = 0$ and $K = N - 1$, there is no difference between the two types of interactions.

Landscapes resulting from the NK-model were originally constructed using point mutation as defining the neighborhood relation $\mathcal{N}$. This results in a boolean hypercube of dimension N. The resulting fitness landscape for the example in Table 13.1 is shown in Figure 13.1. Later in this chapter, an example of an NK-landscape using a different neighborhood relation is shown.

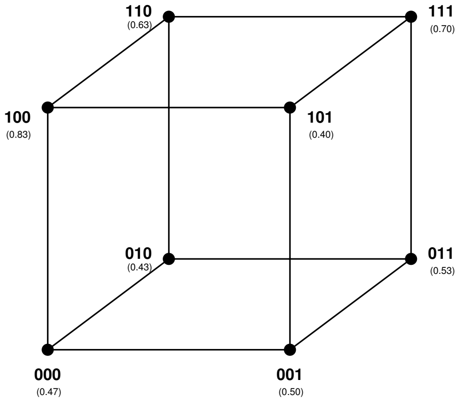


Fig. 13.1 The fitness landscape resulting from the instance of the NK-model in Table 13.1 and using point mutation as the neighborhood relation. Fitness values are given between parentheses.

By increasing the value of K from 0 to $N-1$, the fitness landscapes resulting from the NK-model can be tuned from "smooth" to "rugged". When K is small, neighboring bit strings (i.e., bit strings with a Hamming distance of one) will have a relatively small difference in fitness, as the bit that is different between the two strings influence the fitness contribution of only a small number of other bits in either string. So, this will have a small influence on the overall fitness of both strings, and thus their fitness values will be quite similar. As a consequence, the resulting fitness landscape will be relatively "smooth".

When K is large, however, the bit that is different between two neighboring strings will influence the fitness contribution of a large number of bits in either string. This will have a large influence on the overall fitness of both strings, and their fitness values will be quite different. In the extreme case of $K = N-1$, the resulting fitness landscape will be completely random, since changing the value of only one bit will change the fitness contribution of every bit in the string, and the overall fitness will be completely different. Thus, for increasing values of K, the resulting landscapes will become more and more "rugged". This "ruggedness" of fitness landscapes can be measured and expressed more formally by using a *correlation analysis*, as will be explained in Section 13.3 below.

13.2.4 Coupled Fitness Landscapes

In nature, species do not evolve on their own or in complete isolation. Organisms compete and cooperate in various ways, and the fitness of one individual strongly depends on how well it is able to avoid, catch, or cooperate with another. Hence, fitness landscapes are not static, but change continuously. An adaptation of one species will "deform" the fitness landscape of another one. In other words, the fitness landscapes of different species are "coupled".

One extension of the NK-model, called the NKC-model, tries to capture this coupling [19, 20]. Assume two coevolving "species", represented by bit strings v and w, respectively. Now, next to having K *internal* epistatic interactions, each bit v_i in bit string v also has C *external* epistatic interactions with bit string w. In other words, the fitness contribution of a bit v_i in string v depends on K other bits v_j in string v *and also* on C bits w_j in string w; and *vice versa* for the fitness contributions of the bits w_i in string w. These C external epistatic interactions are chosen at random for each bit independently. This increases the possible neighborhood configurations for each bit to 2^{K+C+1}, each of which is again assigned a random value between 0.0 and 1.0.

This idea of coupled fitness landscapes is interesting in the context of evolutionary computing as well, as the idea of coevolution has been incorporated in evolutionary algorithms in various ways (see e.g. [7, 17, 29] for some early examples). In fact, in some cases this has actually given rise to improvements in performance of the search algorithm. It is therefore useful to have a tunable model of coupled landscapes, as provided by the NKC-model, to study fitness landscapes in a coevolutionary setting as well. To do so, we first introduce a statistical analysis that can be used to measure, express, and compare the structure of fitness landscapes.

13.3 Correlation Structure

As mentioned earlier, the “ruggedness” of NK-landscapes can be tuned by varying the value of K relative to N. This notion of ruggedness can be made more formal, for example by using a statistical analysis to measure and express the *correlation structure* of fitness landscapes. Informally, the correlation structure of a fitness landscape gives an indication of the average fitness differences between neighboring points in the landscape. More formally, it can be expressed as a function of the autocorrelations and partial autocorrelations of a time series of fitness values generated by a random walk through the landscape.

13.3.1 Time Series Analysis

Different measures for the correlation structure of a fitness landscape have been proposed and used in the literature (see e.g. [22, 23, 28]). However, these measures all come down to one single value, such as the correlation length or a related number. In [9] we introduced a more refined and expressive method based on the idea of [28] and a statistical time series analysis known as the *Box-Jenkins approach*.

The Box-Jenkins approach [2] is a useful statistical method of model building which is based on the analysis of a time series $y_1, y_2, \ldots, y_T$ of observed values, generated by a stochastic process. The purpose of the Box-Jenkins approach is to find an ARMA (autoregressive moving-average) model that adequately represents this data generating process. An ARMA(p,q) model is the sum of an autoregressive AR(p) model of order p and a moving-average MA(q) model of order q:

$$y_t = c + \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \cdots + \beta_q \varepsilon_{t-q}, \tag{13.1}$$

where the orders p and q and the coefficients α_i and β_i are the model parameters (to be estimated from the data), and the stochastic variable ε_t is white noise, i.e., $E[\varepsilon_t] = 0$ for all t and $Cov(\varepsilon_{t'}, \varepsilon_t) = 0$ for $t' \neq t$. In words, each value y_t in an ARMA(p,q) process depends on p past values and a weighted sum of q members of a white noise series.

In economics the Box-Jenkins approach is used frequently when a model is needed to make forecasts about future values of some (partly) stochastic variable, for example the price of some commodity, or the index of industrial production [6, 16]. The approach consists of three stages:

(1) *Identification*, in which a choice is made for an appropriate ARMA model (i.e., appropriate values for p and q).
(2) *Estimation*, in which the parameters of the chosen model are estimated (i.e., the coefficients α_i and β_i).
(3) *Diagnostic checking*, in which the estimated model is tested for adequateness.

13.3.1.1 Identification

At the identification stage, an appropriate model within the class of ARMA models is specified on the basis of the *correlogram* and the *partial correlogram*. The (partial) correlogram of a time series y_t is a plot of the estimated (partial) autocorrelations of the given time series.

The *autocorrelation* ρ_i of a time series y_t relates the value of two points in the time series which are i steps (or *time lags*) apart. Estimates r_i of these autocorrelations ρ_i are obtained from the time series as follows:

$$r_i = \frac{\sum_{t=1}^{T-i}(y_t - \bar{y})(y_{t+i} - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2}, \tag{13.2}$$

where $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$. For each i, $-1 \leq r_i \leq 1$. If $|r_i|$ is close to one, there is a large amount of correlation between two values i time lags apart; if it is close to zero, then there is hardly any correlation.

The *partial autocorrelation* can be interpreted as the estimated correlation between y_t and y_{t+i} after the effects of all intermediate y_t's on this correlation are taken out. Estimates of the partial autocorrelation of order k, a_{kk}, can be found by solving the set of simultaneous linear equations

$$r_i = \sum_{j=1}^{k} a_{kj} r_{i-j}, \quad i = 1, 2, \ldots, k \tag{13.3}$$

The choice of an appropriate ARMA model is now made on the following basis:

- If the correlogram "tapers off" (gradually decreases) to zero and the partial correlogram suddenly "cuts off" (i.e., becomes equal to zero) after some time lag, say p, then an appropriate model is AR(p).
- If the correlogram cuts off after some time lag, say q, and the partial correlogram tapers off to zero, then an appropriate model is MA(q).
- If neither diagram cuts off after some time lag, but both taper off, then an appropriate model is ARMA(p,q). The values of p and q have to be inferred from the particular pattern of the two diagrams.

The cut-off point for the estimated (partial) autocorrelations is usually taken as $\pm 2/\sqrt{T}$, where $1/\sqrt{T}$ is the approximate standard error of these estimates. Figure 13.2 below shows an example of correlograms tapering off to zero, and Figure 13.3 shows an example of a partial correlogram cutting off after just one time lag.

13.3.1.2 Estimation

Once the appropriate model is chosen in the identification stage, the coefficients α_i and β_i in Equation (13.1) are estimated. This is achieved by using the estimates of the (partial) autocorrelations of the given time series and a non-linear least squares or a maximum likelihood estimation method. As a measure of *significance* of the estimated parameters, a t-statistic is used, which is calculated as the estimated value of the parameter divided by its estimated standard error. An estimated parameter is considered significant (at a 95% confidence level) if its t-statistic (in absolute value) is larger than 2.

As a measure of "goodness of fit" of the estimated model, the *coefficient of determination* R^2 is used:

$$R^2 = \frac{\left[\sum_{t=1}^{T}(y_t - \bar{y})(\hat{y}_t - \bar{y})\right]^2}{\sum_{t=1}^{T}(y_t - \bar{y})^2 \sum_{t=1}^{T}(\hat{y}_t - \bar{y})^2}, \tag{13.4}$$

where $\bar{y} = \sum_{t=1}^{T} y_t$ and $\hat{y}_t$ is the data as estimated by the model. This value is a measure of the proportion of the total variance in the data that is accounted for by the explanatory variables in the estimated model. A value of R^2 close to one indicates a good fit, while a value close to zero means that the stochastic component of the model plays a dominant role.

13.3.1.3 Diagnostic Checking

Before the estimated model is used, it is important to check that it is indeed an adequate one. The usual test for this is to fit the model on the data and calculate the autocorrelations of the residuals (the difference between the observed values (y_t) and those predicted by the model ($\hat{y}_t$)). These residuals should be white noise (i.e., uncorrelated, zero-mean variables), meaning that all residual autocorrelations should not be significantly different from zero. To check this, the residual autocorrelations are compared with the same $\pm 2/\sqrt{T}$ bound (as with the autocorrelations and partial autocorrelations).

Another test is to fit a slightly higher-order model and then check that the extra parameters are not significantly different from zero. Thus, if for example an AR(p) model is estimated, an AR($p+1$) model can then also be estimated and the extra parameter (α_{p+1}) should not be significant (i.e., its t-statistic should be smaller than two).

In summary, the Box-Jenkins approach is a statistical time series analysis with the purpose of finding an appropriate ARMA model that can be used to explain the process that generated, or to predict future values of, the values of a given time series y_t. The approach is based on the estimated (partial) autocorrelations of the time series, and the adequateness and goodness of fit of the estimated model can be assessed with several statistics (t-statistics, standard error values, and the R^2 value).

13.3.2 Random Walks

In [28], the idea of using a *random walk* to generate a time series of fitness values to measure the correlation structure of a fitness landscape was introduced. This idea can be extended to coupled fitness landscapes as well (as is described later in this section) by introducing a parameter that represents the relative rate of evolution between two (or more) coevolving species. The basic idea of a random walk (on a static landscape) works as follows. Start from a random point v_1 in the fitness landscape, and evaluate and record its fitness value $y_1 = f(v_1)$. Next, choose a point v_2 at random from v_1's neighborhood $\mathscr{N}(v_1)$ and move there. Evaluate and record its fitness value $y_2 = f(v_2)$. Now choose a random point from v_2's neighborhood, and continue these random moves and fitness evaluations for a given number T of time steps, thus generating a random walk through the landscape and a corresponding time series of fitness values y_t, $t = 1 \dots T$. More concretely, in terms of NK-landscapes based on point-mutation as described earlier, start from a random bit string, and at each time step flip a random bit, evaluating fitness values along the way.

This idea can of course be extended to any kind of move operator. For example, in [9], the "random-mate random-child crossover" (RMRC) operator was introduced to investigate fitness landscapes based on the crossover operator used in for example genetic algorithms [5, 8, 24]. Assuming again bit strings, as in the NK-model, the idea is to pick, at each step, a random mate (i.e., a random bit string) to perform one-point crossover with, and choose one of the two resulting children as the next point for which to calculate the fitness value. This process is then repeated for a given number of steps, starting from a random initial bit string, again generating a time series of fitness values, but from a *different* fitness landscape, i.e., one that is defined by a neighborhood relation based on the crossover operator.

Furthermore, especially with other kinds of representations, still other neighborhood relations based on other move operators can be envisioned, such as PMX crossover for permutations, or subtree prune-and-regraft (SPR) moves for tree representations. Of course the random walk method can also still be implemented when probabilities are associated with the neighborhood relation. In each case, the idea of performing a random walk on the resulting landscape is the same: start from a random point, calculate its fitness, apply the chosen move operator to generate a new point, calculate its fitness, and so on.

Finally, this random walk method to generate a time series of fitness values can also be extended to coupled landscapes by introducing one additional parameter, m, which can be interpreted as a *relative rate of coevolution*. Suppose that we have two coevolving species, i.e., two coupled landscapes. A time series y_t of fitness values is then generated as follows. Perform a random walk on the first landscape just as in the single (static) landscape case described above, and record the fitness values $y_t = f(v_t)$ at each time step. However, starting from a random point in the second landscape, move to a random neighbor in this landscape every m steps during the random walk on the first landscape. In other words, perform m steps of the random walk on the first landscape, recording the fitness values, then make a random move in the second landscape, perform m steps on the first landscape again, continuing to record the fitness values, make a random move in the second landscape, and so on until a given number T of steps on the first landscape have been made (and thus T/m steps in the second landscape). This way, random walks can be performed on coupled landscapes, using the relative rate of coevolution m. If $m = 0$, the second species does not evolve at all, and the analysis degenerates to that of a static (uncoupled) fitness landscape (which is still useful for comparing the coevolutionary case with). However, when $m > 0$, each time a random move on the second landscape is made (every m time steps), it *changes* the shape of the first landscape on which the actual times series of fitness values is being generated. Note that due to the way NKC-landscapes are defined, statistically there is no difference when the roles of the first and second landscapes are swapped.

13.3.3 The Correlation Structure of Fitness Landscapes

Combining the two methods described above, i.e., performing a random walk on a given fitness landscape (coupled or not) to generate a time series of fitness values, and then applying a time series analysis such as the Box-Jenkins approach, the correlation structure of a landscape can be measured and expressed as a complete ARMA model. This provides a more expressive and powerful characterization of a landscape's correlation structure than a single number, such as the correlation length used in previous methods.

For example, it can be useful to have a way of predicting the fitness of a next (neighboring) point in the landscape given the fitness values of some previously visited points. As it turns out, always moving uphill as fast as possible (i.e., steepest ascent search) in a landscape does not always lead to the best result [4, 30]. Having an ARMA model that is able to predict the expected (average) increase in fitness can help in selecting a next point during the search that has a fitness increase closer to that as predicted by the model, thus taking advantage of the correlation structure of the landscape. Furthermore, the value of p in the estimated ARMA model indicates exactly how much "memory" is available in the landscape, i.e., how many steps in the past need to be remembered to predict the next step. Finally, additional statistics such as the R^2 of the estimated model, give a clear indication of how reliable the model's predictions actually are.

Having a well-defined and statistically robust method to analyze the correlation structure of fitness landscapes, and also a convenient and tunable model to generate fitness landscapes with varying levels of ruggedness, let us now turn to applying this correlation analysis to fitness landscapes of varying ruggedness.

13.4 Correlation Analysis of Fitness Landscapes

13.4.1 Static Fitness Landscapes

We start by investigating static, or non-coupled fitness landscapes. Using the NK-model as described in Section 13.2, landscapes of various ruggedness are constructed and the correlation analysis as explained in Section 13.3 is then applied to them. The representation space of all landscapes consists of bit strings of length $N = 100$, i.e., $V = \{0,1\}^{100}$. We consider two types of neighborhood relations $\mathscr{N}$ to construct the landscapes: (1) mutation and (2) crossover. With the mutation operator, a random bit in the string is flipped. For crossover, we use the RMRC crossover method as described in Section 13.3. Finally, the fitness function $f(v)$ is determined by an instance of the NK-model for $N = 100$ and a given value of K. Values used for K are 0, 2, 5, 25, 50, and 95, and both random and nearest neighbor interactions are considered. Random walks of 10,000 steps are performed with the given move operator (mutation or crossover) to generate a time series of fitness values for each landscape, on which the Box-Jenkins approach is then applied. The main results are reviewed here. More details can be found in [9].

13.4.1.1 Mutation Landscapes

An example of an NK-landscape with mutation as the neighborhood relation was given in Figure 13.1. Figure 13.2 shows the correlograms (i.e., the estimated autocorrelations) for the NK-landscapes with random epistatic interactions and the various values of K. The $\pm 2/\sqrt{T}$ significance bound (± 0.02 for $T = 10,000$) is also shown. As the figure shows, the correlograms taper off to zero, but more quickly so for larger values of K. The results for nearest neighbor interactions are very similar.

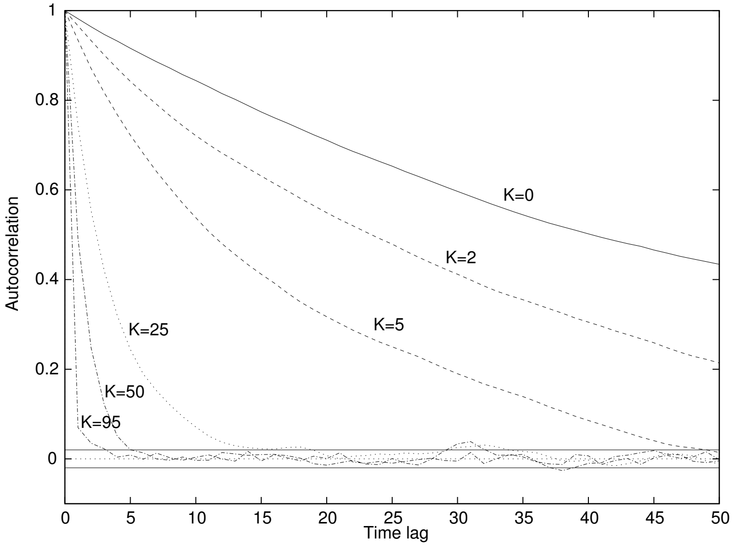


Fig. 13.2 The correlograms for NK-landscapes based on the mutation operator and random epistatic interactions, for various values of K

Table 13.2 The correlation lengths (largest time lag for which the estimated autocorrelation is still significantly different from zero) for the mutation operator for the different values of K

K	random	nearest
0	>50	>50
2	>50	>50
5	49	>50
25	18	14
50	5	7
95	3	1

Table 13.2 gives the correlation lengths for the various landscapes. The correlation length here is defined as the largest time lag for which the estimated autocorrelation is still significantly different from zero, i.e. is outside the $\pm 2/\sqrt{T}$ significance bound. This correlation length decreases with increasing values of K, indicating that the landscapes indeed become more rugged when the number of epistatic interactions becomes larger.

To identify the appropriate ARMA model, Figure 13.3 shows the partial correlogram (i.e., the estimated partial autocorrelations) for the landscape with random interactions and $K = 0$. Note that, by definition, the first partial autocorrelation a_{11} is always identical to the first autocorrelation r_1, which in this case is close to one (the landscapes for $K = 0$ are highly correlated). However, clearly the correlogram cuts off immediately after the first time lag (i.e., all other partial autocorrelations are within the significance bounds), suggesting an AR(1) model. The partial correlograms for the other values of K and for nearest neighbor interactions look similar.

Now that the appropriate model has been identified, its parameters can be estimated for each landscape. Since the identification stage suggests an AR(1) model of the form

$$y_t = c + \alpha_1 y_{t-1} + \varepsilon_t,$$

only the values of the parameters c (the constant) and α_1 need to be estimated. In fact, for an AR(1) process the value of the parameter α_1 is simply equal to the first autocorrelation, i.e., r_1. Table 13.3 shows the parameter estimation results (with t-statistics between parentheses) for the various landscapes, also including the R^2 (goodness-of-fit) values and the variance $Var(\varepsilon_t)$ of the stochastic term.

As the table shows, the estimated parameter values are all significant (t-statistic larger than two), and the α_1 decrease linearly with K, once again indicating the increase in landscape ruggedness with increasing values of K. Also, the R^2 value decreases, and the variance of the stochastic term increases with increasing K, showing that the model becomes less reliable with a larger stochastic component for larger K.

Finally, to check the estimation results, Figure 13.4 shows the residual autocorrelations for the landscape with random interactions and $K = 0$, together with the $\pm 2/\sqrt{T}$ significance level. Clearly, the residual autocorrelations are within the bounds (except for one very minor case, which is negligible). The results for other values of K and for nearest neighbor interactions look very similar. Also, estimating an additional parameter in the suggested model (for example α_2) yields an insignificant result (t-statistic smaller than two) in all cases. So, the identified and estimated models for the mutation-based fitness landscapes are adequate.

To summarize the results, it turns out that in all cases an AR(1) model is the most appropriate for expressing the correlation structure of NK-landscapes based on the mutation operator. This was already suggested in [28], and is thus verified formally by the above results. So, even though this result is not too surprising, it confirms two important aspects: (1) the NK-model indeed generates landscapes with a tunable degree of ruggedness, as intended, and (2) this ruggedness can be formally and meaningfully described with an AR(1) model, providing more information than

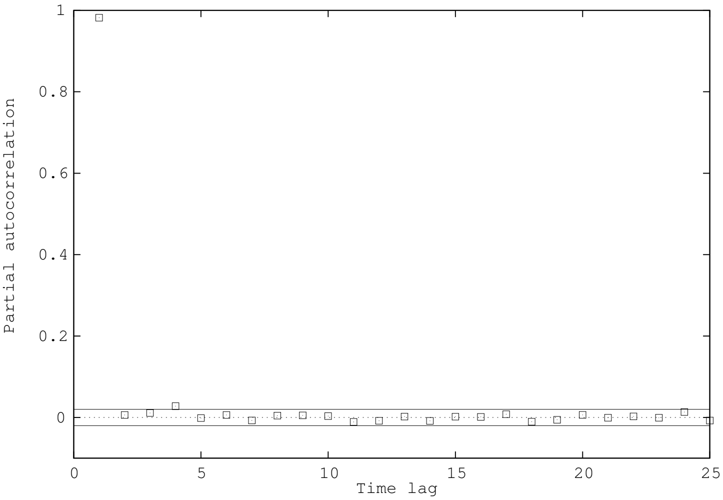


Fig. 13.3 The estimated partial autocorrelations for NK-landscapes based on the mutation operator and random epistatic interactions

Table 13.3 The estimated parameters for the AR(1) models for mutation for the different values of K

	random interactions				nearest neighbor interactions			
K	c	α_1	$Var(\varepsilon_t)$	R^2	c	α_1	$Var(\varepsilon_t)$	R^2
0	0.01 (9.59)	0.98 (518.20)	0.00002	0.96	0.01 (9.94)	0.98 (510.85)	0.00002	0.96
2	0.02 (13.08)	0.97 (375.48)	0.00005	0.93	0.02 (13.47)	0.96 (364.39)	0.00005	0.93
5	0.03 (18.46)	0.93 (261.59)	0.00010	0.87	0.03 (17.55)	0.94 (275.72)	0.00010	0.88
25	0.13 (38.35)	0.74 (110.91)	0.00038	0.55	0.13 (38.86)	0.74 (109.00)	0.00038	0.54
50	0.25 (58.29)	0.49 (56.41)	0.00064	0.24	0.25 (58.40)	0.49 (56.21)	0.00063	0.24
95	0.45 (66.09)	0.07 (6.76)	0.00082	0.01	0.46 (91.76)	0.08 (8.42)	0.00082	0.01

just a single number such as the correlation length (which, however, can still be derived directly from the model estimation results).

To give one example of how the AR(1) model expressing the correlation structure of NK-landscapes can be used, we compare an actual search on a given NK-landscape with a prediction from the AR(1) model. The search algorithm used is “random neighbor ascent hill climbing” (RNAHC) [19], where at each step a random fitter neighbor is chosen, where the neighborhood relation is based on point

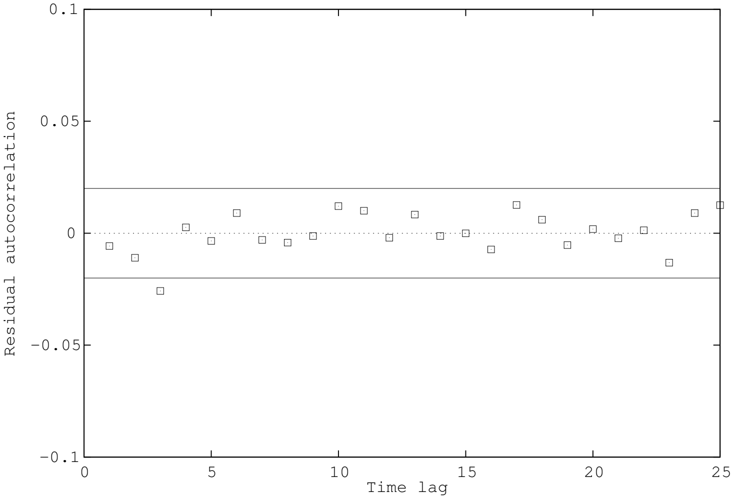


Fig. 13.4 The estimated residual autocorrelations for NK-landscapes based on the mutation operator and random epistatic interactions

mutation. Taking an instance of the NK-model for $N = 100$ and $K = 25$, the RNAHC algorithm is applied for 10,000 function evaluations, starting from a random bit string, and the best fitness value found so far is recorded at each step during the search. Next, the AR(1) model for this landscape

$$y_t = 0.13 + 0.74y_{t-1} + \varepsilon_t, \quad Var(\varepsilon_t) = 0.00038$$

is used to simulate this search process.

Figure 13.5 shows the results, averaged over 100 instances of the RNAHC search and model prediction. Even though the match is not exact, partly due to some small differences between the actual search and the model simulation, it does show the potential predictive value the model has (beyond simply calculating a correlation length). The predicted best fitness rises at a slightly slower rate than the actual best fitness, but eventually converges almost identically.

Having verified that the correlation analysis works correctly and provides meaningful results on NK-landscapes based on the mutation operator, we now turn to landscapes defined by the crossover operator.

13.4.1.2 Crossover Landscapes

An example of a fitness landscape where the neighborhood relation $\mathscr{N}$ is defined by random-mate random-child (RMRC) crossover, and with fitness values as defined by the instance of the NK-model in Table 13.1, is shown in Figure 13.6. Note that, unlike in the boolean hypercube defined by point mutation, not all points $\mathscr{N}(v)$ in the neighborhood of a given point v have the same probability of being reached. In a more elaborate model of fitness landscapes, the edges in the landscape are actually

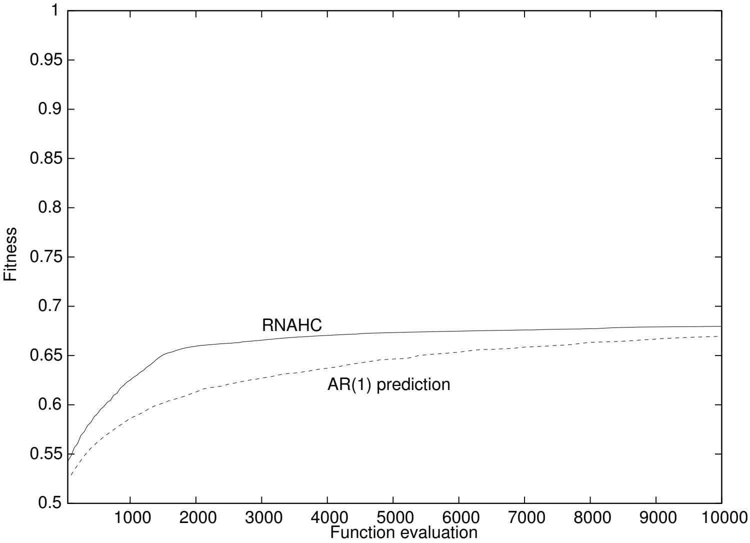


Fig. 13.5 Comparison of an actual search (RNAHC) with an AR(1) model prediction, averaged over 100 instances

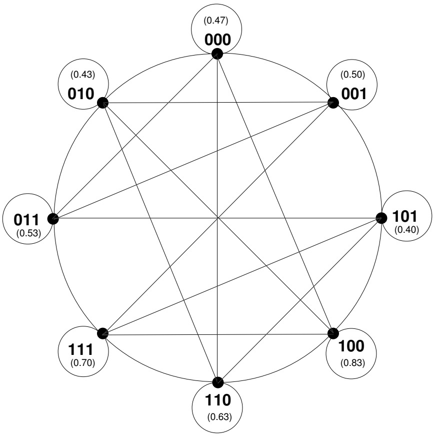


Fig. 13.6 The fitness landscape resulting from the instance of the NK-model in Table 13.1 and with RMRC crossover as the neighborhood relation. Fitness values are given between parentheses.

labeled with their corresponding transition probabilities [13]. The correlograms for the crossover landscapes for random epistatic interactions and the various values for K are shown in Figure 13.7. As with the mutation landscapes, the correlograms taper off to zero, although at a much faster rate. In fact, for the largest values of K, not even the first estimated autocorrelation r_1 is significantly different from zero (i.e., outside the $\pm 2/\sqrt{T}$ error bounds).

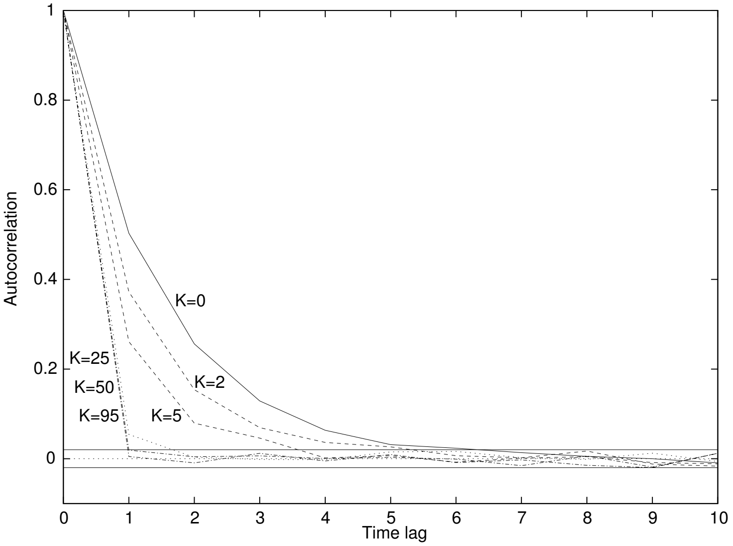


Fig. 13.7 The estimated autocorrelations for NK-landscapes based on the crossover operator and random epistatic interactions

For the crossover landscapes, there actually is a difference between random interactions and nearest neighbor interactions as used in the NK-model. Table 13.4, which shows the correlation lengths, makes this difference clear. Whereas the correlation length becomes zero for $K \geq 50$ with random interactions, there remains a small but positive correlation length even up to $K = 95$ for nearest neighbor interactions. This reflects the fact that crossover is less disruptive (and thus more correlated) for nearest neighbor interactions than it is for random interactions.

Table 13.4 The correlation lengths for the crossover operator for the different values of K

K	random	nearest
0	6	4
2	5	4
5	3	4
25	1	3
50	0	2
95	0	1

As with the mutation landscapes, the partial correlograms for crossover landscapes only have the first value well outside the error bounds (results not shown here), suggesting again that an AR(1) model is the most appropriate. An exception occurs for $K = 50$ and $K = 95$ with random interactions, for which even the first partial autocorrelation is not significantly different from zero, suggesting a purely white noise model instead. Table 13.5 shows the results of the parameter estimations.

As expected, the values for the α_1 coefficients and for the R^2 values decrease, and the values for the variance of the white noise term increases with increasing K. However, the estimation results also show a clear difference between the different types of epistatic interactions in that the α_1 coefficients and the R^2 values are significantly larger for nearest neighbor interactions than for random interactions for the same value of K (except for $K = 0$, for which there are no epistatic interactions at all). We can indeed conclude that nearest neighbor interactions result in a more correlated landscape than random interactions for the crossover operator, which is clearly and quantitatively expressed in the parameter estimates of the corresponding AR(1) models.

As Table 13.5 shows, all parameter estimates are significant (t-statistics larger than two). Furthermore, as with mutation, all residual autocorrelations for all crossover landscapes are within the error bounds, and over-estimating the models (i.e., adding more parameters) leads to non-significant results. Thus, the estimated models appear adequate for the crossover landscapes.

In conclusion, the correlation analysis can clearly be extended to landscapes defined by any search operator of interest (or even any combination of search operators, see e.g. [9]). Overall, NK-landscapes are relatively "well-behaved" though, in that they are *isotropic*, i.e., statistically they look the same everywhere (for any given value of N and K). However, the correlation analysis can also be applied to less isotropic landscapes, for example by restricting the random walks to subspaces in the landscape [10]. One further useful extension, as described in the next subsection, is to apply the analysis to coupled fitness landscapes.

Table 13.5 The estimated parameters for the AR(1) models for crossover for the different values of K

	random interactions				nearest neighbor interactions			
K	c	α_1	$Var(\varepsilon_t)$	R^2	c	α_1	$Var(\varepsilon_t)$	R^2
0	0.26 (57.52)	0.50 (58.21)	0.00034	0.25	0.23 (57.57)	0.50 (57.93)	0.00036	0.25
2	0.32 (67.56)	0.37 (40.06)	0.00061	0.14	0.27 (58.21)	0.49 (56.62)	0.00056	0.24
5	0.37 (76.42)	0.26 (27.04)	0.00073	0.07	0.27 (60.85)	0.46 (51.54)	0.00061	0.21
25	0.47 (94.56)	0.05 (5.41)	0.00084	0.01	0.35 (73.87)	0.29 (30.56)	0.00075	0.09
50	0.49 (97.92)	-	0.00082	-	0.43 (86.14)	0.15 (14.80)	0.00083	0.02
95	0.50 (99.31)	-	0.00085	-	0.49 (97.38)	0.02 (2.48)	0.00082	0.01

13.4.2 Coupled Fitness Landscapes

As explained in Section 13.3, the correlation analysis can also be applied to coupled fitness landscapes by performing a random walk with the move operator of interest on the first landscape, and making a random move on the second landscape every m steps during the random walk on the first landscape. This way a time series of fitness values is generated while taking into account the changes to the first landscape due to moves in the second landscape. The Box-Jenkins approach is then applied to this time series as before, resulting in an appropriate ARMA model to express the correlation structure of the coupled fitness landscape.

Here we use the NKC-model (Section 13.2) to generate coupled fitness landscapes with varying degrees of ruggedness. The parameter values used are as follows: $N = 100$, as with the static landscape. For both K and C the values 0, 2, 10, 20 and 50 are used. We leave out larger values for these parameters, as the results in the previous subsection on static landscapes already showed that there is hardly any correlation on landscapes with a large number of epistatic interactions. Note that for $C = 0$ the NKC-model simplifies to the regular NK-model without external couplings, but this case is included here for comparison. Finally, for m we use 1, 5, 10 and 100.

With the correlation analysis as applied to static landscapes, different (internal) epistatic interactions and also different move operators were used, and the results showed what differences this causes in the correlation structure of the resulting landscapes. Here, we focus primarily on the external epistatic interactions (coupling), and only consider nearest neighbor interactions for the K internal epistatic interactions, and point mutation as the move operator. The main results, split up into three different rates of coevolution (slow, intermediate, and fast), are reviewed here. Full details can be found in [11].

13.4.2.1 Slow Rate of Coevolution

First consider a slow rate of coevolution, in particular $m = 100$. Figure 13.8 shows the correlograms for $K = 2$ (left) and $K = 10$ (right) for the different values of C. As with static landscapes, they taper off to zero, but a subtle and surprising effect shows up. For $K = 2$, the estimated autocorrelations for $C > 0$ (the coupled landscapes) are all well above those for $C = 0$ (the static landscape). It appears that for small values of K, a slow rate of coevolution actually *increases* the correlation length of the coupled landscapes compared to the static one. It is known from genetic algorithm studies that using coevolution can indeed lead to a small increase in the quality of the obtained solutions [7, 17, 29]. Perhaps the above correlation length results can provide an explanation for this phenomenon.

As with static landscapes, the partial correlograms (results for coupled landscapes not shown here) suggest AR(1) models for the coupled landscapes. Figure 13.9 shows the estimated values for the α_1 coefficient represented with a gray scale. As this plot shows, the value of C does not have any significant influence on the value

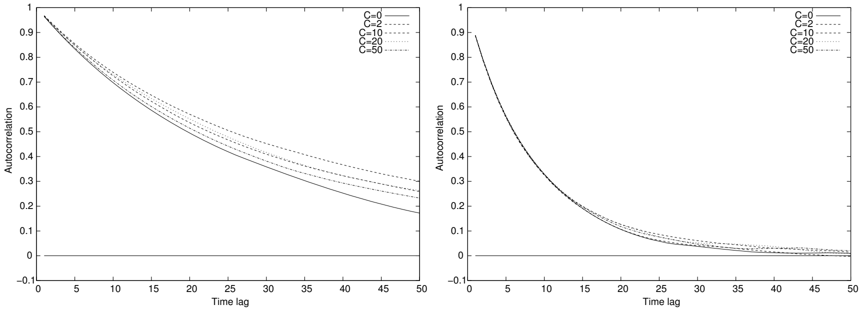


Fig. 13.8 The estimated autocorrelations for NKC-landscapes for $m = 100$, $K = 2$ (left) and $K = 10$ (right), and various values of C

of these coefficients. Within one column (i.e., a particular value for K), there is no perceivable change in the gray levels.

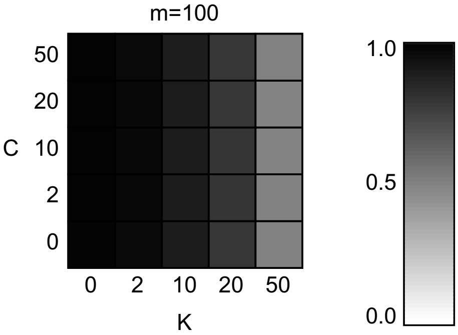


Fig. 13.9 A gray scale plot of the estimated α_1 coefficients for $m = 100$ and different values of K and C

13.4.2.2 Intermediate Rate of Coevolution

Next, consider an intermediate rate of coevolution, with $m = 5$ and $m = 10$. Figure 13.10 shows the correlograms for $m = 10$ and $K = 2$ (left) and $K = 10$ (right). Clearly, an intermediate rate of coevolution causes the correlation lengths to decrease (compared to the static $C = 0$ case), but only significantly so for $C > K$. In other words, the internal epistatic interactions are the dominant ones, and only when the level of external interactions exceeds that of the internal interactions does the coupling have an influence on the correlation structure of the landscape.

Once again, AR(1) models are estimated for all cases, and Figure 13.11 shows the values of the α_1 coefficients represented with a gray scale. These plots confirm

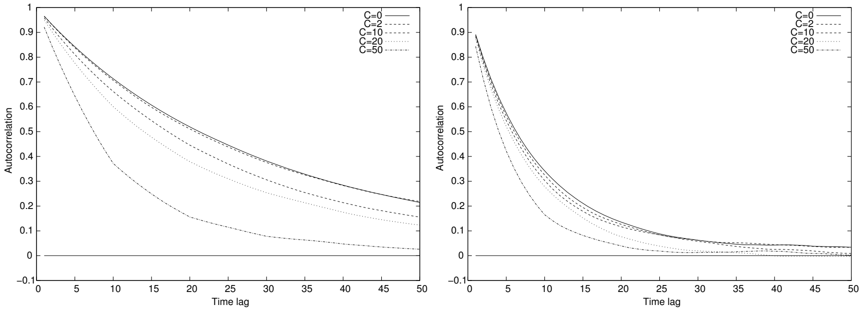


Fig. 13.10 The estimated autocorrelations for NKC-landscapes for $m = 10$, $K = 2$ (left) and $K = 10$ (right), and various values of C

the above results that only for $C > K$ there is an appreciable decrease in the value of these coefficients (which is best visible for $m = 5$).

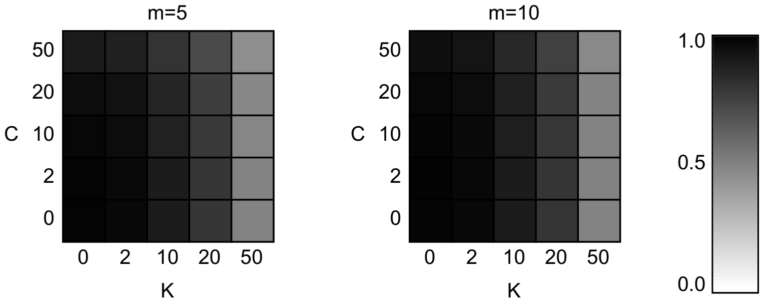


Fig. 13.11 A gray scale plot of the estimated α_1 coefficients for $m = 5$ and $m = 10$ and different values of K and C

13.4.2.3 Fast Rate of Coevolution

Finally, consider a fast rate of coevolution, in particular when the coevolving species moves at the same rate ($m = 1$). Figure 13.12 shows the correlograms for $m = 1$ and $K = 2$ (left) and $K = 20$ (right). The correlation length decreases significantly and ever more drastically for larger values of C, also for $C < K$. In other words, any coupling influences the correlation structure of the landscape, regardless of the level of internal epistasis.

Again, AR(1) models are estimated for all cases, and Figure 13.13 shows the values of the α_1 coefficients represented with a gray scale. It is clear that the values of these coefficients decrease quickly with increasing C for all K.

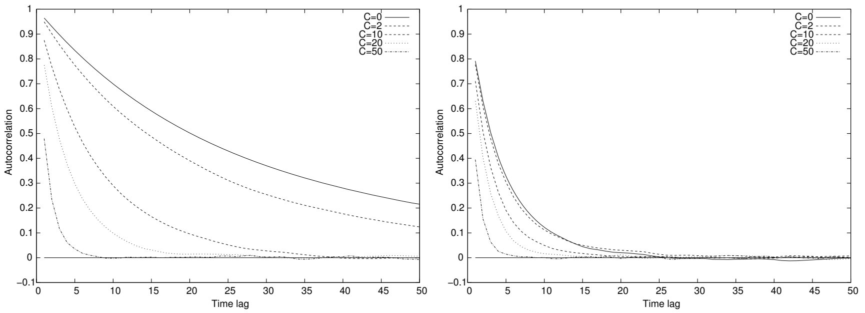


Fig. 13.12 The estimated autocorrelations for NKC-landscapes for $m = 1$, $K = 2$ (left) and $K = 20$ (right), and various values of C

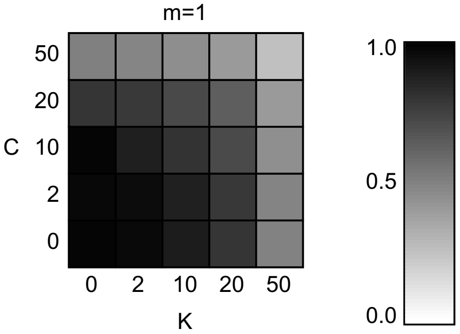


Fig. 13.13 A gray scale plot of the estimated α_1 coefficients for $m = 1$ and different values of K and C

To summarize, for a slow rate of coevolution, there is no appreciable effect on the correlation structure of coupled landscapes, except for very small values of K, where the correlation length actually increases slightly. For an intermediate rate of evolution, there is only an effect when $C > K$, in which case the correlation length of the landscape decreases significantly, with a minor decrease in the α_1 coefficients of the corresponding AR(1) models. For a fast rate of coevolution, any level of external coupling will have a significant negative effect on the correlation length and the value of α_1.

13.4.2.4 Partial Autocorrelations and Coevolution

As with static NK-landscapes, for almost all coupled landscapes only the first estimated partial autocorrelations are significant, indicating that an AR(1) model is the most appropriate. However, there are a few interesting exceptions. Figure 13.14 shows the partial autocorrelations for three different cases: (i) an NK-landscape with $K = 0$, (ii) an NKC-landscape with $K = 20$, $C = 10$, and $m = 5$, and

(iii) an NKC-landscape with $K = 2$, $C = 50$ and $m = 5$. In the first two cases indeed only the first partial autocorrelation is significant (although off the scale on this plot), and the others fall well within the error bound (indicated by the horizontal dashed lines). However, for the third case ($K = 2$, $C = 50$, and $m = 5$), an interesting pattern can be observed, where every fifth partial autocorrelation is significantly larger than the rest, with the effect slowly tapering off for larger time lags. This appears to reflect the fact that $m = 5$ was used in this case, i.e., the second (coevolving) species is mutated every five time steps during the random walk on the first landscape. Several other low to intermediate K and high C cases show a similar pattern. This observation seems to indicate that for highly coupled species with low to intermediate internal epistasis, the partial autocorrelations can provide an indication about their relative rates of evolution.

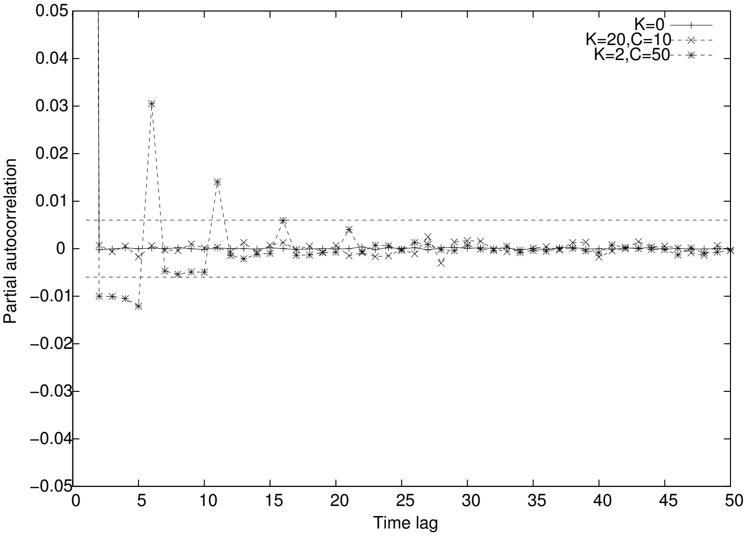


Fig. 13.14 The estimated partial autocorrelations for various NKC-landscapes for $m = 5$

13.5 Conclusions

In this chapter we have provided an overview of a statistical analysis to measure and express the correlation structure of fitness landscapes, and have shown how it can be applied to both static and coupled landscapes. The analysis identifies and estimates an ARMA model that adequately expresses the correlation structure of a given landscape. Also, we have shown (at least on a static landscape) how such a model can be used to generate predictions about an actual search on the landscape, thereby relating the correlation structure of a fitness landscape directly to the performance of an (evolutionary) search algorithm on it.

The results presented here are restricted to NK-landscapes. However, elsewhere it is shown how the correlation analysis can be applied to other types of landscapes as well [10, 12]. Moreover, the results presented here show that the correlation analysis can be applied in a meaningful way to coupled fitness landscapes as well. The results clearly distinguish between different rates of coevolution (i.e., slow, intermediate, and fast), and in some cases the partial autocorrelations might even be able to “recover” the exact rate (i.e., the value of m). In other cases, a slow rate of coevolution actually seems to increase the correlation length of coupled fitness landscapes, which could give an advantage to coevolutionary search algorithms (for which there seems to be at least some evidence).

In conclusion, as with fixed landscapes, the correlation analysis for coupled fitness landscapes generates useful results, and these results can also be directly related to the performance of search algorithms that use a coevolutionary approach. Hopefully these methods and results will provide a constructive contribution to the effort of constructing a true theory of fitness landscapes, relating their structure to search algorithm performance, and increasing our understanding of the (correlation) structure of coupled fitness landscapes and (co)evolutionary computation.

Acknowledgements. The author would like to thank Stuart Kauffman, Terry Jones, and Peter Stadler for stimulating discussions and fruitful collaborations on fitness landscapes over the years, and the editors of the current volume for their kind invitation to contribute this chapter.

The C++ implementation of NK and NKC-landscapes used to perform the experiments described in this chapter can be downloaded from the author’s website at: www.WorldWideWanderings.net.

References

[1] Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press (1996)

[2] Box, G.E.P., Jenkins, G.M.: Time Series Analysis, Forecasting and Control. Holden Day (1970)

[3] Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2003)

[4] Forrest, S., Mitchell, M.: Relative building-block fitness and the building-block hypothesis. In: Whitley, D. (ed.) Foundations of Genetic Algorithms 2, pp. 109–126. Morgan Kaufmann (1993)

[5] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)

[6] Granger, C.W.J., Newbold, P.: Forecasting Economic Time Series, 2nd edn. Academic Press (1986)

[7] Hillis, W.D.: Co-evolving parasites improve simulated evolution as an optimization procedure. Physica D 42, 228–234 (1990)

[8] Holland, J.H.: Adaptation in Natural and Artificial Systems, 2nd edn. MIT Press (1992)

[9] Hordijk, W.: A measure of landscapes. Evolutionary Computation 4, 335–360 (1996)

[10] Hordijk, W.: Correlation analysis of the synchronizing-CA landscape. Physica D 107, 255–264 (1997)

[11] Hordijk, W., Kauffman, S.A.: Correlation analysis of coupled fitness landscapes. Complexity 10(6), 41–49 (2005)

[12] Hordijk, W., Stadler, P.F.: Amplitute spectra of fitness landscapes. Advances in Complex Systems 1(1), 39–66 (1998)

[13] Jones, T.: Evolutionary Algorithms, Fitness Landscapes, and Search. PhD thesis, University of New Mexico, Albuquerque, NM, USA (1995)

[14] Jones, T.: One operator, one landscape. SFI working paper 95-02-025 (1995)

[15] Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Eshelman, L.J. (ed.) Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 184–192. Morgan Kaufmann (1995)

[16] Judge, G.G., Hill, R.C., Griffiths, W.E., Lutkepohl, H., Lee, T.C.: Introduction to the Theory and Practice of Econometrics, 2nd edn. John Wiley & Sons (1988)

[17] Juillé, H., Pollack, J.B.: Coevolutionary learning and the design of complex systems. Advances in Complex Systems 2(4), 371–394 (2000)

[18] Kauffman, S.A.: Adaptation on rugged fitness landscapes. In: Stein, D. (ed.) Lectures in the Sciences of Complexity, pp. 527–618. Addison-Wesley (1989)

[19] Kauffman, S.A.: The Origins of Order. Oxford University Press (1993)

[20] Kauffman, S.A., Johnson, S.: Coevolution to the edge of chaos: Coupled fitness landscapes, poised states, and coevolutionary avalanches. Journal of Theoretical Biology 149, 467–505 (1991)

[21] Kauffman, S.A., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. Journal of Theoretical Biology 128, 11–45 (1987)

[22] Lipsitch, M.: Adaptation on rugged landscapes generated by iterated local interactions of neighboring genes. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 128–135. Morgan Kaufmann (1991)

[23] Manderick, B., de Weger, M., Spiessens, P.: The genetic algorithm and the structure of the fitness landscape. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 143–150. Morgan Kaufmann (1991)

[24] Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press (1996)

[25] Mitchell, M., Forrest, S., Holland, J.H.: The royal road for genetic algorithms: Fitness landscapes and GA performance. In: Varela, F.J., Bourgine, P. (eds.) Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, pp. 245–254. MIT Press (1992)

[26] Stadler, P.F.: Towards a theory of landscapes. In: Lopéz-Peña, R., Capovilla, R., García-Pelayo, R., Waelbroeck, H., Zertuche, F. (eds.) Complex Systems and Binary Networks, pp. 77–163. Springer (1995)

[27] Stadler, P.F.: Fitness landscapes. In: Lässig, M., Valleriani, A. (eds.) Biological Evolution and Statistical Physics. Lecture Notes in Physics, pp. 183–204. Springer (2002)

[28] Weinberger, E.D.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cybernetics 63, 325–336 (1990)

[29] Werfel, J., Mitchell, M., Crutchfield, J.P.: Resource sharing and coevolution in evolving cellular automata. IEEE Transactions on Evolutionary Computation 4(4), 388–393 (2000)

[30] Wilson, S.W.: GA-easy does not imply steepest-ascent optimizable. In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 85–89. Morgan Kaufmann (1991)

[31] Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones, D.F. (ed.) Proceedings of the Sixth International Congress on Genetics, vol. 1, pp. 356–366 (1932)

Chapter 14
Dynamic Real–Valued Landscapes and the Optimization Performance

Krzysztof Trojanowski

Abstract. A large number of heuristic optimization algorithms for dynamic optimization has already been proposed. The aim of this chapter is to discuss methods of evaluating their efficiency. Thus, the following issues have to be considered: (i) measures for performance and associated measurement methods, (ii) dynamic benchmarks and different types for implementing changes, and (iii) the role of time and uncertainty originating from the measurement method. In this chapter the issues are discussed for the case of single-objective optimization in dynamic multimodal fitness landscapes.

14.1 Introduction

Optimization assumes that a primary problem to be solved is given and that there is a function F which assigns values for each of the possible and feasible solutions of the primary problem. The function F introduces an ordering relation for the set of the solutions $\mathscr{D}$. We assume that the cost of generating a solution from the set $\mathscr{D}$ is negligibly small and the main aim of the optimization task is just to find the optimal solution in $\mathscr{D}$, that is, the largest or smallest element of the order.

14.1.1 The Objective

The need for the most appropriate and advantageous solution of our problems demands more and more effective searching algorithms and methods. Evaluation of them is based on their ability to cope with real-world problems as well as artificially

Krzysztof Trojanowski
Institute of Computer Science, Polish Academy of Sciences,
Jana Kazimierza 5, 01-248 Warsaw, Poland
Cardinal Stefan Wyszyński University,
Faculty of Mathematics and Natural Sciences,
Wóycickiego 1/3, 01-938 Warsaw, Poland
e-mail: trojanow@ipipan.waw.pl

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_14, © Springer-Verlag Berlin Heidelberg 2014

generated test-cases. Experimental evaluation of the algorithm efficiency depends on the difficulty of test-cases and the evaluation measure as well as the measurement method.

In dynamic optimization the main aim is to find such a solution from the feasible part of the search space $\mathscr{D}$ which represents an optimum of a given objective function F defined for the space and we assume that the function's fitness landscape[1] can change during the optimization process. In the model $\mathscr{M}$ [28] the dynamic environment P is defined as $\mathscr{M}(P) = (\mathscr{D}, F, \mathscr{C})$ where all three components can change: the domain, that is, the search space $\mathscr{D}$, the objective function, that is, the fitness function F and a set of constraints $\mathscr{C}$ the solution must satisfy. This way, we obtain quite a large number of possibilities for the environment dynamics, however, in practice, the model can be divided further into just six categories, since there are two classes for fitness functions and three classes for sets of constraints:

- the fitness function F may or may not depend on the time variable t, and
- the set of constraints $\mathscr{C}$ can be: (1) empty, (2) non-empty and time independent, and (3) non-empty and time dependent.

Changes in the search space are usually not considered.

In the further text we study just one of the cases, that is, the case where the fitness function can change whereas the domain is constant and the set of constraints is empty. Precisely, we are interested in a single-objective optimization in dynamic multimodal fitness landscapes which is one of four types of uncertainty according to a classification proposed in [12]. In one of the remaining three types, noise is present in the optimized function. Uncertainty of the other type is present when for some reason (for example, high computational costs) we use approximated evaluation of function values. The last type of uncertainty occurs when the main aim is to find a solution not only of the highest quality, but – more importantly – one whose neighbors are equally good, that is, when the most important issue is the robustness of the returned solution.

When we plan to compare empirically the effectiveness of selected optimization algorithms, a number of dynamic problems is needed, that is, a benchmark generator which produces artificial multimodal test-cases of a satisfying level of difficulty. In the literature, numerous dynamic benchmarks can be found which can be further divided into two major categories [37]. Construction of the test-cases from the first category is based on the simple idea of switching between a number of earlier prepared stationary problems. The second category represents a more sophisticated approach where the fitness function is defined with different formulas for different parts of its domain, that is, there is a number of component functions and each of them can change independently according to its own dynamics. Below, just the latter category is a subject of our particular interest.

[1] The fitness landscape metaphor provided by Sewall Wright in the mid-1930s was formally adapted for search algorithms by Jones [14]. The metaphor allows us to express an optimization problem as a surface constructed by the use of a search space and an evaluation function but also some features of the selected search algorithm. In the landscape the higher locations indicate the better solutions.

14.1.2 The Algorithm Performance – Static versus Dynamic Optimization

In the Case of static optimization, the quality of the algorithm usually equals the quality of the best found solution which was obtained with respect to some computational cost constraints. The values of the best found solutions returned by different algorithms working with the same computational cost constraints represent also the quality of these algorithms and allow to find the winner among them. It is necessary to mention here that in the case of performance evaluation of heuristic algorithms we need to take into account their stochastic nature, that is, the fact that the same algorithm running twice for the same configuring parameters and the same test-case can return two more or less different results. Due to this the simulations are usually repeated and the series of results undergoes statistical analysis (for example, the mean value is returned).

In the case of dynamic optimization, the definition of a fitness function is extended by one more parameter – the time. It is also assumed that no rationale is given which would allow for anticipation of the fitness landscape topology changes. Due to this we cannot expect that solutions are defined as functions where one of the parameters represents time. Instead, we want the algorithm to move its current best solutions through the landscape toward the global optimum efficiently when the fitness function is stable and we expect flexible reaction and immediate adaptation in the case of changes. Under these circumstances the value of the best found solution no longer can be interpreted as a quality measure of the algorithm. Additionally, the time is no longer tantamount to the computational cost of the optimization process like it is in the static optimization. Now the time is closely related to change frequency and we neither can estimate the length of the time interval when the changes appear nor their frequency, strength and other features. These differences between the properties of static and dynamic environments demand more specific evaluation criteria of the algorithm efficiency for the latter case.

14.1.3 Outline of the Chapter

The chapter is organized as follows. Different types of measurements performed during the algorithm and a discussion about what we can do with the gathered data is given in Section 14.2. Section 14.3 presents selection of existing benchmark generators described in publications. In Section 14.4 the influence of the constrained computational resources on the expected value of the algorithm outcome is discussed, that is, how to cope in the case where the search process can be interrupted in any moment and the user requests the current best solution. Section 14.5 concludes.

14.2 The Heuristic Algorithm Performance Measures

In studying properties of measures of the algorithm's performance, we assume that the heuristic algorithm is an iterative process where a set of steps is repeated in a loop. Hence, the execution of the algorithm can be called a *search process* where a set of initial feasible solutions of rather poor quality undergoes iterative heuristic perturbations. This means that heuristic algorithms develop solutions, however, there is no guarantee that the global optimum can be found for any finite number of the algorithm's iterations, that is, in any finite time[2]. In dynamic optimization the time parameter is present both in the process of search and in the function which defines dynamics of changes in the fitness landscape. For the sake of unification of the two time parameters we need to make two assumptions. First, it is assumed that subsequent changes appear at any number of time units and this number may or may not be constant but in every case it must be an integer and non-zero value. Second, it is assumed that a unit of time is equivalent to a unit of computations executed by the algorithm.

To make comparing heuristic algorithms fair it is assumed that subsequent measurements are observed between the same numbers of computation units for each of the compared algorithms. This assumption is fundamental for the theory of algorithm performance comparison. The value of an algorithm lies in its ability to discover knowledge from known facts and to use the knowledge for further fruitful mining in the other unknown facts. The values of solutions from the search space are the facts. Appropriate and fast estimation of suboptimal solutions depends on the knowledge build from the facts. The heuristic optimization resembles the game "20 questions" where the ability to ask the right questions is very important in order to get the right information and in reaching the purpose. Every request for the evaluation of a solutions is like a request for a hint. Every algorithm can ask for different hints, that is, ask for evaluation of different points in the search space. The one who needs less hints for a more effective estimation is the winner. So, the subjects of the competition between the algorithms are (a) the number of hints necessary to reach the satisfying suboptimal solution and (b) the quality of the state of knowledge after each subsequent hint. Therefore, the fair competition must be based on the simultaneous measurement performed between the same numbers of subsequent hints. When in experimental research the numbers of computation units between subsequent measurements are not the same, it would be possible to manipulate the algorithm parameters in a way which can lead to false conclusions about the algorithm efficiency. For example, when it is assumed that just the number of iterations between subsequent changes in the environment should be the same, this would allow tuning of the number of fitness function calls in a single iteration in favor of one algorithm and at the expense of another.

[2] This is in contrary to the formal definition of an *algorithm* which says that this is a finite list of steps that lead to the result. Thus, the term *method* would probably be much more appropriate in this case. Instead of this due to historical reasons we say *heuristic algorithm* which means a particular instance of an optimization heuristic.

The quality of an iterative algorithm is a function of its readiness to return a valuable solution from the set $\mathscr{D}$ upon the user request at any time. The quality measure can be defined as:

- quality of the current best found solution $\mathbf{x}^j_{\mathrm{best}}$, that is, the value of $F(\mathbf{x}^j_{\mathrm{best}})$ where j is a number of change in the landscape which occurred during the algorithm run,
- error level, that is, the difference between the quality of the global optimum $F(\mathbf{x}^{*j})$ and the quality of the current best found solution $F(\mathbf{x}^j_{\mathrm{best}})$ evaluated for the j-th shape of the fitness landscape,
- Euclidean distance between the current best found solution $\mathbf{x}^j_{\mathrm{best}}$ and the global optimum $\mathbf{x}^{*j}$ evaluated for the j-th shape of the fitness landscape.

Each of these attributes can be measured in subsequent time units of the optimization process. Obtained series of values can be either presented as a graph or undergo statistical analysis specific for data sets to obtain a single value which is more useful for comparisons.

The last issue is about the identification of changes. Usually, the author of the benchmark knows when and how often the change appears. However, this knowledge does not have to be shared with the algorithm. In this case the algorithm has to have an embedded strategy for detecting changes in the environment. There are publications [21, 22, 24] devoted to methods of identification of changes in the dynamic environment during the process of search. However, in the further text we shall assume that the optimization algorithm is informed of the change as soon as it occurs, and no additional computational effort for its detection is needed.

14.2.1 Quality of the Current Best-Found Solution

A graph of a series of current best solution values measured in subsequent generations (or a graph of the mean values obtained from a series of experiments) is the most intuitive way of presenting results originating from the publications devoted to static optimization (see for example [4, 5, 6, 7, 8, 10, 11, 16, 17, 20, 32, 33]). This graph shows the progress of the search process. However, as it was mentioned earlier, in this case it is necessary to guarantee the same computational cost between subsequent measurements for all compared algorithms. It is also worth noting that in some papers graphs of the average values of all solutions or graphs of the worst solution in the population were analyzed [7, 10, 16, 17].

However, graphs depicting current best (or mean or worst) solutions are hardly comparable to each other. It is much easier to indicate the better algorithm when the quality is represented by a single value. Therefore, for evaluating the algorithm's performance, two other measures proposed by De Jong in 1975 for static optimization were also applied, namely offline and online performance [3, 11, 32, 33]. The offline performance represents the best value in the current population averaged over the entire run. This shows the efficiency of the algorithm during the process of search. Morrison [18] developed this measure further: the obtained offline performance is averaged over the number of experiments across the entire range of landscape dynamics. The outcome is called the collective mean fitness. The latter measure, that is, the online performance, is the average of all fitness function evaluations of the entire run. It shows the impact of the population on the focus of the search.

The last issue concerns the problem of comparability of best-found solution values between test-cases. Since the fitness values of the optimal solutions for different instances of testing environments usually differ from each other, the obtained values of current best-found solutions are hardly comparable. Therefore, a normalization of the measured values is necessary. In Weicker [34] two cases of normalization are discussed: the first one, where the values of the best and the worst values in the entire search space are known, and the other one, where the values are unknown. The former case is simple, since the accuracy of a solution $\mathbf{x}_j$ for the j-th shape of the fitness landscape is just the result of normalization transformation,

$$acc(\mathbf{x}_j) = \frac{F(\mathbf{x}_j) - F_{\min,j}}{F_{\max,j} - F_{\min,j}}, \tag{14.1}$$

where $F_{\max,j}$ and $F_{\min,j}$ represent best and worst known values for the j-th shape of the landscape, respectively.

In the latter case a window based accuracy is proposed, which normalizes values of the solutions respectively to the maximum and minimum values among all the solutions which appeared in the population for a selected time window. The upper boundary of the time window is just the present iteration and the length of the window is constant and the same for all the test-cases. The two measures can be the basis for another optimization process parameters, like stability [34], that is, the influence of the changes on the accuracy. Stability of the algorithm simply represents the maximum difference between the values of the accuracy (or the window based accuracy) measured in subsequent time steps.

14.2.2 Error Level

A difference between values of the current best found solution and the optimum represents another measure used for comparisons. This is particularly useful for evaluation of the results obtained for different instances of the dynamic environment. The difference shows a current lack in quality as distance to the optimum and is less sensitive to the amplitude of optimum quality levels in subsequent landscapes.

The difference between values of the current best found solution and the optimum averaged over the entire run presented in [29] was one of the first propositions. However, this approach was constrained just for steady-state algorithms, that is, those which never lose already found best solutions but continuously improve them in subsequent iterations. The same restriction concerned also the second measure proposed in this paper, which was a mean of a difference between the value of the current best solution in the population of the "just before the change" generation and the optimum value averaged over the entire run. At the same time, a more flexible measure was proposed, which is called the offline error [5] and represents the average deviation from the optimum of the best solution value found since the last change in the fitness landscape's shape. After every change, the best solution value is forgotten and the new candidate for the best solution is selected among the current set of solutions reevaluated respectively to the new shape of the landscape. Sample series of the algorithm outcome for a dynamic maximization problem where the fitness function changed three times every 15 generations are depicted in Figure 14.1. The series of the current best solution values registered every iteration is represented as a dashed line. The solid line segments represent values of the optimum after subsequent changes. The difference which contributes in the offline error evaluation is represented as a distance between the dashed and the solid line. However, not in every case. In the case of deterioration of the value of the current best solution the difference between the quality of the best-found solution since the last change is rather taken into account. The deterioration is possible for heuristic algorithms without a steady-state strategy of the solution population management. In the figure, the total of differences measured during the experiment is represented as a gray area. Eventually, the requested offline error value is the total of the differences averaged over the entire run.

Ten years later, for the purpose of a special session and competition on "Evolutionary Computation in Dynamic and Uncertain Environments"[3] four other parameters were defined based on the error values: the best, the worst, the mean and the standard deviation value of offline error for a minimization problem obtained for a series of fitness function changes in repeated experiments:

$$Avg^{\text{best}} = \frac{\sum_{l=1}^{N_{\text{exp}}} \min_{j=1,\dots,N_{\text{ch}}} E_{\text{last}}^{l,j}}{N_{\text{exp}}}, \tag{14.2}$$

$$Avg^{\text{mean}} = \frac{\sum_{l=1}^{N_{\text{exp}}} \sum_{j=1}^{N_{\text{ch}}} E_{\text{last}}^{l,j}}{N_{\text{exp}} \cdot N_{\text{ch}}}, \tag{14.3}$$

$$Avg^{\text{worst}} = \frac{\sum_{l=1}^{N_{\text{exp}}} \max_{j=1,\dots,N_{\text{ch}}} E_{\text{last}}^{l,j}}{N_{\text{exp}}}, \tag{14.4}$$

$$STD = \left(\frac{1}{N_{\text{exp}} \cdot N_{\text{ch}} - 1} \sum_{l=1}^{N_{\text{exp}}} \sum_{j=1}^{N_{\text{ch}}} (E_{\text{last}}^{l,j} - Avg^{\text{mean}})^2 \right)^{-1}, \tag{14.5}$$

where N_{exp} is the number of restarted experiments for the same control parameter settings of the algorithm and the same test-case, N_{ch} is a number of changes in the fitness function that appear during a single experiment run, and E_{last}^{j} is the absolute function error value.

$$E_{\text{last}}^{j} = |F(\mathbf{x}_{\text{best}}^{j}) - F(\mathbf{x}^{*j})|. \tag{14.6}$$

[3] IEEE CEC-2009, Trondheim, Norway, 2009, 18-21 May.

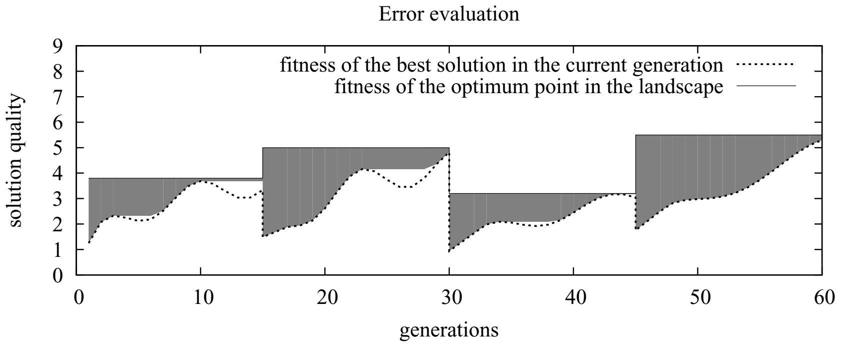


Fig. 14.1 An example series of values measured during the single experiment of continuous optimization in the dynamic fitness landscape. The dashed line represents values of the best solutions in the subsequent iterations, the solid horizontal line segments represent values of the optimum after subsequent changes and the gray area represents set of values summarized and averaged over the entire experiment to be returned as an offline error value

14.2.3 Euclidean Distance to Optimum

Applications of the Euclidean distance in the algorithm performance evaluation measures appear seldom in the literature. The distance between the current best-found solution and the optimum was employed in [35], and the distance of the mass center of the population to the optimum in [26]. In [30] the authors applied Euclidean distance as a measure of unsatisfied realizations in a search space, that is, a mean deviation of the Euclidean distance between the current best found solution and the optimum in the selected time interval below the acceptable level.

However, the obtained minimum Euclidean distance to the optimum for the algorithm effectiveness is never applied as a primary criterion because this demands very specific properties of the search space attributes. Clearly, it must be guaranteed that the change of value for one of the dimensions is equally important for the user as the change for another dimension. In other words, all the solutions located on a hypersphere surrounding an optimum has to be of the same quality. Otherwise, the measure does not allow to differentiation between the better and the worse solution in the search space. Due to this, the Euclidean distance to optimum should be applied as a secondary measure to adjudicate which of any two solutions is better just in the cases where the first measure failed.

14.2.4 Statistical Analysis of the Measurements

Due to the stochastic nature of heuristic algorithms we need to repeat the experiments and statistically analyze the output data. The analysis of a series of measurements is concerned with the basic problem of coping with differences between two means obtained for two independent data series. In the case where the means are similar it is necessary to verify if the differences between the means can be declared statistically significant. There is a number of classic significance tests which can be applied assuming that the sample is without biases (e.g. Student's t-tests).

An interesting performance measure based on statistical analysis can be found in [2]. The measure, called the dissimilarity factor, is a weighted sum of two other factors. They are evaluated with the use of current optimum values $F(\mathbf{x}^*)$ and the values of the current best found solution $F(\mathbf{x}_{\text{best}})$ obtained for all subsequent iterations of a single experiment run. The first factor represents the effectiveness of the algorithm which is an Euclidean distance between the values in the two series:

$$c_{\text{dist}} = \sqrt{\sum_{n=1}^{N_{\text{gen}}} (F(\mathbf{x}_{\text{best}}^n) - F(\mathbf{x}^{*n}))^2} \tag{14.7}$$

The second factor represents the algorithm ability to recover after a change which is a cross correlation value between the values in the two series:

$$c_{\text{corr}} = \frac{\sum_{n=1}^{N_{\text{gen}}} ((F(\mathbf{x}_{\text{best}}^n) - m_{\text{best}}) \times (F(\mathbf{x}^{*n}) - m_{\text{opt}}))}{\sqrt{\sum_{n=1}^{N_{\text{gen}}} (F(\mathbf{x}_{\text{best}}^n) - m_{\text{best}})^2} \times \sqrt{\sum_{n=1}^{N_{\text{gen}}} (F(\mathbf{x}^{*n}) - m_{\text{opt}})^2}} \tag{14.8}$$

where n is the number of the iteration, N_{gen} is a total number of iterations executed during a single experiment run, m_{best} and m_{opt} are the estimated means of $F(\mathbf{x}_{\text{best}})$ and $F(\mathbf{x}^*)$ values, respectively. The means are calculated by taking the average of all the respective values of $F(\mathbf{x}_{\text{best}})$ or $F(\mathbf{x}^*)$ measured during a single run of the experiment. When the value of c_{corr} is close to 1 the series of values are very similar which means that the algorithm recovers efficiently after the change, otherwise when it is close to -1, the series are different and this means that there are some problems with the algorithm's recovery.

Another statistical approach to evaluation of the algorithm ability to recover after a change can be found in [1]. Here a linear regression model is applied to the measurements performed just before the change. The slope of the regression line represents robustness of the algorithm. To obtain this, in the first step the values of the accuracy [34] for each of the changes are averaged over a number of independent runs of the experiment. In the second step a regression model is build where the value of the standardized regression coefficient β describes the algorithm efficiency. In the model the set of accuracy values obtained for all the changes becomes the set of explanatory variables whereas the approximation to the overall accuracy represents a scalar dependent variable. A positive value of β indicates that the algorithm recovers efficiently, whereas a negative value means degradation of the solution quality.

14.2.5 Description of Dynamics

14.2.5.1 The Model

The model of a dynamic environment defined as $\mathscr{M}(P) = (\mathscr{D}, F, \mathscr{C})$ implicitly assumes that the dynamics can occur in any of the components in the run-time of the algorithm. Another model which explicitly defines the presence of time and describes the full range of changes which can occur in the environment has been proposed by [23]. In the model a dynamic landscape is defined as a quintuple, $\Lambda_D = (S, \eta, \Gamma, F, \varphi)$, where S represents the search space, $\eta(\mathbf{x})$ defines the neighborhood structure for the solutions from the search space, Γ represents measuring and ordering scale for the changes, F is the set of fitness functions f such that $f : S \times \Gamma \to \mathrm{I\!R}$, that is, all the functions which define the fitness landscape during the entire time period being the subject of interest, and the last component, φ, is the transition map for the fitness functions from F; that is, φ describes how the landscape transforms from one shape to another. In this model the two components, Γ and φ, allow for a formal definition of dynamics in F. Besides, the model classifies benchmarks respectively to the discrete or continuous nature of time and space. The search space can be discrete or continuous whereas the time can be static, discrete or continuous.

For dynamic optimization a number of features that differentiate dynamic problems were proposed. For example, in [29] we can find the following two: continuous or discrete nature of changes and regularity of changes. The regularity divides changes into predictable ones, that is, regular or even cyclic and non-predictable changes, that is, random changes. Besides, both continuous and discrete changes can be classified further into three sub-types [30]: adiabatic changes, indirect ones, and turbulent ones. Each type of the above mentioned dynamic features can concern any of the benchmark parameters, for example, we can define a benchmark where the coordinates of local optima in the landscape change randomly and continuously whereas their heights vary cyclically and in a discrete manner.

The most in-depth analysis and classification of dynamic fitness landscape features can be found in [23]. The features are divided into two groups: topological and dynamical. Topological features are represented by the number of local optima, their distribution and properties of their basins of attractions. Dynamical features concern speed, that is, the number of changes per time interval, frequency, severity or strength of the changes, level of graduality, and predictability.

14.2.5.2 The Time Parameter

The parameters related to the dynamics can belong both to the algorithm and the benchmark. At the beginning of analysis we need to assume the discrete nature of the time t just for measurement purposes. Clearly, a single time step has to be correlated with a unit of algorithm computations. A single unit should lead the algorithm from one state to another, that is, to the state where the algorithm's knowledge about the problem is at least a bit different from the state it had previously. On the other side, this should also be one of the indivisible steps of the algorithm. A single fitness function call is a good candidate. Besides, it often happens that the fitness function call is truly one of the most computationally expensive and also indivisible steps of the algorithm. When we take the fitness function call as a time unit, even continuous changes occurring between subsequent function calls become negligible from the algorithm point of view. In this case the only important information is the state of the fitness landscape just before subsequent fitness function calls. The fact is, that in most publications the unit cost of computations is represented just by the fitness function call, that is, for example the time span of a single execution of the loop equals the number of the calls. All the other steps in the loop are regarded as negligible.

So, the first parameter of the landscape dynamics is the number of the fitness function calls, that is, the length of the time interval between subsequent changes $M \geq 1$. The length of subsequent intervals can be regular or not: the number of calls can be predictable (constant, or varying cyclically), or nonpredictable (chaotic or random). This allows definition of parameters such as change frequency or maximum and minimum time span between subsequent changes. A number of changes N_{ch} occurring in the environment during a single experiment is also closely related to the above mentioned change period.

14.2.5.3 The Landscape Features

Following the classification given in [23] we divide the set of landscape features into topological and dynamical subgroups. In other words, when talking about dynamic environments we need two define two things: what can change in the landscape and how does it change.

The first parameter from the first subgroup should be the number of the search space dimensions N_{dim} as well as upper and lower boundaries defined for each of the dimensions. Changes of the search space practically redefine the problem, and in this case the search process should be started from scratch. However, this is not a theoretical situation and one can easily imagine real-word problems where the boundaries of the search space dimensions vary during the ongoing search process. The number of local optima is the next one. Due to the fact that in many landscape generators the landscape is composed of a number of component functions this parameter can be replaced by the number of components N. The two numbers do not have to be identical because some components can be defined as multimodal functions. Even in the case of unimodal components identity is not guaranteed in the case where components can overlap. Other parameters define the topological features of the types of components. They are represented as component function parameters and define coordinates, shape, slope and height of the landscape components. Each of the types has its own set of parameters with respect to the function formula.

The second subgroup consists of parameters of the component transformation mechanisms, that is, scaling, translation and rotation. These are scaling factor, translation vector and rotation matrix, respectively. Each of them has assigned a pair of boundaries which define feasible intervals of values. There are also parameters which define dynamics of the transformation mechanism parameters, that is, how the mechanisms are applied and how they change in time. The transformation mechanisms can be applied regularly or not and the dynamics can be common for all the components or individually defined. The strength of transformation can be predictable (constant, or varying cyclically), or nonpredictable (chaotic or random). A discussion of the example rules and their application can be found in Section 14.3.

14.3 Real-Valued Dynamic Optimization Landscapes

14.3.1 Landscapes with Symmetric Components

The first landscapes designed for real-valued dynamic optimization were based on a simple idea of a number of unimodal component functions defined for the given search space where each of the components occupies its part of the space. The components represented by peaks, cones or hills (see Figure 14.2) are assembled to create a multimodal fitness landscape. Random changes in the component width, height or coordinates were the first types of changes tested in the landscapes.

14.3.1.1 Moving Peaks Landscape

Landscape generators were where the landscape is created using a number of component unimodal functions [5, 19]. In the case of DF1 [19] the landscape is defined for a 2-dimensional search space and consists of a number of cone functions. In the case of MPB (Moving Peaks Benchmark[4]) [5] the search space is five-dimensional and the landscape consists of a number of peaks (scenario 1) or cones (scenario 2 and 3). In both approaches every i-th peak has its height h_i, width w_i, and coordinates of its optimum point $\mathbf{cmax}_i$.

The i-th peak function is defined as follows:

$$f_i(x_1,\cdots,x_n) = \frac{h_i}{1+w_i\cdot\sum_{d=1}^{n}(x_d-\mathbf{cmax}_i[d])^2}, \tag{14.9}$$

whereas the i-th cone function is:

$$f_i(x_1,\cdots,x_n) = h_i - w_i\cdot\sqrt{\sum_{d=1}^{n}(x_d-\mathbf{cmax}_i[d])^2}. \tag{14.10}$$

The graphs of the two types of components are depicted in Figure 14.2.

The final value of a solution $\mathbf{x}$ in the search space is evaluated as follows:

[4] The name of this test-case generator contains the word *Benchmark*, which means just an example test problem with often very special properties. However, in this particular case it is justified to call this a *landscape* generator, because MPB generates search spaces with a notion of adjacency, nearness and distance. Additionally, for every solution from the space there is a fitness value defined that is the result of the evolutionary optimization dynamics. Clearly, the test-cases generated with MPB have all the properties that fitness landscapes have in evolutionary computation.

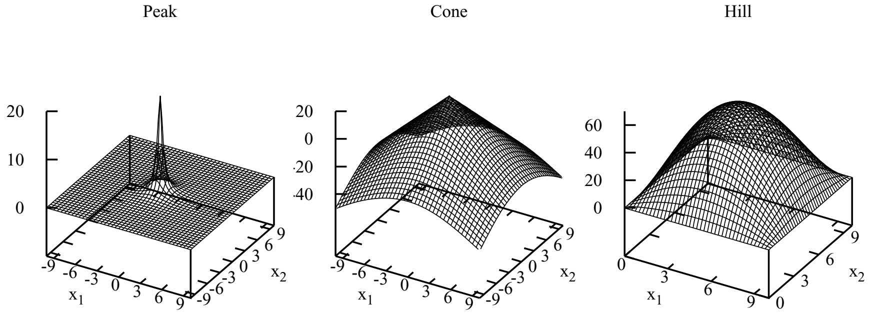


Fig. 14.2 Peak, cone and hill defined in 2-dimensional search space

$$f(\mathbf{x}) = \max_{i=1,\dots N} f_i(\mathbf{x}). \tag{14.11}$$

The i-th component movement is performed simply by change of the coordinates of its optimum point $\mathbf{cmax}_i$:

$$\mathbf{cmax}_i(t) = \mathbf{cmax}_i(t-1) + \mathbf{v}_i(t) \tag{14.12}$$

$$\mathbf{v}_i(t) = \frac{v_{\text{length}}}{|\mathbf{r} + \mathbf{v}_i(t-1)|}((1-\lambda)\mathbf{r} + \lambda \mathbf{v}_i(t-1)) \tag{14.13}$$

where v_{length} is the maximum step length, $\mathbf{v}_i(t)$ is the current translation vector, $\mathbf{r}$ is the vector of independent random variables from the interval $[0,1]$, and λ defines correlation between subsequent versions of the vector $\mathbf{v}$: $\lambda = 0$ means no correlation, that is, subsequent vectors $\mathbf{v}$ are independent. In the case when the new coordinates of the optimum point are out of the feasible region of the space the calculation is repeated.

The next values of the component height and width are evaluated as follows:

$$h_i(t) = h_i(t-1) + s_{\text{h}} \cdot \sigma \tag{14.14}$$

$$w_i(t) = w_i(t-1) + s_{\text{w}} \cdot \sigma \tag{14.15}$$

where $h_i(t)$ is the new height, $w_i(t)$ is the new width, s_{h} is the height change coefficient, s_{w} is the width change coefficient, and σ is a random value from [0,1].

Test-cases based on scenario 2 with $\lambda = 0$, that is, with no correlation between subsequent versions of $\mathbf{v}$ are selected for experiments in the majority of published research where the MPB generator was applied. In the test-cases subsequent localizations of the optimum point generate paths which have different features depending on v_{length} as one can see in the pictures in Figure 14.3. For every picture the first localization was generated randomly within the range [30,50] for both coordinates. Particularly for larger values of v_{length}, that is, when the optimum point walks easily over the entire search space the adverse phenomenon of optimum point attraction to the border areas of the space can be observed.

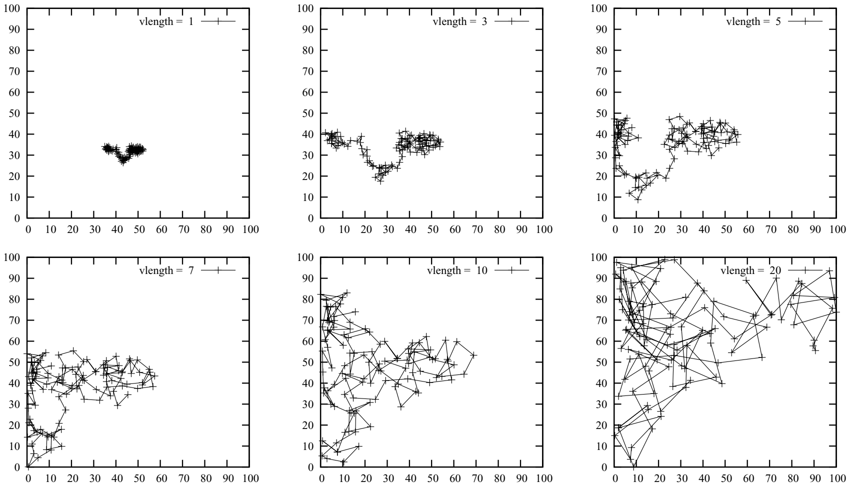


Fig. 14.3 Paths with 110 subsequent localizations of the optimum point for v_{length}: 1,3, 5, 7, 10 i 20 in 2-dimensional search space

14.3.1.2 Simple Test Case Generator

The simple test case generator (STCG) [29] defines a dynamic landscape in two-dimensional search space. Construction of the landscape divides the space into a number of subspaces just like squares in a chessboard. For every subspace $S_{a,b}$ a hill function (Figure 14.2) is defined:

$$f_{a,b}(x_1,x_2) = \begin{cases} \alpha_{a,b}(c_{\text{r}}(a)-x_1)(x_1-c_{\text{l}}(a))(c_{\text{t}}(b)-x_2)(x_2-c_{\text{b}}(b)) & \text{if } (x_1,x_2) \in A_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{14.16}$$

where $\alpha_{a,b} = 16/[(c_{\text{r}}(b)-c_{\text{l}}(b))^2(c_{\text{t}}-c_{\text{b}})^2]$, a is a column number and b is a row number in the chessboard, $c_{\text{r}}(a)$ and $c_{\text{l}}(a)$ represent the right and left boundaries of the a-th column, and $c_{\text{t}}(b)$ and $c_{\text{b}}(b)$ represent the top and bottom boundaries of the b-th row.

The final value of a solution **x** in the search space is evaluated as follows:

$$f(\mathbf{x},t) = \sum_{a=1}^{N_{\text{col}}} \sum_{b=1}^{N_{\text{row}}} p_{a,b}(t) \cdot f_{a,b}(\mathbf{x}) \tag{14.17}$$

where N_{col} and N_{row} represent total numbers of columns and rows, respectively, $p_{a,b} \in [0, p_{max}]$ represents the height of the (a,b)-th hill in the landscape; in the benchmark it was assumed that $p_{max} = 100$. Application of different rules for evaluation of $p_{a,b}$ allows for generation of landscapes both with cyclic or acyclic changes.

The idea of a landscape which is build of varying but immovable components can also be found in the multi-objective optimization-based dynamic test problems generator [13]. In this approach each of the components occupies its part of the space and is controlled by a vertical scale coefficient varying within the range from 0 to 1. When the coefficient equals zero the component completely disappears from the landscape, that is, the landscape is flat. In the opposite case the component participates in the landscape with its full height. The final value of a solution **x** is also evaluated as defined in Equation (14.11).

14.3.2 Landscapes with Non-symmetric Components

Unimodal functions with a symmetric graph shape allow for horizontal and vertical scaling as well as for shifting throughout the search space. However, the rotation of such a shape has no effect. An example of a benchmark build of component functions with a non-symmetric shape of the landscape can be found in [38] (the full description of this benchmark generator was published three years later in [9]) called *the max-set of Gaussian landscape generator*.

The unimodal component is defined with the Gaussian function:

$$g(\mathbf{x}) = \left[\frac{1}{(2\pi)^{N_{\mathrm{dim}}/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu)\Sigma^{-1}(\mathbf{x}-\mu)^T)\right]^{\frac{1}{N_{\mathrm{dim}}}} \qquad (14.18)$$

where μ is a vector of means of size N_{dim} and Σ is a covariance matrix of size $N_{\mathrm{dim}} \times N_{\mathrm{dim}}$. Like in the previous approach the final value of a solution **x** is evaluated according to Equation (14.11).

Each of the components is controlled by the following three parameters: μ, Σ, and a vertical scale coefficient. This allows for the change of the asymmetric shape of the function graph as well as its height. However, there is no possibility to move the component to another point in the search space (like in STCG).

14.3.3 Transformation Matrices

An approach opposite to the construction of landscapes presented in previous sections can be found in [27, 15]. Previous landscapes consisted of a number of components where variability is the source of dynamics in the landscape. In this approach the components are constant and no change is applied to them at any time. Instead, the source of the landscape dynamics is the variability of the coordinates which have to be evaluated. The procedure of evaluation of a point **x** from the search space is extended by one more step. Before the evaluation begins the transformed coordinates of point **x** are generated: $\mathbf{x}' = \mathbf{x} \cdot A$, where A is a transformation matrix. Then, $\mathbf{x}'$ is the subject of evaluation, however, the returned value is assigned to the solution **x**, that is, the unmodified vector of coordinates. After that $\mathbf{x}'$ is no longer useful and is forgotten. Such an extended procedure of evaluation is applied every fitness function call. This way the fitness landscape is not defined just by the set of component functions but also by the transformation matrix A. For the case where A varies in time, a dynamic fitness landscape is obtained.

An example of such an approach is the Continuous Dynamic Problem Generator (CDPG) [27]. The generator originates from the idea of the XOR generator [36, 37] which was proposed for binary encoded problems. The idea of application of a binary mask to the binary representation of a solution was transferred to the CDPG approach. In CDPG the transformation matrix is applied to the real-valued vector of solution coordinates in the search space. There are two versions of CDPG and in both of them the solutions are rotated in the search space. In the first one the selected coordinates are just multiplied by -1. In the second one a rotation procedure is implemented. The procedure is based on the simple idea of rotation in the multidimensional search space by the plane defined just for two dimensions [25].

For the rotation of a solution $\mathbf{x}$ by the d_1,d_2-th plane, $\mathbf{x}$ is multiplied by the rotation matrix $R_{d_1,d_2}(\theta)$ where θ is the angle of rotation. An example matrix used for rotation by the plane $[3,5]$ in six-dimensional space looks like:

$$R_{3,5} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{14.19}$$

The construction of the matrix A starts with the definition of the list of rotation planes $\mathbf{r} = [r_1, r_2, ..., r_{N_{\mathrm{rot}}}]$ and the respective angles of rotation. The vector $\mathbf{r}$ contains a list of $N_{\mathrm{rot}}/2$ couples of dimension numbers where each couple represents a rotation plane: $[1,2]$, $[3,4]$, $[5,6]$, $\ldots$ $[N_{\mathrm{rot}}-1, N_{\mathrm{rot}}]$. Then the matrix A is obtained as the result of multiplication of rotation matrices:

$$A = R_{\mathbf{r}[1],\mathbf{r}[2]}(\theta_{1,2}) \cdot R_{\mathbf{r}[3],\mathbf{r}[4]}(\theta_{3,4}) \cdots R_{\mathbf{r}[N_{\mathrm{rot}}-1],\mathbf{r}[N_{\mathrm{rot}}]}(\theta_{N_{\mathrm{rot}}-1,N_{\mathrm{rot}}}). \tag{14.20}$$

The same rotation procedure can also be found in the dynamic rotation peak benchmark generator (DRPBG) [15].

14.3.4 Generalized Approach

The generalized model of a dynamic environment $\mathscr{E}$ is represented as a quadruple:

$$\mathscr{E} = (\mathscr{D}, \mathrm{I}, \mathrm{T}, \mathrm{S}) \tag{14.21}$$

where:

$\mathscr{D}$ is the search space,
I is a set of instances of component functions defined for different, not necessary disjunctive parts of $\mathscr{D}$,
T is a set of transformation mechanisms applied to the component function instances from I,
S is the definition of the transformation mechanism dynamics that controls the strength of changes applied to the parameters of the component function instances.

In the model every landscape consists of a number of components. The component properties are controlled by parameters. For each of the components defined two groups of parameters are defined: the parameters embedded in the component function formula and the parameters “surrounding” the formula. A modification of the parameters values is implemented by the component transformation mechanism. The mechanisms are assigned to respective parameters or sets of parameters in the component function formula. For example, when we want to stretch the landscape over the search space each of the solution coordinates is multiplied by a scaling factor. For a non–uniform stretching we need to use a vector of factors containing individual values for each of the coordinates. We call this type of modification a horizontal scaling and the factors belong to the first group of the component parameters. Another example is a vertical scaling where only the fitness value of a solution is multiplied by a scaling factor. This factor represents the second group of parameters. The first group controls horizontal translation, horizontal scaling, and rotation. For simplicity they are called *horizontal changes* in the further text. The second group of parameters controls vertical scaling and vertical translation (called respectively *vertical changes*). All of the changes can be obtained by dynamic modification of values of respective parameters during the process of search.

Eventually, the structure of the generated test-case is a logical consequence of the following assumptions:

(1) the fitness landscape consists of a number of different component landscapes,
(2) the dynamics of each of the components can be individually controlled,
(3) a component can be defined for a part or the whole of the search space, thus, in the case of a solution covered by more than one component the value of this solution can be the minimum, the maximum or the sum of values returned by the covering components.

Such a view of the definition of a dynamic environment can be found in the generalized dynamic benchmark generator (GDBP) [15]. The GDBP consists of two benchmark generators: The dynamic rotation peak benchmark generator already mentioned above, and dynamic composition benchmark generator (DCBG). In DCBG one can apply five types of component changes: horizontal translation, scaling and rotation and vertical translation and scaling.

In GDBP the evaluation of a solution **x** is defined as follows:

$$F(\mathbf{x},\phi,t)=\sum_{i=1}^{N}(w_i\cdot(f_i'((\mathbf{x}-\mathbf{O}_i(t)+\mathbf{O}_{i\mathrm{old}})/\lambda_i\cdot\mathbf{R}_i)+H_i(t))) \tag{14.22}$$

where:
$f_i'(\mathbf{x})=C\cdot f_i(\mathbf{x})/|f_i^*|$ is the normalized component function $f_i(\mathbf{x})$, where C equals 2000, and f_i^* denotes the value of the i-th component optimum point.
ϕ are the landscape parameters; $\phi=(\mathbf{O},\mathbf{R},\mathbf{H})$, where
$\mathbf{O}_i(t)$ is the optimum point coordinates for the i-th component and for the time t,
$\mathbf{O}_{i\mathrm{old}}$ is the initial optimum point coordinates for the i-th component; it is assumed that $O_{i\mathrm{old}}$ equals 0 for all the component functions,
$\mathbf{R}_i$ is the rotation matrix for the i-th component; every component has its own matrix

which is defined once at the beginning of the experiment and remains unchanged, H_i is the i-th value in the vector $\mathbf{H}$; the value represents a height coefficient for the i-th component.

The following types of the component functions can be found in the GDBP: Sphere, Rastrigin, Weierstrass, Griewank and Ackley (presented in Table 14.1).

Table 14.1 Component functions from the GDBP benchmark generator

name	function	domain
Sphere	$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	[-100,100]
Rastrigin	$f(\mathbf{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[-5,5]
Weierstrass	$f(\mathbf{x}) = \sum\limits_{i=1}^{n} (\sum\limits_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))]) - n \sum\limits_{k=0}^{k_{\max}} [a^k \cos(\pi b^k)]$	
	$a = 0.5, b = 3, k_{\max} = 20$	[-0.5,0.5]
Griewank	$f(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^{n} (x_i)^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-100,100]
Ackley	$f(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{n} \sum\limits_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n} \sum\limits_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	[-32,32]

Recently, yet another benchmark generator, `Syringa` [31], was proposed aiming to be a sum of a number of existing generators. Its architecture and source code is in accordance to the model $\mathscr{E}$ where the landscape consists of a number of simple, individually controlled components. In particular `Syringa` is able to simulate behavior of two generators, that is, MPB and GDBG, and to create completely new instances of the dynamic problem as well. The landscape created with `Syringa` consists of a number of components of any type controlled by a number of parameters. Each of the components covers a subspace of the search space. The final landscape is the result of a union of a collection of components such that each of the solutions from the search space is covered by at least one component. In the case when the solution belongs to the intersection of a few subspaces the solution value equals (i) the minimum (for minimization problems) or (ii) maximum (otherwise) or (iii) this can be also a sum of the fitness vales obtained from these components.

The version of `Syringa` presented in [31] consists of six types of component functions: peak, cone, sphere, Rastrigin, Griewank and Ackley. Four different characteristics of variability, originating from MPB and GDBG, are defined and can be applied to the component parameters: small step change (Equation (14.23)), large step change (Equation (14.24)), and two versions of random changes: (Equation (14.25) and Equation (14.26)). The change Δ of a parameter value can be calculated as follows:

$$\mathit{\Delta} = \alpha \cdot \mathrm{rand} \cdot (\max - \min), \tag{14.23}$$

$$\text{where } \alpha = 0.04, \mathrm{rand} = U(0,1),$$

$$\mathit{\Delta} = (\alpha \cdot sign(\mathrm{rand}_1) + (\alpha_{\max} - \alpha) \cdot \mathrm{rand}_2) \cdot (\mathscr{D}_\mathrm{t} - \mathscr{D}_\mathrm{t}), \tag{14.24}$$

$$\text{where } \alpha = 0.04, \alpha_{\max} = 0.1, \mathrm{rand}_{1,2} = U(0,1)$$

$$\mathit{\Delta} = N(0,1) \tag{14.25}$$

$$\mathit{\Delta} = U(r_{\min}, r_{\max}) \tag{14.26}$$

In the above-mentioned equations $\mathscr{D}_\mathrm{t}$ and $\mathscr{D}_\mathrm{b}$ represent the upper and lower boundaries of the search space, respectively, $N(0,1)$ is a random value obtained with standardized normal distribution, $U(0,1)$ is a random value obtained with uniform distribution form the range $[0,1]$, and $[r_{\min}, r_{\max}]$ defines the feasible range of $\mathit{\Delta}$ values.

14.4 Prediction of the Algorithm Answer

In dynamic optimization the raw outcome of the algorithm activity is always a series of values regardless of whether the measured value is the quality of the current best-found solution, the error level or the Euclidean distance to the optimum. Thus, the forms of the algorithm outcome presentation discussed in the previous section belong to one of the two types: (1) graphs with series of values or (2) single statistic values, for example the averaged best values obtained for each of the time intervals between subsequent changes in the fitness landscape, or averages of a series of values measured in subsequent time steps. The first form of the algorithm outcome presentation was applied mostly in the early publications devoted to dynamic optimization and quickly became replaced by the second form, that is, statistic values which are much more useful for comparisons. The second form has many versions which differ in the type of measured value and the frequency of measurements. What is more important, for the second form it is always taken for granted that the user is ready to wait for the result for the full time interval between one change in the landscape and the next one. However this does not have to be the case in real world situations. The fact is that in the real world there are usually two unknown parameters we have to cope with: the time we have to the next change in the environment and the time which the user is going to wait for the recovery of the result after the last change. We assume that the former parameter is greater than the latter one.

Therefore, in spite of the full access to the entire series of values measured during simulations for each of the time intervals between subsequent changes we need to take into account the stochastic nature of both time interval length and the user impatience. Particularly, we have to assume that the user can break the search process after each time step and request the solution. The user decision concerning selection of a time step for the break may be related with a number of circumstances, for example, with a quality of the algorithm current outcome. However, here we assume that the user motivation to stop the search process comes solely from outside. For example in the case of an airport where flights are disrupted due to thick fog we can use the optimization algorithm to find the optimal sequence for departures of delayed planes when the fog lifts. In this case the optimization can be performed as long as possible because the more computation is performed the better solution can be obtained. The only stopping criterion is the fog lifting which is the moment when the current best found solution has to be provided to the airport control center.

14.4.1 Evaluation of the Algorithm's Outcome for Random Stopping Time

The solution request in the m-th time step, that is, a stopping time can be represented as a random variable $\tau(m) = m$. It can take numerical values from the interval $[1,\ldots,M]$, where M represents the maximum number of time steps between subsequent changes in the fitness landscape.

The physical implementation of the random number generator for τ can be represented as an array of M independent continuous random number generators which generate uniformly distributed real numbers from the interval $[0,1]$. The user's single decision to request the solution at m-th time step depends on the generator in the m-cell of the array and a threshold assigned to this generator. The m-th threshold represents the probability of the user request at the m-th time step. The value generated by the generator is compared with the threshold. When the value is above the threshold the user is ready to wait and the optimization process continues, otherwise the user requests the solution. Clearly, the higher the threshold the stronger the impatience of the user.

The chance that the user requests the solution precisely just in the m-th time step from the last change can be calculated as follows: Let p_{true} represent the probability that the generator in the m-cell of the array returned a value below the threshold, that is, the user requests the solution. Then, for example, the case where the user requested the solution just after the third time step represents the following situation: the generator in the first cell returned a value above the threshold, the generator in the second cell also returned a value above the threshold but the generator in the third cell returned a value below the threshold. Thus, the expected value of the algorithm outcome in the m-th time step equals:

$$E[\Delta|\tau = m] = E\Delta(m). \tag{14.27}$$

In our analysis it is assumed that the algorithm outcome for the given optimization problem is already known, that is, it is deterministic. For example this is the case when experiments have already been done and a dataset which contains the algorithm's entire outcome is saved. Lets have a series of outcome values for subsequent time steps averaged over the number of runs for the same optimization problem, the same algorithm configuration, and the same starting conditions. Now we want to evaluate an expected value of the outcome for the case where there exist randomness originating from the user as described above. In this model the choice of the breaking moment has nothing in common with the quality of the current solution $\Delta(m)$ and depends on external conditions only. Thus, the expected value Δ of the outcome in the m-th time step simply equals

$$E[\Delta|\tau = m] = \Delta(m). \tag{14.28}$$

Of course, due to the fact that one of the time steps from $[1,\dots,M]$ must be chosen by the user (just like in the case of rolling 6-sided die where one of the numbers always must be rolled), the chances for selecting any of the time steps from $[1,\dots,M]$ equal 1:

$$p_m = \begin{cases} (1-p_{\text{true}})^{m-1} \cdot p_{\text{true}} & \text{if } m < M \\ 1 - \sum_{m=1}^{N-1} \left[(1-p_{\text{true}})^{m-1} \cdot p_{\text{true}}\right] & \text{if } m = M \end{cases} \tag{14.29}$$

where p_m represents probability $P(\tau = m)$. In Figure 14.4 a graph with p_m for $p_{\text{true}} \in \{0.01, 0.03, 0.06, 0.09\}$ is depicted. One can see that p_m decreases in every case and the higher value of p_{true}, the steeper incline of the line.

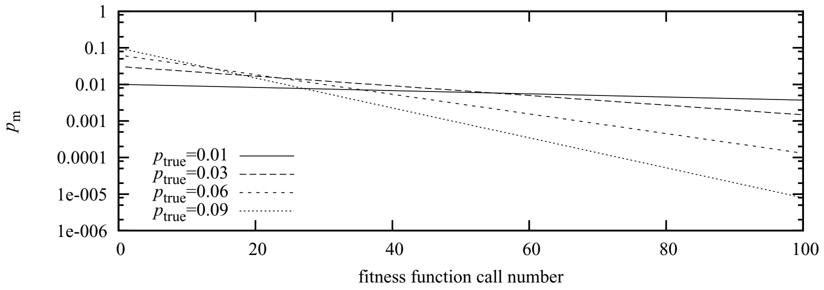


Fig. 14.4 The subsequent values of p_m for $M = 100$ and for $p_{\text{true}} \in \{0.01, 0.03, 0.06, 0.09\}$

Then, the expected value Δ of the outcome for a single time interval between subsequent changes in the environment equals:

$$E[\Delta] = \sum_{m=1}^{M} \Delta(m) \cdot p_m \tag{14.30}$$

where $\Delta(m)$ is the value from the outcome series of data for the m-th time step.

14.4.2 Evaluation of the Algorithm's Outcome with Respect to the User Preferences

It is realistic to assume that thresholds for the subsequent random number generators in the array are not the same, that is, the user is ready to wait for the solution for some time. However, the probability of his request changes in subsequent time steps. Example threshold functions given in Table 14.2 are depicted in Figure 14.5. For each of them, p_{true}^{m} represents a different probability of the solution request by the user in the m-th step. In the graphs it is assumed that there are 50 time steps between subsequent changes. This group of functions can be divided into two subgroups which represent: decreasing impatience of a user (Figure 14.5, the top row) and increasing impatience ((Figure 14.5, the bottom row). Both types of impatience are expressed by the threshold which changes linearly (the first column), with acceleration (the second column), or with delay (the third column).

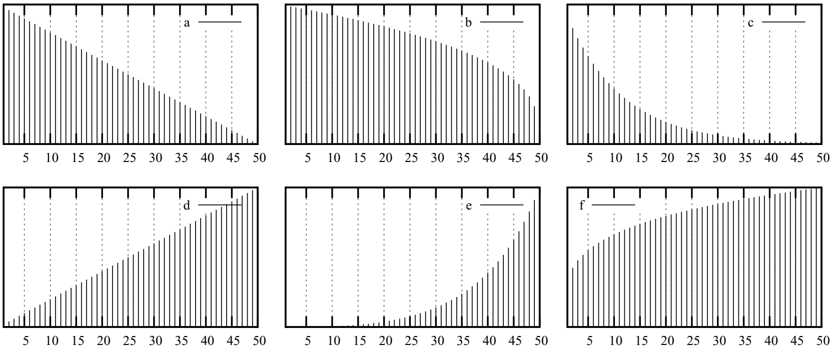


Fig. 14.5 Graphs of of subsequent values of p^m_{true} for six example threshold functions which represent: decreasing (the top row) and increasing impatience of a user (the bottom row); $N = 50$

Table 14.2 Example threshold functions which represent: decreasing ((a), (b) and (c)) and increasing ((d), (e) and (f)) impatience of a user in Figure 14.5

(a)	$f_a(m) = 1 - m/M$	(b)	$f_b(m) = \sqrt[3]{1 - m/M}$
(c)	$f_c(m) = (1/1000)^{m/M}$	(d)	$f_d(m) = m/M$
(e)	$f_e(m) = \frac{\exp((m/M)^3) - \exp(0)}{\exp(1) - \exp(0)}$	(f)	$f_f(m) = \frac{\ln(\sqrt[3]{(m/M)} + 1) - \ln(1)}{\ln(2) - \ln(1)}$

For the case of varying threshold values the formula for the expected value Δ of the algorithm outcome remains the same as for the case with equal thresholds of the generators (Equation (14.30)), however, p_m differs:

$$p_m = \begin{cases} \left[\prod_{k=1}^{m-1}(1 - p^k_{\text{true}})\right] \cdot p^m_{\text{true}} & \text{if } m < M \\ 1 - \sum_{m=1}^{M-1} \left[\prod_{k=1}^{m-1}(1 - p^k_{\text{true}})\right] \cdot p^m_{\text{true}} & \text{if } m = M \end{cases} \qquad (14.31)$$

For each of the user impatience characteristics depicted in Figure 14.5 respective graphs with subsequent values of p_m for $M = 50$ and for $M = 5000$ are shown in Figure 14.6. It is worth noting that for different values of M the graphs for the same threshold functions differ significantly. Particularly, for $M = 5000$ subsequent values of p_m obtained for threshold functions (a), (b) and (c) decrease so rapidly that the ranges for X-axis in the graphs are trimmed just to first 10 values in the series.

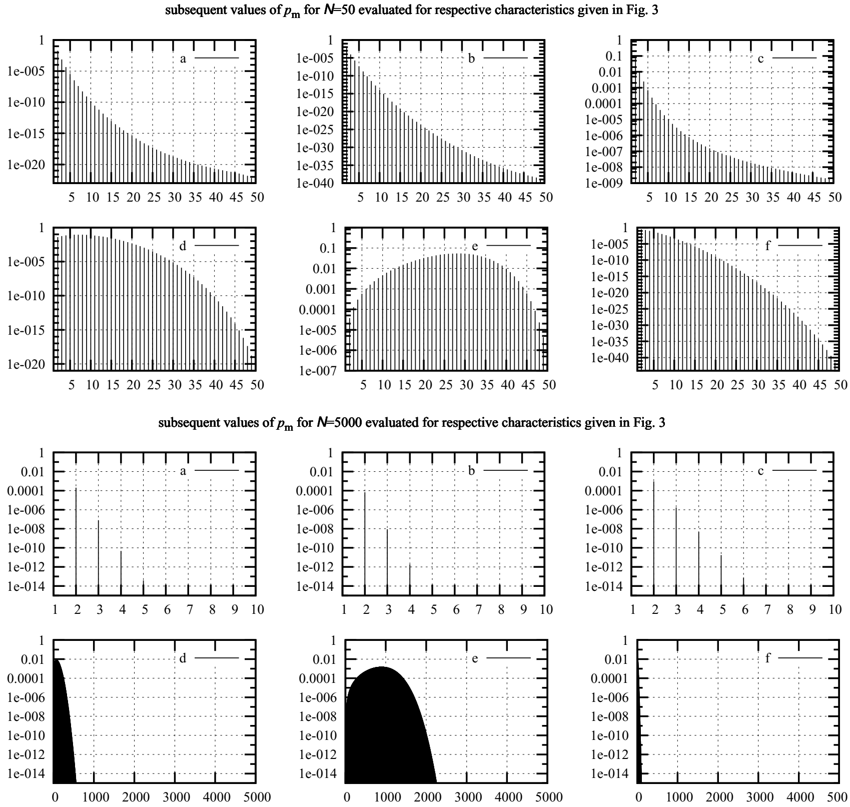


Fig. 14.6 Graphs with subsequent values of p_m for $N = 50$ and for $N = 5000$ evaluated for respective characteristics given in Figure 14.5

The graphs in Figure 14.6 were generated for p^m_{true} varying within the range $[0,1]$. One can see that in every case there are a few highest values of p_m which are at least several orders of magnitude higher than the remaining values of p_m present in the sequence. This makes the threshold functions rather useless due to the fact that the expected value of the algorithm outcome $E[\Delta]$ evaluated according to Equation (14.30) shall be dominated by few outcome values in the series, that is, just the ones which are multiplied by the few highest values of p_m. Besides, one can observe in the graphs that for longer series the fraction of highest values of p_m decreases, that is, the peaks in the graphs are broader for $M = 50$ than for $M = 500$.

The graph of p_m generated for threshold function (e) is the exception. In this case quite a large fraction of values in the series (for $M = 50$ this is about a half) belongs to the group of the largest and they differ from each other by no more than one order of magnitude. The threshold function (e) has the following formula:

$$f_e(m) = \frac{\exp((m/M)^q) - \exp(0)}{\exp(1) - \exp(0)} \tag{14.32}$$

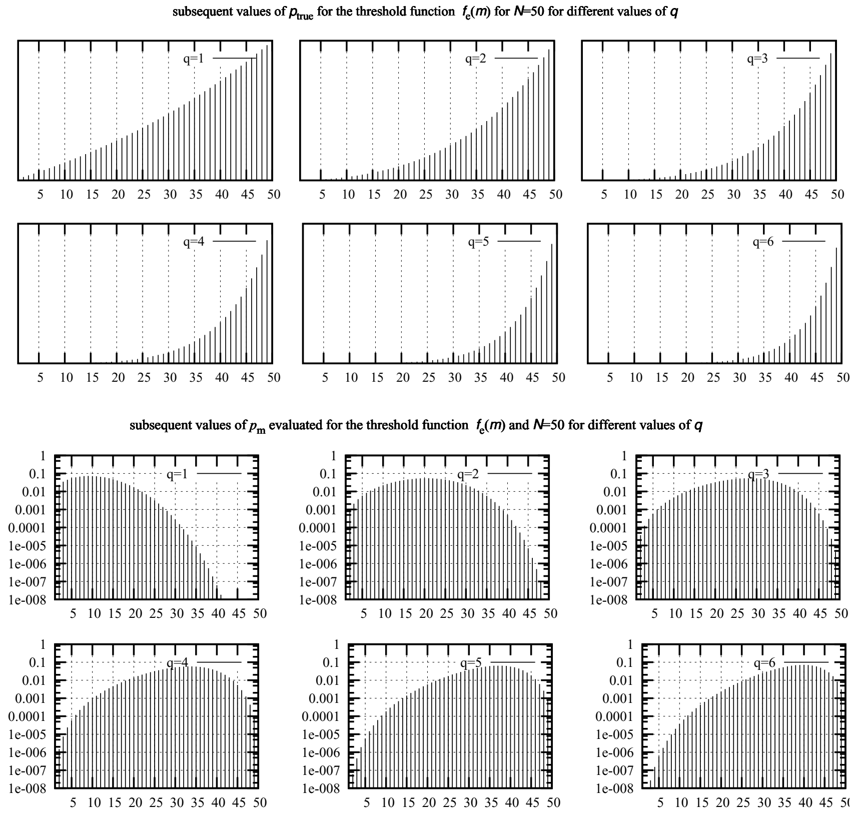


Fig. 14.7 Graphs of of subsequent values of p^m_{true} for the threshold function $f_e(m)$ (Equation 14.32) and respective graphs of subsequent values of p_m for $m \in [1,50]$ and for $q \in [1,6]$

where q is the parameter which controls the strength of acceleration. In the graph presented in Figure 14.5 the value of q equals 3. Graphs for threshold functions obtained for $f_e(m)$ where $q \in [1,6]$ and $m \in [1,50]$ as well as respective graphs with the subsequent values of p_m are presented in Figure 14.7.

The graphs in Figure 14.7 show that the parameter q controls the range of outcome values which contribute the most to the expected value of the algorithm outcome $E[\Delta]$. In Figure 14.7 for $q = 1$ this is the range $m \in [1,21]$, for $q = 2$, $m \in [7,33]$, for $q = 3$, $m \in [13,39]$, for $q = 4$, $m \in [19,43]$, for $q = 5$, $m \in [23,46]$, and for $q = 6$, $m \in [26,47]$. The ranges move to the right for subsequent values of q, in accordance with differences between graphs of threshold functions obtained for subsequent values of q.

14.4.3 Application in Practice

An example application of the theory presented in previous subsections is given below. Lets have a dataset which represent an optimization algorithm outcome generated for the time of first three changes in the environment. The values in the series represent the algorithm output generated after every subsequent evaluation of the fitness function, thus, in this case the single fitness function call represents a single time step. There were 5000 time steps between changes and three subsequent changes occurred, thus, the entire series consists of 15000 numbers. At every time step the user is allowed to stop the optimization process and request the output value. For the given dataset evaluations were performed according to five versions of the user characteristic: $f_e(m)$ where $q \in \{1,3,6,9,12\}$. The graph of the data as well as the graphs with subsequent results of multiplication p_m by respective values from the series are depicted in Figure 14.8.

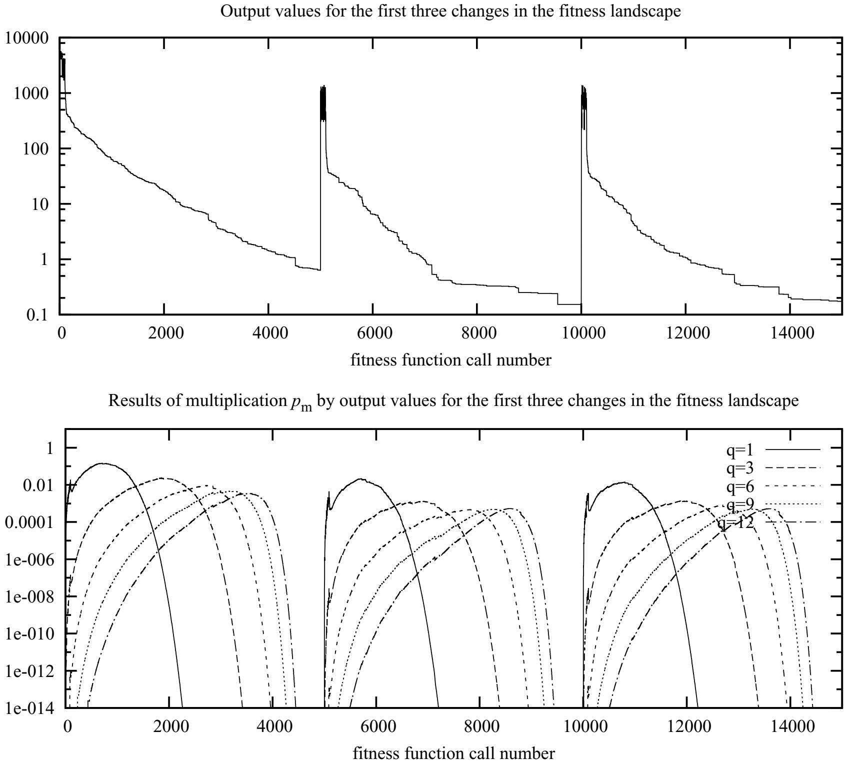


Fig. 14.8 Graphs with sample output data (top graph) and with subsequent results of multiplication p_m by respective values from the series (bottom graph)

The average values for each of the three subseries equal: 124.79 for the first shape of the landscape, 18.76 for the second shape, and 19.88 for the third shape. The least found values equal: 0.633 for the first shape, 0.153 for the second shape, and 0.174 for the third shape. Values of expected value of output value for the three subseries are presented in Table 14.3. One can see that depending on the value of q the expected value of output value decreases as q grows.

Table 14.3 Expected value for 3 subseries of output dataset evaluated for $q \in \{1,3,6,9,12\}$

shape nr	$q = 1$	$q = 3$	$q = 6$	$q = 9$	$q = 12$
0	92.8817	19.0969	7.33602	3.79088	2.47320
1	11.1796	1.1904	0.41183	0.34034	0.31844
2	8.0305	1.1970	0.59881	0.38653	0.32193

14.5 Summary

Methods of estimation of the heuristic algorithm performance are discussed in the chapter above. A number of existing methods of performance measurements is presented. Among them one can mention the value of the current best found solution or its error, mean values of the population attributes (like, for example, mean values of the worst solutions in subsequent iterations), or the ability to quickly recover the population quality after a change in the environment. The chapter also contains a review of existing dynamic benchmark generators. The generalized model $\mathscr{E}$ of the dynamic environment based on this review is also proposed. The model $\mathscr{E}$ is a quadruple: (1) the search space, (2) a set of component functions, (3) a set of transformation mechanisms and (4) a set of definitions of the mechanism dynamics.

The model covers all existing benchmarks and allows for building completely new ones. Besides, it is also important to take into account probability of the search process break caused by the user request for the result. The request can appear much earlier than just before the next change.

The main conclusion arising from the review of the performance measures concerns the meaning of the phrase “the algorithm performance” which varies depending on two issues: the selected measure and the nondeterministic features of the environment as well as the nondeterministic preferences of a user. The comment on the first issue is simple: every measure of the algorithm performance estimates a different feature of the algorithm activity during the process of search. The second issue concerns the presence of uncertainty which occurs when we do not know how much time we have to the next change in the environment and how much time the user is going to wait for the recovery of the result after the last change. Particularly in the latter case the expected value of the algorithm outcome should rather be applied for comparisons of the performance as it has been proposed in Section 14.4.

References

[1] Alba, E., Sarasola, B.: Measuring fitness degradation in dynamic optimization problems. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 572–581. Springer, Heidelberg (2010)

[2] Ayvaz, D., Topcuoglu, H.R., Gürgen, F.S.: Performance evaluation of evolutionary heuristics in dynamic environments. Appl. Intell. 37(1), 130–144 (2012)

[3] Bäck, T.: On the behaviour of evolutionary algorithms in dynamic environments. In: Proc. of the Fifth IEEE Conf. on Evolutionary Computation, pp. 446–451. IEEE Press (1998)

[4] Bäck, T., Schütz, M.: Intelligent mutation rate control in canonical genetic algorithm. In: Michalewicz, M., Raś, Z.W. (eds.) ISMIS 1996. LNCS, vol. 1079, pp. 158–167. Springer, Heidelberg (1996)

[5] Branke, J.: Memory enhanced evolutionary algorithm for changing optimization problems. In: Proc. of the Congr. on Evolutionary Computation, vol. 3, pp. 1875–1882. IEEE Press (1999)

[6] Cedeño, W., Vemuri, V.R.: On the use of niching for dynamic landscapes. In: Proc. of the 1997 IEEE Int. Conf. on Evolutionary Computation, pp. 361–366. IEEE Press (1997)

[7] Cobb, H.G., Grefenstette, J.F.: Genetic algorithms for tracking changing environments. In: Proc. of the Fifth Int. Conf. on Genetic Algorithms (ICGA 1993), pp. 523–530. Morgan Kaufmann Publishers (1993)

[8] Dasgupta, D., McGregor, D.R.: Nonstationary function optimization using the structured genetic algorithm. In: Parallel Problem Solving from Nature 2, PPSN-II, pp. 147–156. Elsevier (1992)

[9] Gallagher, M., Yuan, B.: A general-purpose tunable landscape generator. IEEE Trans. Evol. Comput. 10(5), 590–603 (2006)

[10] Goldberg, D.E., Smith, R.E.: Non-stationary function optimisation using genetic algorithms with dominance and diploidy. In: Proc of the 2nd Int. Conf. on Genetic Algorithms and Their Applications, pp. 59–68. Lawrence Erlbaum Associates (1987)

[11] Grefenstette, J.J.: Genetic algorithms for changing environments. In: Parallel Problem Solving from Nature 2, PPSN-II, pp. 139–146. Elsevier (1992)

[12] Jin, Y., Branke, J.: Evolutionary algorithms in uncertain environments – a survey. IEEE Trans. Evol. Comput. 9(3), 303–317 (2005)

[13] Jin, Y., Sendhoff, B.: Constructing dynamic optimization test problems using the multi-objective optimization concept. In: Raidl, G.R., et al. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 525–536. Springer, Heidelberg (2004)

[14] Jones, T.: Evolutionary algorithms, fitness landscapes and search. Ph.D. thesis, University of New Mexico (1995)

[15] Li, C., Yang, S.: A generalized approach to construct benchmark problems for dynamic optimization. In: Li, X., et al. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 391–400. Springer, Heidelberg (2008)

[16] Mori, N., Kita, H., Nishikawa, Y.: Adaptation to a changing environment by means of the thermodynamical genetic algorithm. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 513–522. Springer, Heidelberg (1996)

[17] Mori, N., Kita, H., Nishikawa, Y.: Adaptation to a changing environment by means of the feedback thermodynamical genetic algorithm. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 149–158. Springer, Heidelberg (1998)

[18] Morrison, R.W.: Performance measurement in dynamic environments. In: GECCO 2003: Proc. of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conf., pp. 99–102. AAAI (2003)

[19] Morrison, R.W., De Jong, K.A.: A test problem generator for non-stationary environments. In: Proc. Congress on Evolutionary Computation, pp. 1859–1866. IEEE Press (1999)

[20] Ng, K.P., Wong, K.C.: A new diploid scheme and dominance change mechanism for non-stationary function optimization. In: Proc. of the Sixth Int. Conf. on Genetic Algorithms, pp. 159–166. Morgan Kaufmann, San Francisco (1995)

[21] Richter, H.: Change detection in dynamic fitness landscapes: An immunological approach. In: World Congress on Nature and Biologically Inspired Computing, pp. 719–724. IEEE (2009)

[22] Richter, H.: Detecting change in dynamic fitness landscapes. In: IEEE Congress on Evolutionary Computation, pp. 1613–1620. IEEE (2009)

[23] Richter, H.: Evolutionary optimization and dynamic fitness landscapes; from reaction–diffusion systems to chaotic CML. In: Zelinka, I., Celikovsky, S., Richter, H., Chen, G. (eds.) Evolutionary Algorithms and Chaotic Systems. SCI, vol. 267, pp. 409–446. Springer, Heidelberg (2010)

[24] Richter, H., Dietel, F.: Change detection in dynamic fitness landscapes with time-dependent constraints. In: World Congress on Nature and Biologically Inspired Computing, pp. 580–585. IEEE (2010)

[25] Salomon, R.: Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions. BioSystems 39(3), 263–278 (1996)

[26] Salomon, R., Eggenberger, P.: Adaptation on the evolutionary time scale: A working hypothesis and basic experiments. In: Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (eds.) AE 1997. LNCS, vol. 1363, pp. 251–262. Springer, Heidelberg (1998)

[27] Tinós, R., Yang, S.: Continuous dynamic problem generators for evolutionary algorithms. In: The Congress on Evolutionary Computation, pp. 236–243. IEEE (2007)

[28] Trojanowski, K., Michalewicz, Z.: Evolutionary approach to non-stationary optimisation tasks. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS, vol. 1609, Springer, Heidelberg (1999)

[29] Trojanowski, K., Michalewicz, Z.: Searching for optima in non-stationary environments. In: Proc. of the Congress on Evolutionary Computation, vol. 3, pp. 1843–1850. IEEE Press (1999)

[30] Trojanowski, K., Obuchowicz, A.: Measures for non-stationary optimization tasks. In: 5th ICANNGA: Artificial Neural Nets and Genetic Algorithms, pp. 244–247. Springer (2001)

[31] Trojanowski, K., Raciborski, M., Kaczyński, P.: Self-adaptive differential evolution with hybrid rules of perturbation for dynamic optimization. Journal of Telecommunications and Information Technology 4, 18–28 (2011)

[32] Vavak, F., Fogarty, T.C.: A comparative study of steady state and generational genetic algorithms for use in nonstationary environments. In: Fogarty, T.C. (ed.) AISB-WS 1996. LNCS, vol. 1143, pp. 297–304. Springer, Heidelberg (1996)

[33] Vavak, F., Fogarty, T.C.: Comparison of steady state and generational genetic algorithms for use in nonstationary environments. In: Int. Conf. on Evolutionary Computation, pp. 192–195. IEEE Press (1996)

[34] Weicker, K.: Performance measures for dynamic environments. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 64–76. Springer, Heidelberg (2002)

[35] Weicker, K., Weicker, N.: On evolution strategy optimization in dynamic environments. In: Proc. of the Congress on Evolutionary Computation, vol. 3, pp. 2039–2046. IEEE Press (1999)

[36] Yang, S.: Non-stationary problem optimization using the primal-dual genetic algorithm. In: The Congress on Evolutionary Computation, vol. 3, pp. 2246–2253. IEEE (2003)

[37] Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft Comput. 9(11), 815–834 (2005)

[38] Yuan, B., Gallagher, M.: On building a principled framework for evaluating and testing evolutionary algorithms: a continuous landscape generator. In: IEEE Congress on Evolutionary Computation, pp. 451–458. IEEE (2003)

Part IV
Visualization and Characterization

Chapter 15
Fractal Analysis of Fitness Landscapes

Ivan Zelinka, Oldrich Zmeskal, and Petr Saloun

Abstract. Complex optimization problems may have fitness landscapes with fractal characteristics. This chapter reviews landscapes obtained from basic artificial test functions as well as cost functions of real application problems which have the property to be fractal. We will discuss the description, structure and complexity of these fractal fitness landscapes. A major topic of this chapter is to use elements from fractal geometry to measure attributes of fractal landscapes. Also, structural as well as functional properties of the landscape are discussed. The examples used in this chapter are two–dimensional, however it is possible to extend the proposed analysis to *n* dimensions.

15.1 Cost Function Landscapes

In engineering design we are generally facing complex optimization problems whose solutions can be described as a space of possible solutions that can be visualized as a surface. This surface can be interpreted as a fitness landscape. Frequently, irregularities as well as high levels of complexity can be observed in such fitness landscapes. For example, for functions with a high degree of nonlinearity, various types of complex geometrical structures can occur, see for instance Figure 15.1 (a), (c) and (e). A further level of complex geometrical structure is obtained if the landscape's surface has a fractal characteristics. Fractal means that the Hausdorff dimension of the geometrical object studied is not an integer. A fractal fitness

Ivan Zelinka · Petr Saloun
VSB-TUO, Faculty of Electrical Engineering and Computer Science, 17. listopadu 15,
708 33 Ostrava-Poruba, Czech Republic
e-mail: {ivan.zelinka,petr.saloun}@vsb.cz

Oldrich Zmeskal
Faculty of Chemistry, Brno University of Technology, Purkyňova 118,
612 00 Brno, Czech Republic
e-mail: zmeskal@fch.vutbr.cz

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_15, © Springer-Verlag Berlin Heidelberg 2014

landscape, therefore, is a landscape whose “fitness surface” has a real–valued dimension. In particular, if we consider two–dimensional projections of the “fitness surface”, its dimension must be larger than two to qualify as fractal.

While fitness landscapes have been the topic of intense research, see for instance [5, 18, 34] for some recent results, fractal landscapes received far less attention. The first to notice that fitness landscape may be fractal were Weinberger and Stadler [38]. These authors discuss fitness landscapes based on biological data, i.e. on biological and biochemical measurements, that can be understood as a fitness of a particular genome. They further showed that these data can be treated as a fitness landscape and that there is a connection between the landscape’s correlation and its self–similarity (fractalness). Fractal fitness landscapes have also been observed in application studies, for instance in evolutionary robotics [14] or dynamic optimization [27]. A more technically oriented research can be found in Ribeiro et al. [25]. In this paper a novel numerical procedure that can be easily implemented is considered to evaluate the complexity of two or higher-dimensional patterns including fractal landscapes artificially generated.

In this chapter, we will approach the topic of fractal fitness landscapes from a more general perspective. We will on the one hand show how fractal landscapes can be artificially constructed by applying standard test functions (also called benchmarks) used in evolutionary computation which are made fractal by combining them with functions with known self–similarity, for instance with a version of the Weierstrass–Mandelbrot function. On the other hand, we will consider cost functions that appear in experimental chaos control and synchronization. These cost functions give raise to cost function landscapes which also show fractal characteristics for selected experimental parameters. Both types of fractal landscapes, artificially constructed and experimentally obtained, are further investigated using a fractal image analysis.

15.2 Selected Test Functions and Their Fitness Landscapes

The landscapes used in this chapter are taken from widely–known artificial test functions and are listed in Table 15.1. For a graphical description of some of these functions, see Figure 15.1. Also, pseudo-fractal functions (based on the Weierstrass–Mandelbrot function) whose complexity can vary according to suitable parameter settings, are used (see Figures 15.9 and 15.10). Finally, we study selected cost function landscapes from experiments on deterministic chaos synchronization and control (see [29, 40], and also Figures 15.6 and 15.7).

The artificial test functions have been selected from the pool of commonly used benchmarks. The test functions which were selected for our experiments range from simple to very complex and irregular to demonstrate the performance and capability of fractal analysis.

Even more complex and closer to real world problems are fitness landscapes generated by our experiments on deterministic chaos control (for classical chaos control, see [31, 32, 40] and for CML system control, see [39]) and synchronization

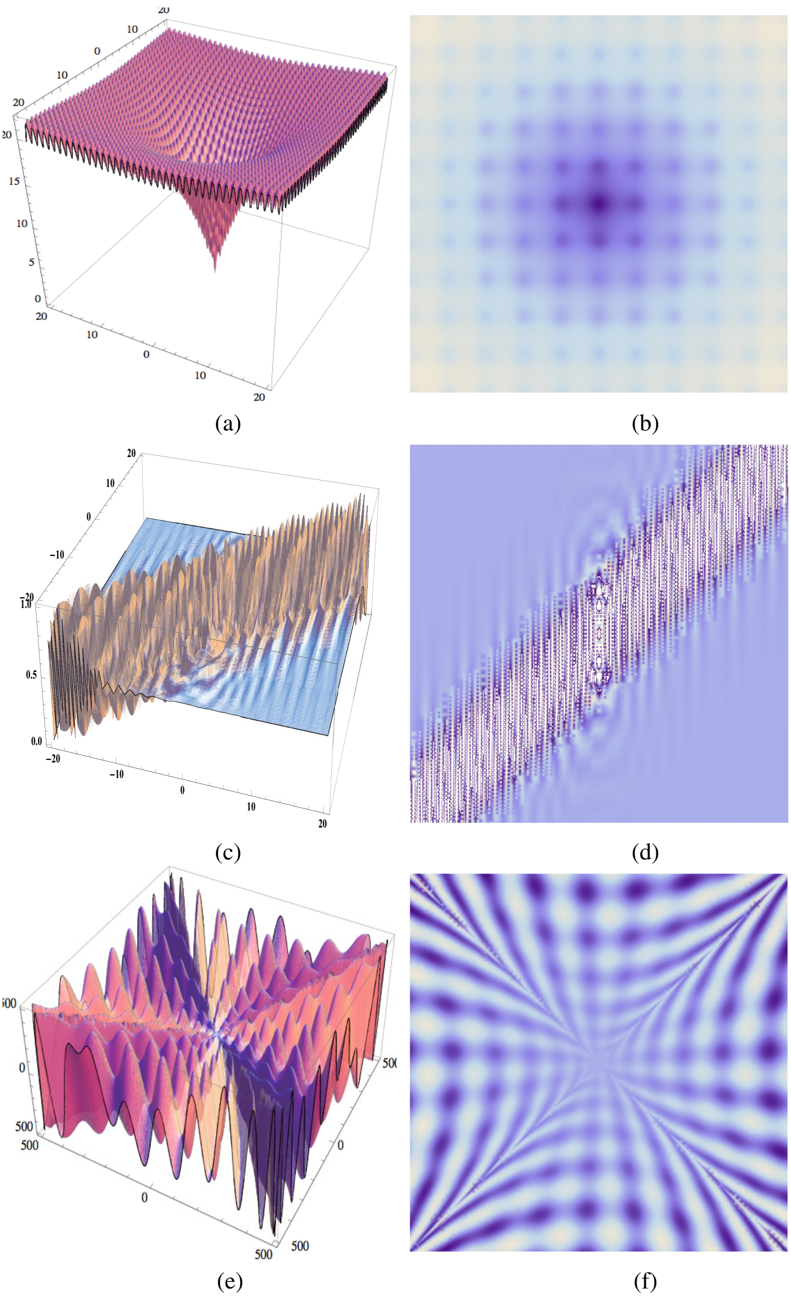


Fig. 15.1 3D fitness landscapes (a) (c) and (e) and their 2D projection (b), (d) and (f) as used for fractal analysis (by software HarFA) [43]: (a) Ackley function, (c) Pathological function, (e) Rana function

Table 15.1 Test functions used for constructing fractal fitness landscapes

#	Name	$f(x) =$
1	1st De Jong's function	$\sum_{i=1}^{n} x_i^2$
2	Rastrigin's function	$\sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i)$
3	Schwefel's function	$\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$
4	Griewangk's function	$1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$
5	Stretched V sine wave function	$\sum_{i=1}^{n-1} \left(\sqrt[4]{(x_i^2 + x_{i+1}^2)} \sin(50 \sqrt[10]{(x_i^2 + x_{i+1}^2)})^2 + 1\right)$
6	Ackley's function	$\sum_{i=1}^{n-1} \left(20 + e - \frac{20}{e^{0.2\sqrt{\frac{(x_i^2 + x_{i+1}^2)}{2}}}} - e^{0.5(\cos(2\pi x_i)+\cos(2\pi x_{i+1}))}\right)$
7	Egg Holder	$\sum_{i=1}^{n-1} \left(\begin{array}{l} -x_i \sin(\sqrt{\lvert x_i - x_{i+1} - 47\rvert}) - \\ (x_{i+1} + 47) \sin(\sqrt{\lvert x_{i+1} + 47 + \frac{x_i}{2}\rvert}) \end{array}\right)$
8	Rana's function	$\sum_{i=1}^{n-1} \left(\begin{array}{l} x_i \sin(\sqrt{\lvert x_{i+1} + 1 - x_i\rvert}) \cos(\sqrt{\lvert x_{i+1} + 1 + x_i\rvert}) + \\ (x_{i+1} + 1) \cos(\sqrt{\lvert x_{i+1} + 1 - x_i\rvert}) \sin(\sqrt{\lvert x_{i+1} + 1 + x_i\rvert}) \end{array}\right)$
9	Pathological test function	$\sum_{i=1}^{n-1} \left(0.5 + \frac{\sin(\sqrt{100x_i^2 - x_{i+1}^2})^2 - 0.5}{(1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)^2)}\right)$

(see [39]: Chapter 12). Both kinds of problems (i.e. control and synchronization) are defined as optimization on predefined cost functions. In Zelinka et al. [39] several case studies of deterministic chaos control and synchronization are described.

In general, synchronization is a dynamical process during which one system (synchronized, slaved) is remote from but connected to another (synchronizing, master) so that the synchronized system in a certain manner follows the behavior of the master system (see Figure 15.2). Synchronization can be divided into the following classes [7, 21, 29]:

- **Identical synchronization.** This synchronization may occur when two identical chaotic oscillators are mutually coupled (unidirectional or bidirectional coupling), or when one of them drives the other, which is the case of a numerical study reported in this chapter (the Lorenz - Lorenz synchronization as defined in Equation (15.1)). Basically, if $\{x_1, y_1, z_1\}$ is the set of dynamical state variables of the master system and $\{x_2, y_2, z_2\}$ is the set of the variables of the slave system, then both systems are synchronized if, under certain initial conditions and $t \to \infty$, it is true that $\lvert x_1 - x_2\rvert \to 0$.
- **Generalized synchronization.** This differs from the previous case by the fact that the coupled chaotic oscillators are different and that the dynamical state of one of the oscillators is completely determined by the state of the other. This is the case of another numerical study in this chapter (the Rössler - Lorenz synchronization given in Equation (15.3) [39]).
- **Phase synchronization.** This is another case of synchronization which occurs when the coupled oscillators are not totally identical and the amplitudes of the

oscillator remain unsynchronized, while only the oscillator phases evolve in synchrony. There is a geometrical interpretation of this case of synchronization. It is possible to find a plane in the phase space on which the projection of the trajectories of the oscillator follows a rotation around a well-defined center. The phase is defined by the angle $\varphi(t)$, described by the segment which is joining the center of the rotation and the projection of the trajectory point onto the plane.

- **Anticipated and lag synchronization.** Let us assume that we have a synchronizing system with state variables $\{x_1, y_1, z_1\}$ and a synchronized system with state variables $\{x_2, y_2, z_2\}$. Anticipated and lag synchronization occur if it is true that $x_1(t) = x_1(t+\tau)$. This relation, in fact, says, that the dynamics of one of the systems follows, or anticipates, the dynamics of the other and that this dynamics is described by delay differential equations.
- **Amplitude envelope synchronization.** This is a kind of synchronization which may appear between two weakly coupled chaotic oscillators. Compared with another cases of synchronization, there is no correlation between phases or amplitudes. One can observe a periodic envelope that has the same frequency in the two systems. The magnitude of that envelope has the same order than the difference between the average frequencies of the oscillations of both systems. It is important to note that phase synchronization can develop from amplitude envelope synchronization if the strength of the coupling force between two amplitude envelope synchronized oscillators increases in time.

A rich amount of literature about synchronization exist, see for example [7, 21, 29]. Other research works are on synchronization based on time series analysis [3, 35] and robustness of synchronized systems [28].

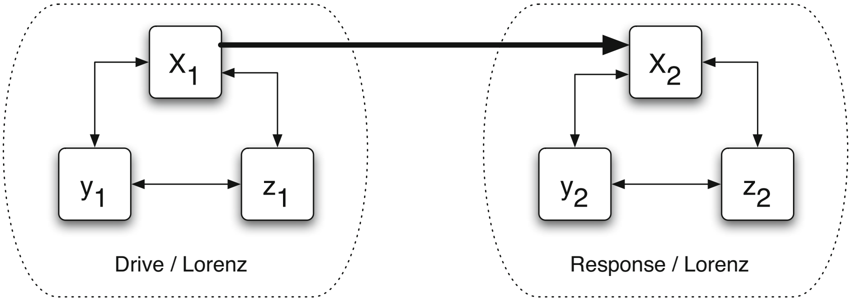


Fig. 15.2 The scheme of Lorenz - Lorenz synchronization

As an example of the numerical experiments we can discuss the so–called Lorenz - Lorenz system for synchronization:

$$\begin{array}{l}
\textbf{Lorenz}-\textbf{Lorenz synchronization}\\
\textbf{Lorenz system (master)}:\\
\dot{x}_1(t)=-a\,(x_1(t)-y_1(t))\\
\dot{y}_1(t)=-x_1(t)z_1(t)+bx_1(t)-y_1(t)\\
\dot{z}_1(t)=x_1(t)y_1(t)-cz_1(t)\\
\\
\textbf{Lorenz system (slave)}:\\
\dot{x}_2(t)=-a_2\,(x_2(t)-y_2(t))+d\,(x_1(t)-x_2(t))\\
\dot{y}_2(t)=-x_2(t)z_2(t)+bx_2(t)-y_1(t)\\
\dot{z}_2(t)=x_2(t)y_2(t)-z_2(t)\\
\\
a=10,\ b=28,\ c=2.66.
\end{array} \tag{15.1}$$

The synchronization has been done by coupling of the variables $x_1(t)$ and $x_2(t)$ via the parameter d [39]. The difference between the variables has been calculated and multiplied by the parameter d, see $(d\,(x_1(t)-x_2(t)))$ in Equation (15.1). In fact, a search for optimal parameter settings can be done depending on two parameters; i.e. d and a_2 in Equation (15.1).

The cost function of the Lorenz - Lorenz (LL) synchronization is

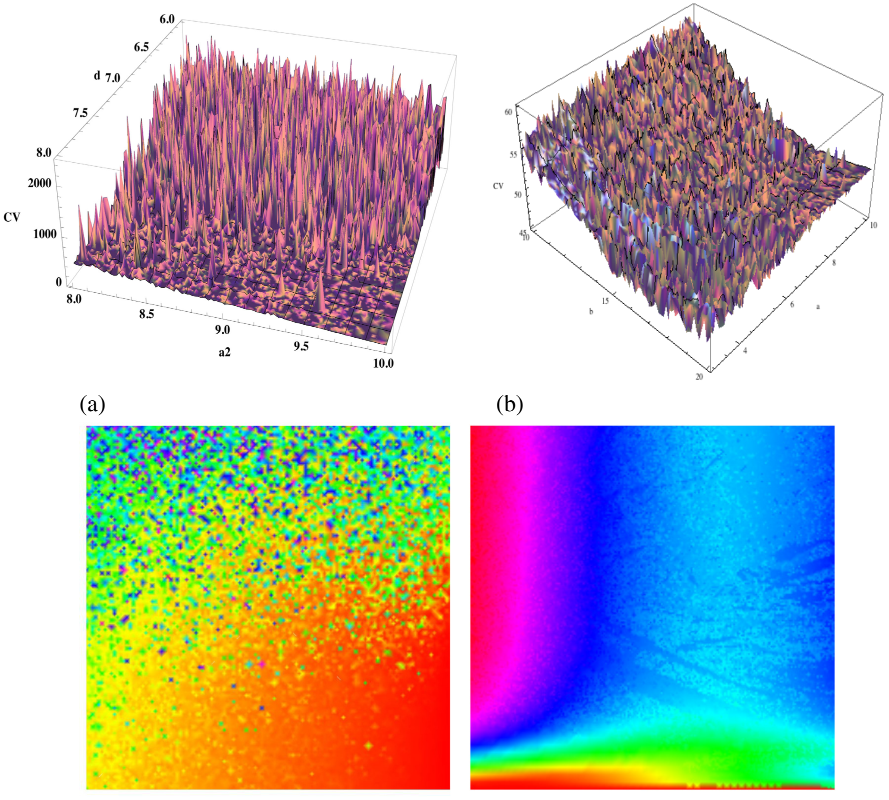
$$CV_{LL}(a_2,d)=\int\limits_0^{100}|x_1(t)-x_2(t)|+|y_1(t)-y_2(t)|+|z_1(t)-z_2(t)|\,\mathrm{d}t. \tag{15.2}$$

A second example is synchronization between different systems. We consider the case of a Rössler system connected with a Lorenz system

$$\begin{array}{l}
\textbf{Rössler}-\textbf{Lorenz synchronization}\\
\textbf{Rössler system (master)}:\\
\dot{x}_1(t)=-y_1(t)-z_1(t)\\
\dot{y}_1(t)=-x_1(t)-\frac{y_1(t)}{5}\\
\dot{z}_1(t)=(x_1(t)-5.7)\,z_1(t)+0.2\\
\\
\textbf{Lorenz system (slave)}:\\
\dot{x}_2(t)=-a\,(x_2(t)-y_2(t))\\
\dot{y}_2(t)=-x_2(t)z_2(t)+bx_2(t)+c\,(y_1(t)-y_2(t))\\
\dot{z}_2(t)=x_2(t)y_2(t)-z_2(t)\\
\\
c=69.737
\end{array} \tag{15.3}$$

with a cost function of the Rössler - Lorenz (RL) synchronization which depends on the parameters a and b in Equation (15.3):

$$CV_{RL}(a,b)=\int\limits_0^{200}|(y_1(t)-y_2(t)|\,dt. \tag{15.4}$$



(a)

(b)

(c)

(d)

Fig. 15.3 Fractal landscape of the synchronization cost function. (a) Cost function (15.2) generated by Lorenz - Lorenz synchronization. The cost function value depend on the coupling parameter d and the parameter a_2. The value of each point on Z (labeled CV) axis is in fact the gray difference from Figure 15.4. (b) Cost function (15.4) generated by Rössler - Lorenz synchronization depending on the parameters a and b. (c) 2D projection of the cost function in (a). (d) 2D projection of the cost function in (b).

The complexity and fractalness of the cost function landscape CV_{LL} (see also Equation (15.2)), depending on both parameters d and a_2 is depicted in Figure 15.3a. All parameters were varied around nominal values, as referred to in the literature. A visual inspection of the landscape reveals a highly irregular surface which suggest a fractal characteristic. In the fractal landscape a linear-like trend is also visible. Thanks to this trend it can be seen that the minimum can be expected at the position $\{d, a_2\} = \{8, 10\}$.

The cost function in Equation (15.2) has been calculated according to the distance between the desired synchronizing system state and the actual synchronized system state (see Figure 15.4). The cost function value is, in fact, the absolute value of the summarization of the gray areas between the synchronizing and synchronized system output (time series) as shown in Figure 15.4. The difference between the

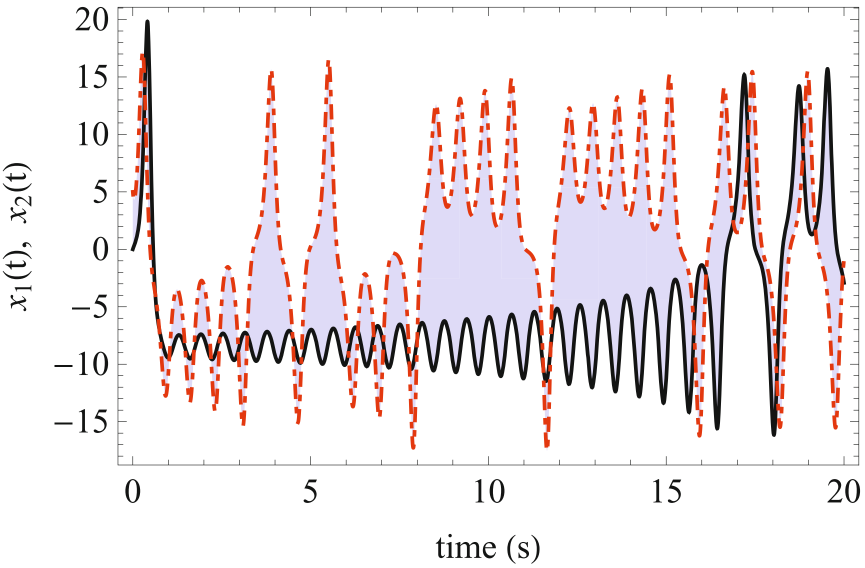
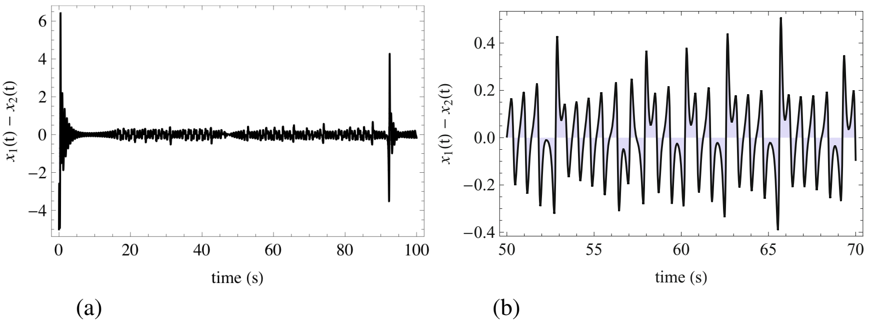


Fig. 15.4 Description of the cost function calculation in Equation (15.2). For all three variables $x_1(t)$, $y_1(t)$ and $z_1(t)$ we have calculated the difference between the behavior of the synchronizing and synchronized system (light gray area).



(a)

(b)

Fig. 15.5 Cost function in Equation (15.4). (a) Total time series (b) Amplification of the time interval $[50s, 70s]$.

worst and the best behavior of x_1 and x_2 for the cost function in Equation (15.4) is depicted in Figure 15.5. The biggest impact on the cost value comes from the interval $[0, 7s]$ before the systems are well synchronized. Another anomaly is at the position $92s$, which is just a sharp peak with little impact on the cost value. An amplification of the middle part is depicted in Figure 15.5b.

The minimal value of the cost function guarantees the best solution. The aim of all simulations is to find the best solution, i.e. a solution that returns a cost value as small as possible. The difference between an identical synchronization as accounted in Equation (15.2) and a general synchronization as in Equation (15.4) is in the number of state variables used. In the case of Lorenz - Lorenz (Equation (15.2) it is logical to expect that all three state variables will be synchronized perfectly, while in the case of Rössler - Lorenz, we expect that only the synchronized variable, in this case $y_2(t)$, will be synchronized in an acceptable manner. For the Rössler - Lorenz system synchronization described by Equation (15.3) and the cost function in Equation (15.4) we obtain a fitness landscape as shown in Figure 15.3b. Again, the lowest point in the fitness landscape represent the best synchronization.

The numerical examples of chaos control are defined in a similar way, see [29, 31, 33, 39]. The first example is controlling of the Henon system (Equation (15.5))

$$\begin{aligned} x_{n+1} &= a - x_n^2 + by_n \\ y_{n+1} &= x_n \end{aligned} \tag{15.5}$$

by a method called ETDAS (Equation (15.6)), see for example [22, 23, 33]. The ETDAS method calculates

$$\begin{aligned} & x_{n+1} = a - x_n^2 + by_n + F_n \\ & F_n = K((1-R)S_{n-m} - x_n) \\ & S_n = x_n + RS_{n-m} \\ & \\ & a = 6/5,\ b = 3/10,\ R = 0.32. \\ & -F_{max} \leq F_n \leq F_{max}. \end{aligned} \tag{15.6}$$

The cost function used to calculate the fitness landscape is given by

$$\begin{aligned} & CV_{Henon}(K, F_{max}) = \sum_{i=0}^{n} |TS_i - AS_i|. \\ & TS_i \ - \ target\ state\ (expected\ state\ of\ the\ system) \\ & AS_i \ - \ actual\ state\ given\ by\ the\ evolutionary\ synthesized\ controller \end{aligned} \tag{15.7}$$

Two cases of control have been calculated: stabilization of chaotic dynamics at a period-1 orbit (i.e. stabilized state), see Figure 15.6a for the resulting cost function landscape and stabilization of chaotic dynamics at a period-6 orbit (Figure 15.6b). In Equation (15.6), K and R are adjustable constants, F_n is the perturbation factor, S_{n-m} is given by a delay equation utilizing previous states of the system with m the period that is stabilized. The perturbation F_n may have an arbitrarily large value, which can cause divergence of the system outside the interval $F_n \in [0, 0.5]$ for the Henon map. Therefore, F_n should have a value between $[-F_{max}, F_{max}]$ and setting the value of F_{max} is hence another parameter to find for the chaos control scheme in Equation (15.6).

A more complex and harder problem of control is CML (coupled map lattices) control, see Equation (15.8) and [29]. A typical example is a CML based on the logistic equation, see Equation (15.9) and [4, 11, 19], which is used to simulate the behavior of the system. The CML consists of n mutually joined cells, where the neighboring cells i, $i-1$ and $i+1$ are linked via a nonlinear coupling, denoted

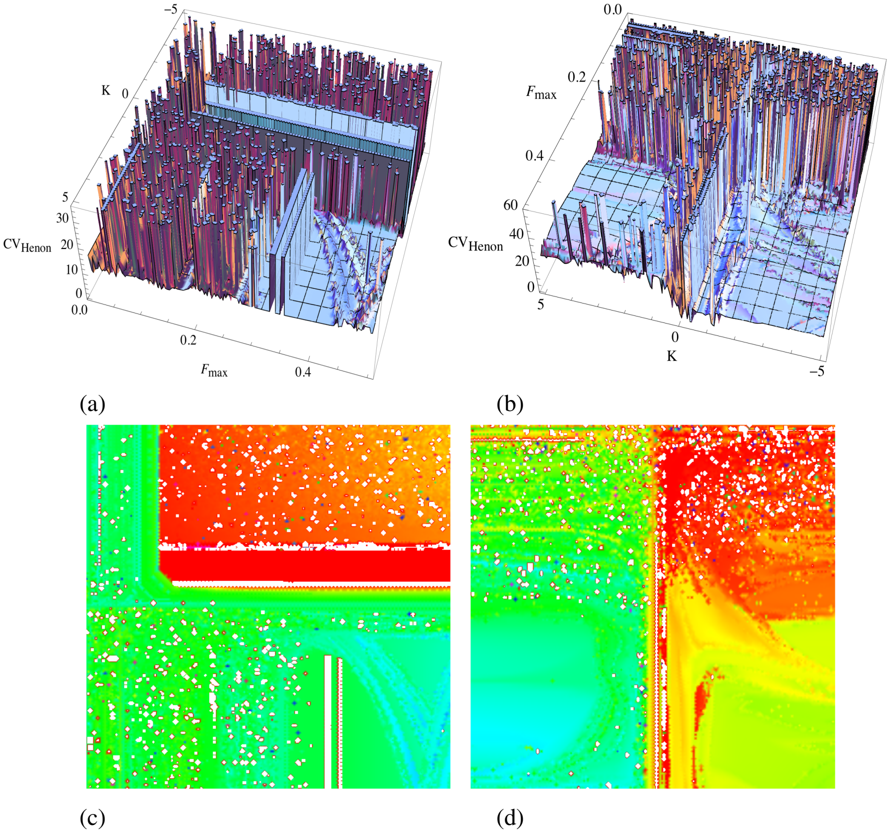


Fig. 15.6 Cost function landscape in Equation (15.7): (a) period-1 orbit stabilization. (b) period-6 orbit stabilization. (c) 2D projection of the cost function in (a). (d) 2D projection of the cost function in (b).

with $\varepsilon \in [0,1]$. With the discrete time variable k we get the temporal evolution of the CML as follows:

$$x_{k+1}(i) = (1-\varepsilon)f(x_k(i)) + \tfrac{\varepsilon}{2}(f(x_k(i-1)) + f(x_k(i+1))) \quad i = 0,1,\ldots,n. \tag{15.8}$$

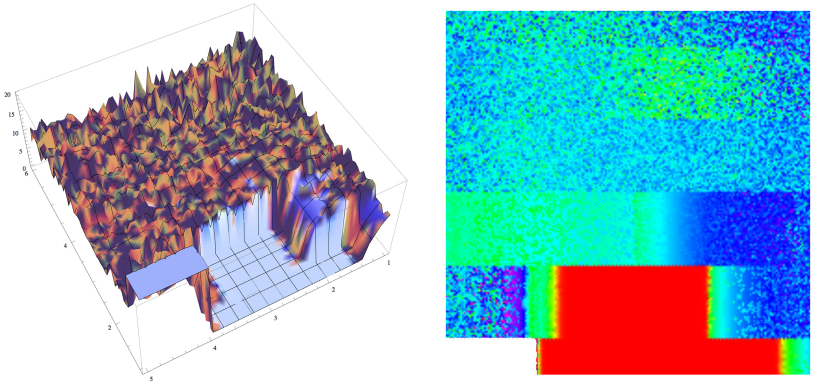
The function $f(x_n(i))$ is an “arbitrary” discrete system. In this case study we control mutually coupled n chaotic logistic equations,

$$f(x_n(i)) = Ax_n(i)(1-x_n(i)), \tag{15.9}$$

and the aim is to stabilize the whole system on predefined pattern (see [29] for classical and [39] for evolutionary control). The cost function returning the cost value is given by

$$\begin{array}{l}
CV_{CML} = p1 + \left(p2 + \sum_{i=1}^{30} \sum_{j=a}^{b} \left| T_i S_j - CML_{i,j} \right| \right)^2 \\
T_i S_j - \text{target state of CML} \\
CML_{i,j} - \text{actual state of controlled CML} \\
p1 - \text{number of actually selected pinning sites} \\
p2 - 100,\ \text{heuristically set weight constant} \\
\{a,b\} = \{80,100\}\ \text{for}\ \mathrm{T_1 S_1}\ \text{and}\ \{\mathrm{a,b}\} = \{580,600\}\ \text{for}\ \mathrm{T_1 S_2}
\end{array} \qquad (15.10)$$

The meaning of used terms is: $p1$ is the number of pining sites (PS), which is the value of control inputs (number of n possible control inputs). The term $T_x S_y$ is a description of whether the expected stabilized state is flattened (no periodicity), i.e. $x = y = 1$, or has periodicity in time, i.e. $x > 1$, or space, i.e. $y > 1$. The state of the controlled CML represents the stabilized pattern after the transient state that can have various shapes. The cost value CV_{CML} is calculated so that in the given range of CML iterations we compare the expected CML states and the actually stabilized CML states under the given control. The differences were summarized and the lowest difference has been searched for. The fitness landscape, describing various control qualities, is shown in Figure 15.7. The coordinates of the lowest point at this fitness landscape is the best control setting.



(a) (b)

Fig. 15.7 Fitness landscape from CML chaos control problem described in Equation 15.10. (a) 3D fitness landscape. (b) 2D projection.

15.3 Harmonic and Fractal Image Analysis

HarFA[1] is a software package that was developed to perform harmonic and wavelet analysis of digitized images and to calculate their fractal parameters. Harmonic analysis means 1D or 2D Fourier transform of image information. The harmonic analysis is provided by using a fast Cooley - Tukey discrete Fourier transform algorithm (DFFT). This algorithm works if the length of the dataset is N (or $N \times N$ in 2D case), where N is an integer power of 2. Therefore the size of the analyzed area can be set to 32, 64, 128, 256, 512, 1024, ... pixels. The dataset can be filled by values of intensity (shades of gray), hue, red, green or blue components of the color information. The Fourier transform is presented as magnitude transfer function (MTF), phase transfer function (PTF), and real and imaginary part of the Fourier spectrum. The harmonic analysis can be performed either in 1D space or in 2D space. The results can be viewed as 1D, 2D (transformed images) and 3D (2D FFT only) graphs. 1D graph data can be stored as text files. Transformed images can be saved as bitmaps. A new method for calculating fractal dimensions and fractal measures is based on this harmonic analysis.

Wavelet analysis means a 1D or 2D Haar transform of the image information. The wavelet analysis is provided by using fast algorithmic calculations. This algorithm works if the length of the dataset is N (or $N \times N$ in 2D case), where N is an integer power of 2. Therefore, the size of the analyzed area can be set to 32, 64, 128, 256, 512, 1024, ... pixels. The dataset can be filled by values of intensity (shades of gray), hue, red, green or blue components of color information. The wavelet analysis can be performed either in 1D space or in 2D space. The results can be viewed as 1D, 2D (transformed images) and 3D (2D HT only) graphs. 1D graph data can be stored as text files. A new method for calculating fractal dimensions and fractal measures is based on this wavelet analysis. For special cases (thresholded figure with area size equal to 1, 2, 4, 8, 16, 32, 64, 128, 256, ...), the calculated fractal dimension is equivalent to the classical box-counting method. The fractal analysis is a means to determine the fractal dimension and the fractal measure of an image. The fractal dimension and the fractal measure are obtained by using a variation of the box counting method. By this technique we can examine black and white fractal structures which come into existence during a process called thresholding. Thresholding transforms colored image objects into black and white ones. There are many criteria which can be changed to derive many different fractal structures from one image (e.g. one can alter the minimal value of hue to be thresholded as black, or one can determine that black will be all pixels which fulfill the conditions of their RGB channels, e.g. ($87 \leq \mathrm{R} \leq 145$) AND ($63 \leq \mathrm{G} \leq 146$) AND ($77 \leq \mathrm{B} \leq 255$)), and all the other pixels become white). So one can get various fractal dimensions and measures for one image. If one wants to characterize an image by one of its fractal dimensions, it may be unclear which of them is appropriate. Therefore there is the possibility to establish a fractal dimension of an image in the whole range of

[1] HarFA (Harmonic and Fractal image Analyser) [42, 43] is a software package developed at the Brno University of Technology, Czech Republic. For download and further information, see `http://www.fch.vutbr.cz/lectures/imagesci/`

thresholding conditions. A fractal spectrum is then obtained, where the fractal dimension is presented as a function of the thresholding condition (e.g. fractal dimension as a function of masked intensity (shade of gray) values).

Several filtration algorithms are included (sharpening, smoothing, median and Kuwahara filtering, different kinds of derivative filters). Tools are available to suppress the thermal noise of the charge-coupled device (CCD) cam and for the elimination of image sample illumination non-uniformity (e.g. samples prepared by using optical microscope and CCD). Image information can be handled in four color spaces - intensity (shades of gray), HSB/HSV (hue, saturation, brightness/value), HLS (hue, lightness, saturation) and RGB (red, green and blue channel).

15.4 Fitness Landscapes Exhibiting Fractal Structures

15.4.1 Fractal Geometry – Basics Ideas

At the turn of the 19th and 20th century the exploration of mathematical structures began and showed strange formations which differ considerably from the classical geometrical ideal objects. One of the first such structures, discovered by Karl Weierstrass in 1872, was a continuous function that did not have a derivative at any point. Then in 1884 Georg Cantor proposed the discontinuum - sets with nonzero dimensions less than 1. After that, in 1904, Helge von Koch introduced a curve of infinite length, which encloses the finite surface (see Figure 15.8). In 1918 Gaston Julia and Pierre Fatou described a set design called Julia sets.

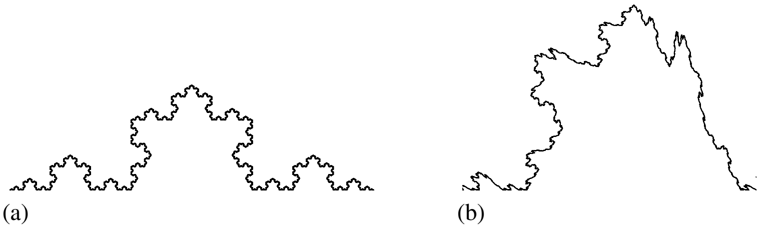


Fig. 15.8 (a) Koch snowflake (b) Irregular fractal modeling a coast line, which is a randomly modified Koch snowflake

These and other structures were received as mathematical scarecrows or monsters by the mathematical public. Later it became clear that such a geometry was able to describe some natural phenomena much better than classical geometry. The complexity of the description is not the only attribute that differs for the two different geometries. Another aspect is the dimension of the object. In classical geometry, one can find the Euclidean dimension as an integer number starting from zero. For instance, we have a dimension of 1 (line), two dimensions (surface) and three dimensions (space). The fractal geometry [16, 17] is different from traditional geometry as we can have non-integer dimensions. Such a non-integer dimension can be formalized as the (fractal) Hausdorff dimension, which is a real and often an irrational number for many objects. It has been shown that the Euclidean dimension is the limit of the fractal dimension. In other words, the fractal dimension may even be integer, which turns out to be equal to the Euclidean dimension. Fractal objects usually show high levels of complexity (see Figure 15.8), that can often be observed in natural forms and processes as well as in the solutions of the engineering problems, as demonstrated in the further sections. For more about fractal geometry we refer to [1, 16, 17, 20].

15.4.2 Fractal Dimension

It is possible to describe mathematically the degree of structuring of objects by dimensions in form of special numbers. Those are numbers characterizing fractals like the Hausdorff - Besicovic (fractal) dimension [1, 20]. For fractal objects the numerical value of this dimension is larger than the topological dimension, but smaller than the Euclidean dimension. Non-fractal objects have the property that, by reducing the length scale of the object (circuit), this length goes to a limit value. This is not the case for fractals, the length is constantly increasing. This feature is called the Richardson effect. There are several definitions of dimensions, which in principle can be divided into two groups:

- Metric dimensions, depending on metric properties.
- Information dimensions, depending on probabilistic properties.

An example of the group of metric dimensions is the Hausdorff-Besicovic dimension, also called Kolmogorov dimension or capacity, eventually fractal dimension. It is expressed by the following formula:

$$D_k = \lim_{\varepsilon \to 0} \frac{\ln N(\varepsilon)}{\ln\left(\frac{1}{\varepsilon}\right)} = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log\left(\frac{1}{\varepsilon}\right)}, \tag{15.11}$$

where $N(\varepsilon)$ is the minimum number of elementary bodies (e.g. R^2 squares of side length ε) needed to cover the considered set. This dimension reflects quantitatively the degree of complexity (“structured”) of a given set. From Equation (15.11) it can be seen that the value of the fractal dimension does not depend on the base used for the logarithm. Information dimensions are very often used in dynamical systems as they are suitable for describing the time evolution of systems. Information dimensions can also be applied if the time evolution of the system is stochastic, which leads to a probabilistic description of dynamics. An example of information dimensions is the Lyapunov dimension.

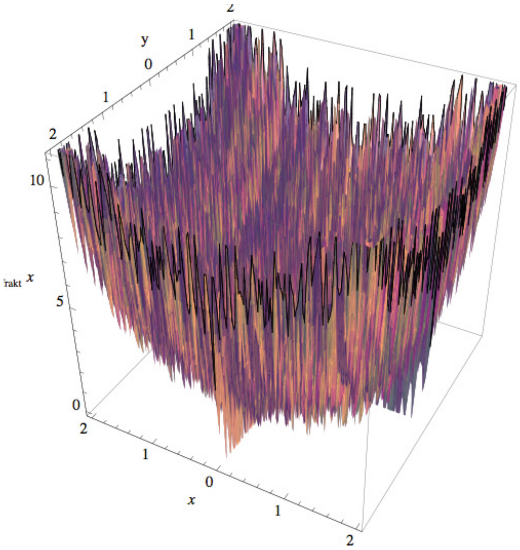


Fig. 15.9 The fractal function - 1^{st}DeJong function with added fractal function

15.4.3 Fractal Fitness Landscape – How to Build It

A class of fitness landscapes exhibits fractal structures. In fact, real fractals, as defined, do not exist in the real world, they are only mathematical structures living in the mathematical universe. On the other hand, we can numerically approximate them on computers to study their properties. For this purpose we define a fractal function that can been added to selected classical test function, in this case the 1^{st}DeJong, see Table 15.1 #1. The resulting function is then an artificial fractal test function. The complexity and fractalness of such a function can be varied by parameters of the fractal dimension, see Figure 15.10. This kind of test function can be seen in two ways. In a first it can be taken as a special mathematical object which is infinitely complex. This perspective of a fractal makes localization of its extreme theoretically impossible. This fact stems from the principles of fractal geometry, see for instance Mandelbrot [16], according to which self-similar or self-affine structures can be observed. The exact shape of the fractal (especially in the case of self-affine fractals) is difficult to predict and therefore the precise location of the global extreme of fractal functions is impossible. From a practical point of view it can be stated that the use of fractal test functions for today's digital computers has its limitations in terms of accuracy. A computer-generated "fractal" always has a finite complexity. Taking into account the above limitation, fractal test function can be used. For fractal functions we chose a modification of the Weierstrass - Mandelbrot function [16], which is given as an infinite series

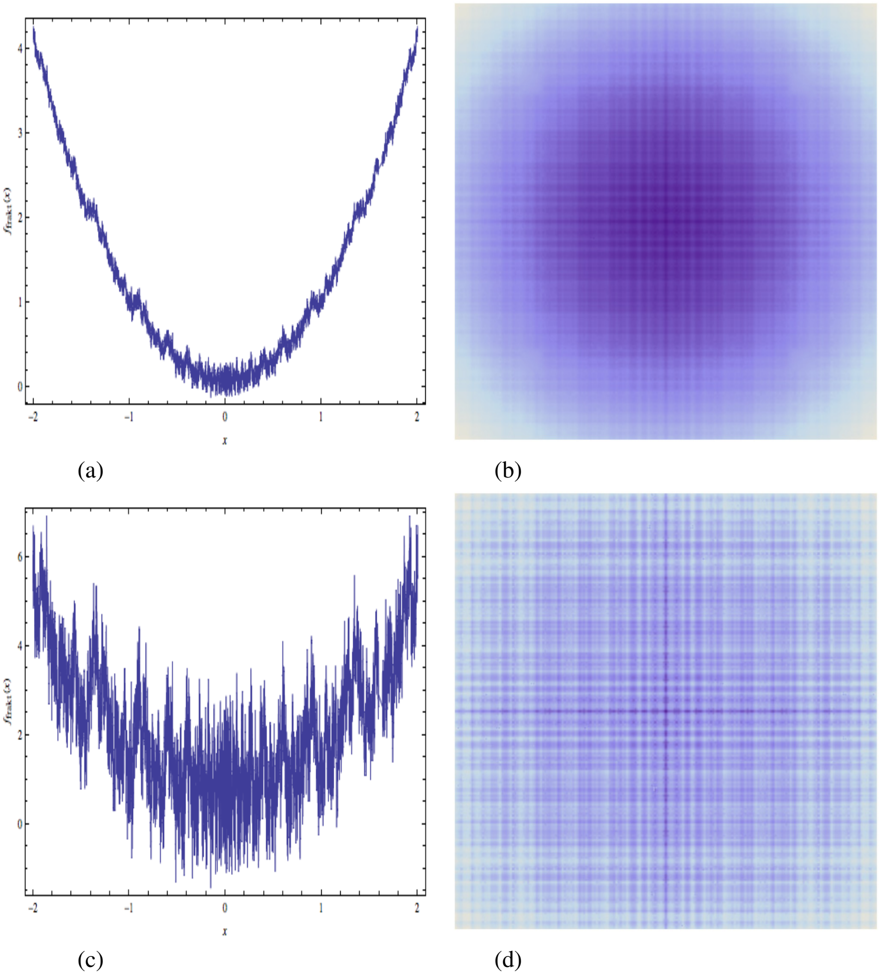


Fig. 15.10 1^{st}DeJong function with added fractal function ($D = 1.85$). (a) Amplification of the fractal component $A = 1$; (c) $A = 10$; (b) and (d) 2D projection that is used to perform the fractal analysis.

$$W(x) = \sum_{j=-\infty}^{\infty} \frac{\left(1 - \exp\left(ib^j x\right)\right) \exp\left(i\varphi_j\right)}{b^{(2-D)j}} \tag{15.12}$$

where i is the imaginary unit, $b > 1$ affects the optical clarity of graph complexity, ϕ_j is an arbitrary phase angle, and D ($1 < D < 2$) is the fractal dimension of the curve W. This curve, discovered by Karl Weierstrass in 1872, is the graph of a continuous function, but it has no derivative at any point. This mathematical property also applies to the real part of the simplified function in Equation (15.12) ($\phi_j = 0$):

$$C(x) = \sum_{j=-\infty}^{\infty} \frac{1-\cos\left(b^j x\right)}{b^{(2-D)j}} \,. \tag{15.13}$$

This version of the Weierstrass - Mandelbrot function is also called the Weierstrass - Mandelbrot fractal cosine function [16]. According to [2], Equation (15.13) shows a trend, which is influenced by parameter D. For our numerical experiments we have set the summation over j in Equation (15.13) to the interval $[-100, 100]$ (see Figure 15.9 and an application on the 1^{st}DeJong function see Figure 15.10). The boundary value of 100 was estimated heuristically. When Equation (15.13) is plotted for different j, then after $j > 50$, the value for $C(x)$ is saturated, so the value of 100 shall be enough. To get a fractal function free of any kind of trend, a variant of the function in Equation (15.13) was designed in which the relevant trend is eliminated. This variant is given by Equation (15.14):

$$C'(x) = \begin{cases} \frac{C(x)}{C(1)|x|^{2-D}}\,, & \text{if} \quad x \neq 0 \\ 1 & \text{for} \quad x = 0\,. \end{cases} \tag{15.14}$$

Equation (15.15) demonstrates the use of the fractal function in Equation (15.14) on the 1^{st}DeJong test functions:

$$f_{fract}(\mathbf{x}) = \sum_{i=1}^{n} \left(AC'(x) + x_i^2 - 1\right) \tag{15.15}$$

with A an amplification factor for the fractal component of the landscape. Figures 15.9 and 15.10 show the effect. The resulting complexity and fractalness of the landscape can be controlled by parameter D in Equation (15.14). In this way we can fractalise any kind of fitness landscape.

15.5 Fractal Analysis – Selected Methods

The entropy [6, 13] of a region of size r can be determined by either radius fractal dimension or box counting fractal dimension. The radius method of fractal dimension calculation is proper for spherical structures (e.g. a snowflake as in Figure 15.11), the box counting can be used generally for any fractal structure. The box counting method can be done by using a wavelet (Haar) transformation method.

The radius method (Figure 15.11b) is based on the distribution of black and white pixels at thresholded images in defined regions with radius r. The fractal dimension [15, 17] is determined from the logarithmic dependence of the number of white (or black) pixels on the logarithm of the radius. The fractal dimension of white pixels gives information about the fractal character of the thresholded image of the snowflake (Figure 15.11a).

The box counting method (Figure 15.11c) is based on the number of white (partially white and black) squares with varying size of edge length $\varepsilon = 1/r$ of the thresholded image. The fractal dimension is determined from the logarithmic

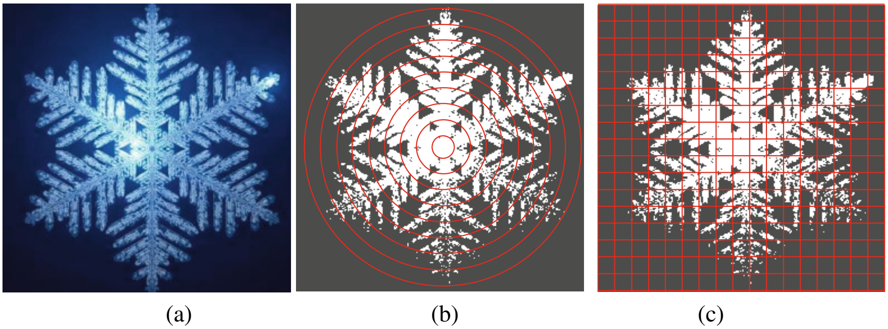


Fig. 15.11 The principle of fractal dimension calculation: (a) original image, (b) radius method, (c) box counting method applied to thresholded image brightness component

dependence of the number of white squares (all pixels in a square are white), squares with some white pixels, and black squares (all pixels in a square are black) on the logarithm of the edge length. The fractal dimension calculated from the number of white squares (D_{WBW}) gets the information about the fractal character of the thresholded image of the snowflake, the fractal dimension calculated from partially white squares (D_{BW}) about the interface of the snowflake, and the fractal dimension calculated from the number of black squares D_{BBW} (including the interface) about the surrounding of the snowflake. This method can be used for any fractal structure found in a thresholded image. The fractal measures can be used to determine areas of white (K_{WBW} - include interface), black (K_{BBW} include interface), and of borderlines between white and black regions (K_{BW}) [12].

The wavelet method is based on using a two-dimensional Haar transformation [12, 36]. It is an integral transformation which is based on "square-shaped" functions (called Haar wavelet, see Figure 15.12 top left). Wavelet analysis is similar to Fourier analysis in the sense that it allows a target function over an interval to be represented in terms of an orthonormal function basis. The principle of the wavelet method is based on rasterization of black and white images (see Figure 15.12). Rasterization can be generally realized by low level filtering. In the case of Haar transformation it can be very simply calculated by averaging the nearest four pixels (or squares of rasterized image). Note that by averaging all white pixels (squares), the new square will be white, by averaging all black pixels (squares), the new square will be black, and for all other combinations the new square will be gray scaled. The fractal parameter calculation is the same as for the box counting method. Both methods are identical for squares sizes: 1, 2, 4, 8, ..., pixels.

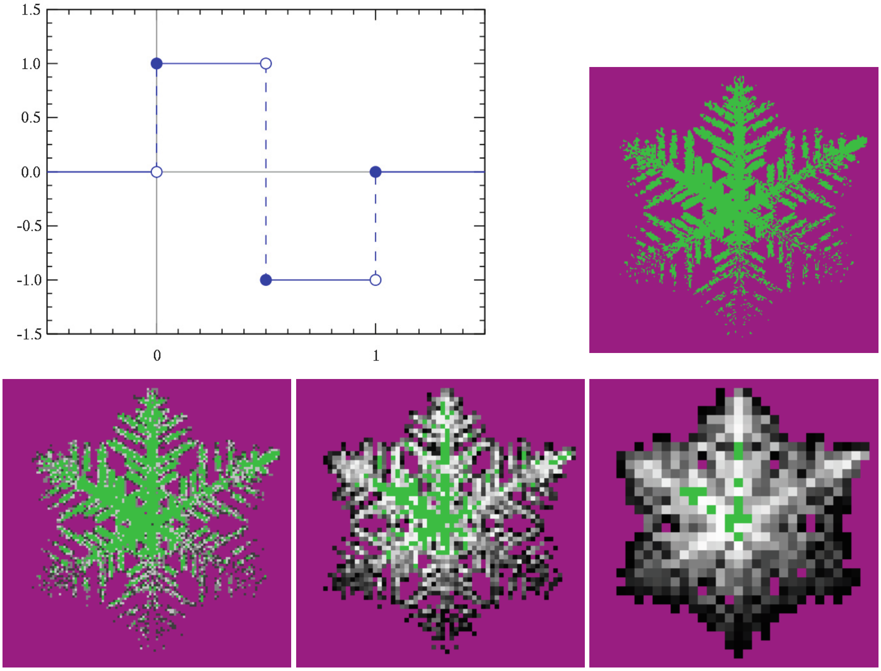


Fig. 15.12 Principle of using wavelet method for determination of fractal parameters (violet pixels are black, green pixels are white, other pixels are grayscale). Haar wavelet left upper image.

15.5.1 Fractal Analysis of One- and Two-Dimensional Landscapes

Fractal analysis is used to determine the fractal parameters (the fractal dimension D and the fractal measure K) and other quantities derived from them. It can be applied to one or more dimensional mathematical (fractal) structures using discrete experimental data. Fractal parameters can be defined using the area under the 1D functions (see Figure 15.10 for the studied landscape and Figure 15.13 for results) or statistical parameters: mean, variance (mean, standard deviation) and higher moments (skewness, kurtosis). The fractal dimension also determines the entropy of the data under analysis. We examine the fractal 1D function in Equation (15.15) which originates from superposition of the 1^{st}DeJong function and the fractal function in Equation (15.14) with $D = 1.85$ (Figure 15.10). The results of the fractal analysis using HarFA v.5.5 are shown in Figure 15.13. The figure shows a logarithmic dependence of the number of squares on the size of squares (the box counting method has been used). The slope of this dependence determines the fractal dimension, the intersection of the size scale $r = 1$ gives the fractal measure. Regression lines apply for the area under the curve, including the interface ($B+BW$), and for the area above the curve, including the interface ($W+BW$). Similarly, fractal parameters of two-dimensional functions can be determined. The basic methods of analysis are called threshold analysis (it determines the so-called Hausdorff dimension) and non-threshold analysis (it determines the so-called information dimension). For threshold functions (the values below a defined threshold is assigned to black, the values above this threshold are white) we can determine the fractal parameters depending on the size of the threshold (Figure 15.14) and the size of black surfaces, white surfaces and interfaces (or lengths bounding curves) for various values of thresholds, as well as other statistical parameters as for $1D$ functions. In Figure 15.14 we give the results for a 2D image analysis of thresholds for the 2D 1^{st}DeJong function (see Equation (15.15) and Figure 15.9) depending on the value of the threshold (0-255). Figure 15.14(a) describes the dependence of the fractal dimension. Figure 15.14(b) describes the dependence of the fractal measure for determining parameters of black areas, including the interface ($B+BW$), white areas, including the interface ($W+BW$), and the desktop interface (BW). It is obvious that the degree of fractal dimension and fractal interface is largest for the threshold value 105, the value of the fractal dimension is 1.805, a value close to the value that was used for the calculation ($D=1.85$).

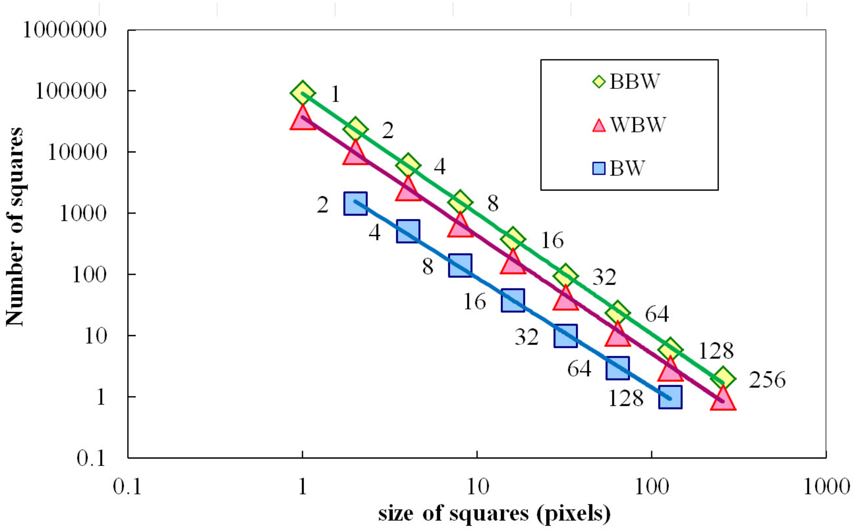
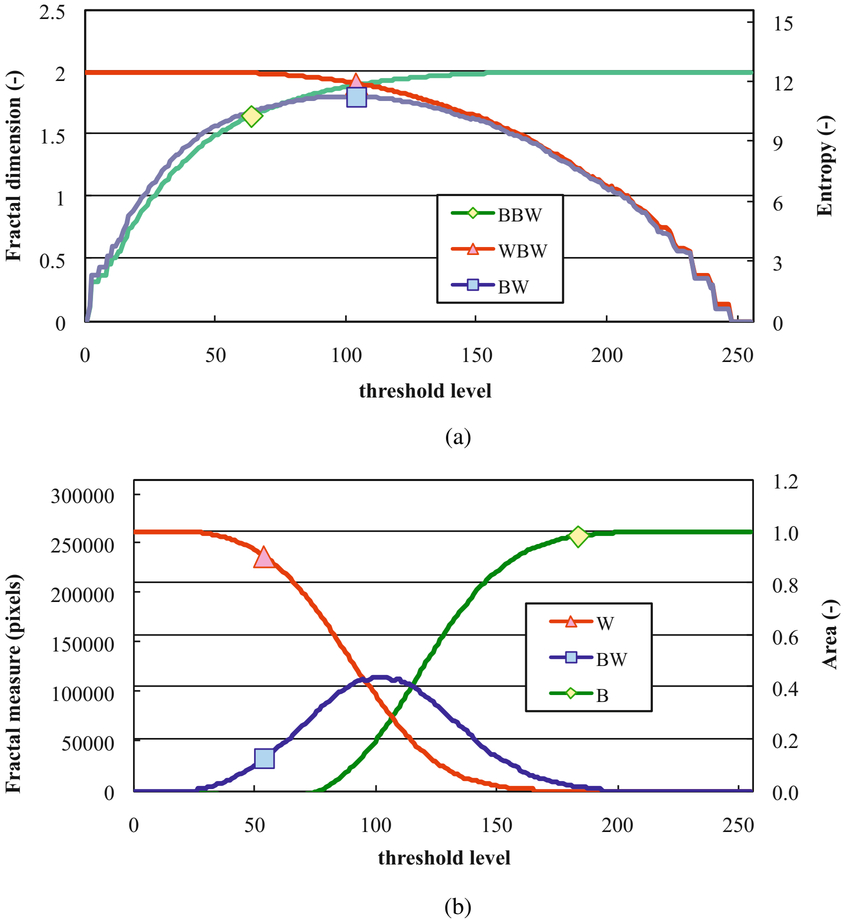


Fig. 15.13 Fractal analysis of fractal 1D 1^{st}DeJong function (15.15). WBW white squares, BW partially white squares, BBW black squares.



(a)

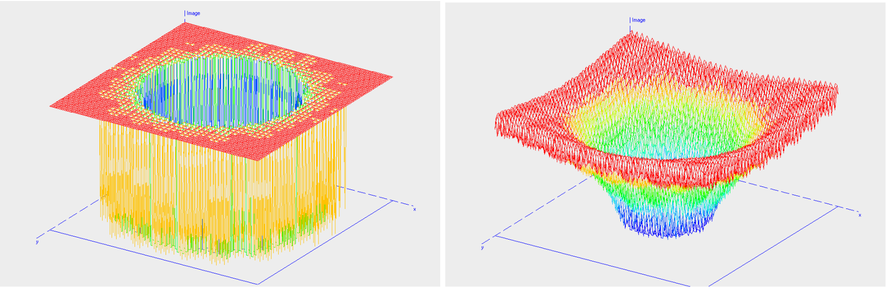
(b)

Fig. 15.14 Fractal analysis of the fractal 2D 1^{st}DeJong function in Equation (15.15). (a) Fractal dimension and (b) fractal measure.

15.5.2 Fractal Analysis of Threshold Functions

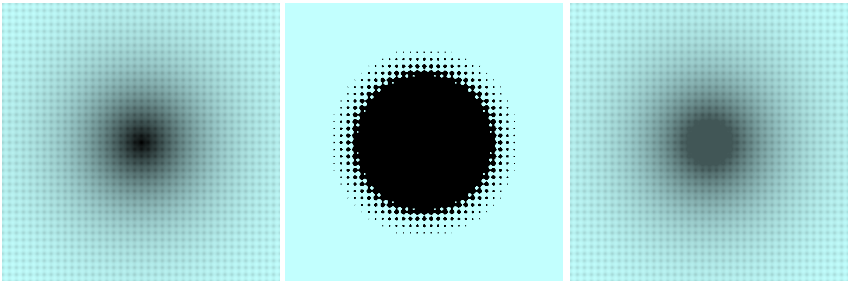
The next example we consider is the Ackley function as presented in Table 15.1 #6, and in Figure 15.1a. We study the landscape derived from this function (called the original function) and also the threshold and flood functions shown in Figure 15.15. The following parameters can be determined from fractal analysis

- Fractal dimension end entropy of the original function
- Fractal measure and volumes (surface) of the original function
- Fractal dimension and entropy of the cut (threshold) function
- Fractal measure and surfaces of the cut areas of the function
- Fractal dimensions and entropies of the flood function
- Fractal measures and volumes (surface) of the flood function



(a) (b)

Fig. 15.15 (a) Analysis of the threshold function - Ackley (threshold level = 210); (b) Analysis of the flood function - Ackley (flood level = 105)



(a) (b) (c)

Fig. 15.16 Analysis of (a) original, (b) threshold and (c) flood function

The threshold and flood regime used in the analysis are presented in Figure 15.16. The original function (Figure 15.16(a)) was cut for different levels and the resulting threshold images were analyzed. The dependence of the area size and the surrounding curve on the threshold level is analyzed in Figure 15.16(b). From the subtraction of the volumes of original and flood functions (Figure 15.16(c)) volumes of a part of the function can be determined. The different types of functions were analyzed.

Some of these functions have Euclidean character (with an integer-valued dimension), and some have a fractal character (with a real-valued dimension) of different complexity.

The results of the Ackley image analysis are presented in two graphs, Figures 15.17 and 15.18. Figure 15.17 depicts the dependences of all three Hausdorff fractal dimensions (left axis) and appropriate entropies (right axis) on the threshold level calculated for the whole image. The *BBW* line describes the dependence of the fractal dimension of the black part of the threshold image together with the interface ($B+BW$) on the threshold level. The *WBW* line analogically describes the dependence of the fractal dimension of the white part of the threshold image together with the interface ($W+BW$) on the threshold level; the *BW* line describes the dependence of the fractal dimension on the interface only. The whole image is black and the fractal dimension $D_{\mathrm{BBW}} = 2$ for low threshold levels, for high threshold levels the whole image is white and the fractal dimension $D_{\mathrm{WBW}} = 2$. The fractal dimension of the interface D_{BW} has its maximum near the midpoint of the dependence (the structure is most highly varied). The dependences of entropies for all three fractal dimensions are calculated using the equation $S_0(r) = D_0 \ln r$, where $r = 512$ pixels. From the figure it is evident that the entropy of white (WBW) is highest for the zero threshold level (probability of one black pixel at white areas is maximal), and the entropy of black (BBW) is highest for the maximal threshold level (probability of one white pixel at black areas is maximal). The maximal entropy for this cases is: for $r = 512$ pixels and $D_{\max} = 2$ we obtain $S_{\max} = 2 \ln 512 \cong 12.476$.

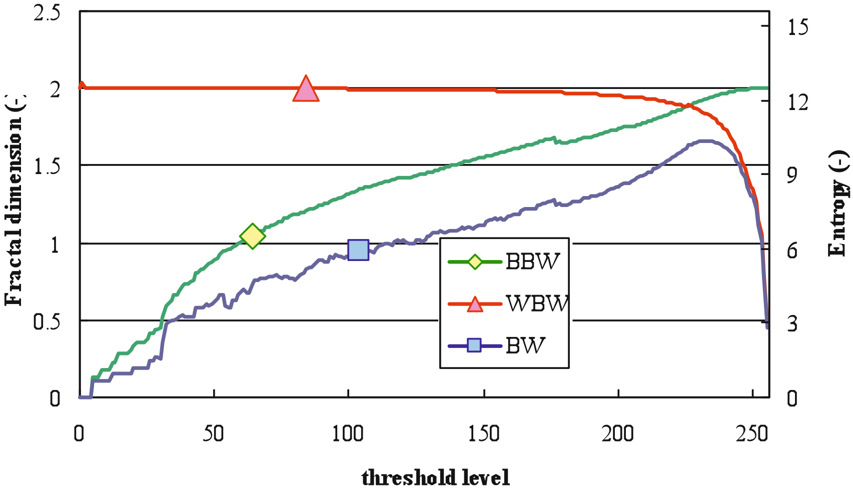


Fig. 15.17 The results of threshold analysis of the Ackley function. Dependence of all three fractal dimensions (*WBW* - white surface include interface, *BBW* - black surface include interface, and *BW* - interface) on the threshold level. Right axis is recalculated using Equations (15.16) - (15.18) order $q = 0$, size of analyzed region $r = 512$ pixels.

The values of the fractal measure can be used to determine area sizes. The values of all three fractal measures $K_{\mathrm{BBW}}, K_{\mathrm{WBW}}$, and K_{BW} are recalculated to the fractal measures, $K_{\mathrm{B}}, K_{\mathrm{W}}$, and K_{BW} (left axis) [12, 42]. The values are proportional to the size of the white area S_{W}, the black area S_{B} and the interface area S_{BW} (plotted at the graph in relative units, $S_{\mathrm{B}} + S_{\mathrm{W}} + S_{\mathrm{BW}} = 1$, right axis) as defined by Equations (15.16) - (15.18). Note that this set of equations is related to threshold 2D pictures. For fractal analysis we further take into account the fractal measure ($K_{\mathrm{S\_BBW}}$, $K_{\mathrm{S\_WBW}}$, $K_{\mathrm{S\_BW}}$; lower index S means that K is related to the surface, see Equation (15.16) - (15.18)) and the fractal dimension of threshold images (D_{WBW}, D_{BBW}, D_{BW}). The measures are counting for the black area (S_{B}), the white area (S_{W}) and the border (S_{BW}, this area is proportional to the boundary curve):

$$S_{\mathrm{B}} = \frac{K_{\mathrm{S\_BBW}} - K_{\mathrm{S\_BW}}}{K_{\mathrm{S\_BBW}} + K_{\mathrm{S\_WBW}} - K_{\mathrm{S\_BW}}} \tag{15.16}$$

$$S_{\mathrm{W}} = \frac{K_{\mathrm{S\_WBW}} - K_{\mathrm{S\_BW}}}{K_{\mathrm{S\_BBW}} + K_{\mathrm{S\_WBW}} - K_{\mathrm{S\_BW}}} \tag{15.17}$$

$$S_{\mathrm{BW}} = \frac{K_{\mathrm{S\_BW}}}{K_{\mathrm{S\_BBW}} + K_{\mathrm{S\_WBW}} - K_{\mathrm{S\_BW}}}. \tag{15.18}$$

The maximal fractal measure has a value of $K_{max} = K_{\mathrm{S\_B}} + K_{\mathrm{S\_W}} + K_{\mathrm{S\_BW}} = (512 \times 512)$ pixels.

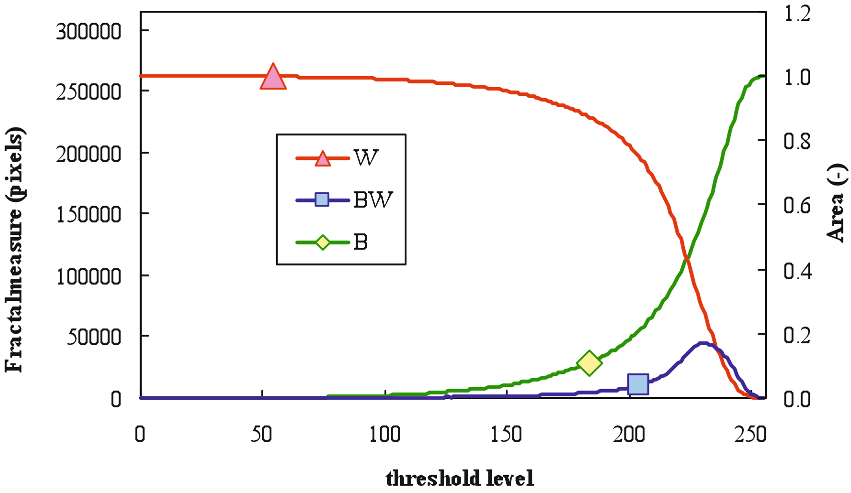


Fig. 15.18 The results of threshold analysis of the Ackley function. Dependence of all three fractal measures (*W* - white surface without interface, *B* - black surface without interface, *BW* - surface of interface proportional to the interface length) on the threshold level. Right axis is recalculated to relative values $(B + BW + W = 1)$.

Quite interesting is a comparison of the fitness landscapes related to experiments with deterministic chaos (see Figure 15.19 and 15.20). Both figures summarize the results obtained from the cost function landscapes, where L1 refers to chaos control as in Equation (15.7) with period-1 illustrated in Figure 15.6(a). L2 refers to the Rössler - Lorenz synchronization in Equation (15.4) illustrated in Figure 15.3(b). L3 refers to chaos control in Equation (15.7) with period-6 shown in Figure 15.6b. Finally, L4 refers to chaos control of the CML problem in Equation (15.10) shown in Figure 15.7. The landscapes from Figures 15.6(a) and 15.6(b) have an almost identical characteristic of the fractal dimensions, while the landscape in Figure 15.3(b) differs in the boundary areas (for small and big values of the threshold). An interesting result is observable in the landscape in Figure 15.7, where there is a big visible maximum. This fitness landscape has been generated by the CML system, so its surface is generated by a chaotic system of a different kind. This is maybe the source of that difference and deserves further research. It is also observable that the fractal measures are almost the same for all chaotic structures. The measure are slowly decrease.

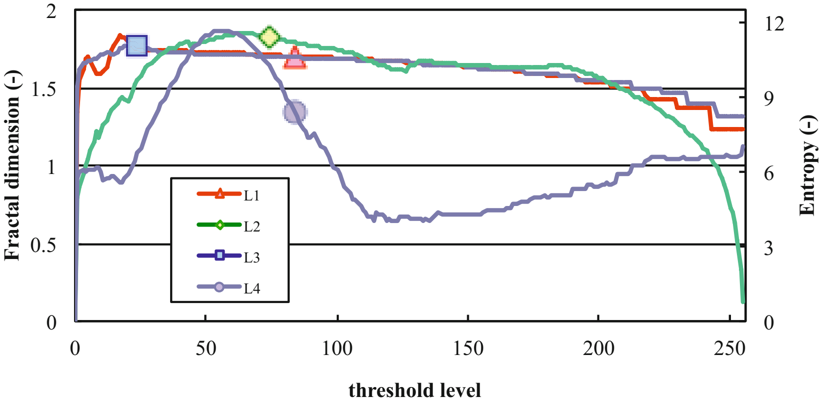


Fig. 15.19 Dependence of fractal dimension on threshold level. Mutual comparison of fitness landscapes; L1: Chaos control in Equation (15.7), period-1, Figure 15.6(a); L2: Rössler - Lorenz synchronization in Equation (15.4), Figure 15.3(b); L3: Chaos control in Equation (15.7), period-6, Figure 15.6(b); L4: Chaos control of CML problem in Equation (15.10), Figure 15.7.

15.5.3 Fractal Analysis of Original (Non-threshold) Functions

The information content [10], fractal dimensions [15, 16] and Shannon entropies [30] $S_1(r) = D_1 \ln r$ of volumes below (including surface, marked as BBW), under (including surface, marked as WBW) and of surfaces (marked as BW) can be calculated for original functions by a procedure described in [42]. The value of the

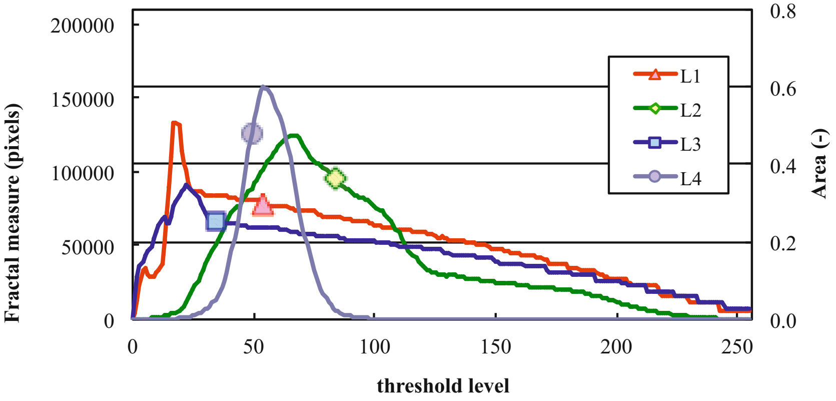


Fig. 15.20 Dependence of fractal measure on threshold level. Mutual comparison of fitness landscapes; L1: Chaos control in Equation (15.7), period-1, Figure 15.6(a); L2: Rössler - Lorenz synchronization in Equation (15.4), Figure 15.3(b); L3: Chaos control in Equation (15.7), period-6, Figure 15.6(b); L4: Chaos control of CML problem in Equation (15.10), Figure 15.7.

dimension for an Euclidean volume is equal to 3, for an Euclidean surface it is equal to 2. If a fractal character of volumes and surfaces occurs, fractal dimensions of volumes are lower than 3, and fractal dimensions of surfaces are higher than 2. The entropy of a fractal function depends on the size of the analyzed image, in our case (256 × 256) pixels. As is evident from the definition of the Shannon entropy [30], the zero value is for square (1 × 1) pixels and the slope of the change in entropy with the square size is a fractal dimension.

The fractal dimensions of all analyzed surfaces are presented in Table 15.2. From these results it is evident that some simple surface such as the 1^{st}DeJong has Euclidean character, and its fractal dimension is less than 2.1. More complex structures such as the fractal landscapes from chaos control in Equation (15.7), period-6, Figure 15.6(b) have a dimension of the surface that is higher than 2.9.

The fractal dimensions involve information about the volumes and surfaces of fractal functions. The fractal measures can be used for the determination of volume and surface sizes. The values of all three fractal measures ($K_{\mathrm{V\_BBW}}, K_{\mathrm{V\_WBW}}$, and $K_{\mathrm{V\_BW}}$) are recalculated to fractal measures ($K_{\mathrm{V\_B}}, K_{\mathrm{V\_W}}$, and $K_{\mathrm{V\_BW}}$, left axis) [12, 42]. The values are proportional to the size of volumes below V_{B}, volumes under V_{W} and volumes of surfaces V_{BW} (at graph in relative units, $V_{\mathrm{B}} + V_{\mathrm{W}} + V_{\mathrm{BW}} = 1$), as defined by Equations (15.19), (15.20) and (15.21). Note that this set of equations is related to threshold 3D pictures (the third dimension is for example the gray level). Again the fractal dimensions and fractal measures are calculated and this time the volume under the surface area including the volume defined levels of shade ($K_{\mathrm{V\_BBW}}$) over the volume above surface ($K_{\mathrm{V\_WBW}}$) and volume area ($K_{\mathrm{V\_BW}}$, is proportional to the surface; lower index V means that K is related to the volume in Equation (15.19) - (15.21)) is the result. From these parameters, which have the same meaning as in the threshold pictures, the volume under the surface (V_{B}), the volume over the area (V_{W}) and the volume area (V_{BW}) are counted. The equations therefore differ only on the left side, meaning the parameter on the right side is the same, but in the first case, the threshold images are taken into consideration while in the latter case the non-threshold images are considered:

$$V_{\mathrm{B}} = \frac{K_{\mathrm{V\_BBW}} - K_{\mathrm{V\_BW}}}{K_{\mathrm{V\_BBW}} + K_{\mathrm{V\_WBW}} - K_{\mathrm{V\_BW}}} \tag{15.19}$$

$$V_{\mathrm{W}} = \frac{K_{\mathrm{V\_WBW}} - K_{\mathrm{V\_BW}}}{K_{\mathrm{V\_BBW}} + K_{\mathrm{V\_WBW}} - K_{\mathrm{V\_BW}}} \tag{15.20}$$

$$V_{\mathrm{BW}} = \frac{K_{\mathrm{V\_BW}}}{K_{\mathrm{V\_BBW}} + K_{\mathrm{V\_WBW}} - K_{\mathrm{V\_BW}}}. \tag{15.21}$$

The maximal fractal measure will be calculated by multiplying the size of the image (512 × 512) by the number of scale levels of e.g. brightness (256) and has the value $K_{\mathrm{max}} = K_{\mathrm{V\_B}} + K_{\mathrm{V\_W}} + K_{\mathrm{V\_BW}} = (512 \times 512 \times 256)$ voxels (volume pixels) (see Table 15.2). The fractal measures of all analyzed surfaces are presented in Table 15.2. From these results it is evident that some simple surfaces such as 1^{st}DeJong have Euclidean character, and its fractal measure is very low (two orders less than more complex fractal structures, for instance the fractal landscapes from chaos control in Equation (15.7), period-6, Figure 15.6(b).

Table 15.2 Numerical values of fractal measure and dimension

	Fractal Dimension	Fractal Measure
1st DeJong, Table 15.1 #1	2.081	0.002
Rastrigin, Table 15.1 #2	2.227	0.004
Fractal function, Equation (15.15), $A = 1$, Figure 15.10a	2.344	0.007
Schwefel, Table 15.1 #3	2.347	0.009
EggHolder, Table 15.1 #7	2.432	0.013
Ackley, Table 15.1 #6	2.489	0.009
StretchedSine, Table 15.1 #5	2.533	0.028
Rana, Table 15.1 #8	2.551	0.036
Griewangk, Table 15.1 #4	2.656	0.053
Chaos control of CML problem in Equation (15.10), Figure 15.7	2.678	0.023
Fractal function, Equation (15.15), $A = 10$, Figure 15.10c	2.788	0.062
Rössler - Lorenz synchronization in Equation (15.4), Figure 15.3b	2.799	0.068
Chaos control in Equation (15.7), period-1, Figure 15.6a	2.894	0.087
Pathological, Table 15.1 #9	2.910	0.114
Chaos control in Equation (15.7), period-6, Figure 15.6b	2.940	0.091

15.6 Conclusion

The research reported in this chapter focused on how fractal analysis can be used for various fitness landscapes (visualized cost functions) with different levels of complexity. For the analysis we employed wavelet (modified box counting) methods for calculating fractal parameters. Simple functions such as the Ackley function (Figure 15.1a) as well as more complex functions such as the Rana function (Figure 15.1c) have been used. We also studied fitness landscapes that originated from numerical chaos control and chaos synchronization experiments. The last class of tested fitness landscapes was generated by means of fractal geometry, and is described in Section 15.4 and depicted in Figure 15.10.

One dimensional functions were analyzed and the results (numerically estimated fractal parameters) were compared with the fractal parameters used for generating the fractal functions used in this analysis. It has been found that the difference between the original parameters and their numerical estimation is very small. For generating a fractal 1^{st}DeJong function (see Figure 15.10), we used a fractal parameter $D = 1.85$. The result of the analysis is the fractal dimension $D = 1.78$. Further analysis was carried out on threshold images. There we found a match for the function based on Equation (15.15) and depicted at Figure 15.10 for the largest fractal interface that is very good, $D = 1.805$. In this chapter, this analysis is also performed for the Ackley function.

The last analysis was carried out for 2D non-threshold functions. The result is a comparison of fractal parameters (fractal dimension and fractal measure) for all analyzed features. From the graphs it is clear that the smallest dimension belongs to the function 1^{st}DeJong (geometric features - ideal $D = 2$, the result of analysis of $D = 2.1$ due to the boundary conditions: the final image size, finite pixel size), the largest contrast function (D = 2.9).

Despite the fact that interesting results have been demonstrated, observed and confirmed with the fractal theory, a few important issues still remain unanswered. We have analyzed static fitness landscapes. However, an analysis of dynamic fitness landscapes (i.e. changing in time) has not been done and can reveal important clue about the internal dynamics of the system that "generates" the studied fitness landscape. Also the relation between evolutionary algorithm dynamics (during evolutionary search on the fitness landscape) and the fractal properties of the fitness landscape still remains unanswered and is one of our current research topics.

Acknowledgements. The following two grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic - GACR P103/13/08195S, by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic, partially supported by Grant of SGS No. SP2013/114, VŠB - Technical University of Ostrava, further by projects from the Ministry of Industry and Trade of the Czech Republic (Grant FR-TI1/144), and from "Centre for Materials Research at FCH BUT" No. CZ.1.05/2.1.00/01.0012 supported by ERDF.

References

[1] Barnsley, M.F.: Fractals Everywhere. Academic Press Professional, San Diego (1993)

[2] Berry, M.V., Lewis, Z.V.: On the Weierstrass - Mandelbrot fractal function. Proceedings Royal Society of London A370, 459–484 (1980)

[3] Brown, R., Rulkov, N.F., Tracy, E.R.: Modeling and synchronization chaotic system from time-series data. Phys. Rev. E49, 3784–3800 (1994)

[4] Chen, G.: Controlling Chaos and Bifurcations in Engineering Systems. CRC Press, Boca Raton (2000)

[5] Crona, K., Greene, D., Barlow, M.: The peaks and geometry of fitness landscapes. J. Theor. Biol. 317, 1–10 (2013)

[6] Garrido, A.: Classifying entropy measures. Symmetry 3, 487–502 (2011)

[7] Gonzalez-Miranda, J.M.: Synchronization and Control of Chaos. An introduction for scientists and engineers. Imperial College Press, London (2004)

[8] Grassberger, P., Procaccia, I.: Characterization of strange attractors. Phys. Rev. Lett. 50, 346–349 (1983)

[9] Grassberger, P., Procaccia, I.: Estimation of the Kolmogorov entropy from a chaotic signal. Phys. Rev. A 28, 2591–2593 (1983)

[10] Han, T.S., Kobayashi, K.: Mathematics of Information and Coding. American Mathematical Society, Providence (2002)

[11] Hilborn, R.C.: Chaos and Nonlinear Dynamics. Oxford University Press, Oxford (1994)

[12] Jerabkova, P., Zmeskal, O., Haderka, J.: Fractal analysis of the images using wavelet transformation. In: Complexus Mundi-Emergent Patterns in Nature, pp. 300–312. World Scientific, Singapore (2006)

[13] Higashi, M., Klir, G.J.: Measures of uncertainty and information based on possibility distributions. Int. J. General Syst. 9, 43–58 (1982)

[14] Hoshino, T., Mitsumoto, D., Nagano, T.: Fractal fitness landscape and loss of robustness in evolutionary robot navigation. Autonomous Robots 5, 199–213 (1998)

[15] Mandelbrot, B.B.: How long is the coast of britain? Statistical self-similarity and fractional dimension. Science 156, 636–638 (1967), doi:10.1126/science.156.3775.636

[16] Mandelbrot, B.B.: Fractal Geometry of Nature. W. H. Freeman and Co., New York (1983)

[17] Mandelbrot, B.B.: A fractal set is one for which the fractal (Hausdorff-Besicovitch) dimension strictly exceeds the topological dimension. In: Fractals and Chaos. Springer, Berlin (2004)

[18] Martin, C.H., Wainwright, P.C.: Multiple fitness peaks on the adaptive landscape drive adaptive radiation in the wild. Science 339, 208–211 (2013)

[19] May, R.: Simple mathematical model with very complicated dynamics. Nature 261, 45–67 (1976)

[20] Peitgen, H.O., Jürgens, H., Saupe, D.: Chaos and Fractals, New Frontiers of Science. Springer, Berlin (1992)

[21] Pikovsky, A., Rosemblum, M., Kurths, J.: Synchronization: A Universal Concept in Nonlinear Sciences. Cambridge University Press, Cambridge (2001)

[22] Pyragas, K.: Continuous control of chaos by self-controlling feedback. Phys. Lett. A 170, 421–427 (1992)

[23] Pyragas, K.: Control of Chaos via Extended Delay Feedback. Phys. Lett. A 206, 323–330 (1995)

[24] Renyi, A.: Probability theory. Elsevier, Amsterdam (1970)

[25] Ribeiro, H.V., Zunino, L., Lenzi, E.K., Santoro, P.A., Mendes, R.S.: Complexity-entropy causality plane as a complexity measure for two-dimensional patterns. Plos One 7(8) (2012), doi:10.1371/journal.pone.0040689

[26] Richardson, L.F., Ashford, O.M., Charnock, H., Drazin, P.G., Hunt, J.C.R., Smoker, P., Sutherland, I.: The Collected Papers of Lewis Fry Richardson, Cambridge (1993)

[27] Richter, H.: Analyzing dynamic fitness landscapes of the targeting problem of chaotic systems. In: Di Chio, C., et al. (eds.) EvoApplications 2012. LNCS, vol. 7248, pp. 83–92. Springer, Heidelberg (2012)

[28] Rulkov, N.F., Sushchik, M.M.: Robustness of synchronized chaotic oscillations. Int. J. Bifurc. Chaos 7, 625–643 (1997)

[29] Schuster, H.G.: Handbook of Chaos Control. Wiley-VCH, New York (1999)

[30] Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. 27, 379–423, 623–656 (1948)

[31] Senkerik, R., Zelinka, I., Navratil, E.: Optimization of feedback control of chaos by evolutionary algorithms. In: First IFAC Conference on Analysis and Control of Chaotic Systems, Reims, France, June 28-30 (2006)

[32] Senkerik, R., Zelinka, I.: E Navratil, Investigation on evolutionary EDTAS chaos control. In: Proc. 20th European Simulation Multiconference (ESM 2006), Germany, Bonn, May 28-31 (2006)

[33] Senkerik, R., Davendra, D., Zelinka, I., Oplatková, Z., Jasek, R.: Performance comparison of differential evolution and SOMA on chaos control optimization problems. Int. J. Bifurc. Chaos 22, 1230025–1–16 (2012)

[34] Stein, D.L., Newman, C.M.: Rugged landscapes and timescale distributions in complex systems. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) Proc. International Conference of Numerical Analysis and Applied Mathematics, ICNAAM (2012)

[35] Sushchik, M.M., Rulkov, N.F., Tsimring, L.S., Abarbanel, H.D.I.: Generalized synchronization of chaos in directionally coupled chaotic systems. In: Proceedings of 1995 Intl. Symp. on Nonlinear Theory and Appl., pp. 949–952. IEEE-Press, Piscataway (1995)

[36] Tomankova, K., Jerabkova, P., Zmeskal, O., Vesela, M., Haderka, J.: Use of the image analysis to study growth and division of yeast cells. J. Imaging Sci. Technol. 50, 583–589 (2006)

[37] Trochet, H.: A History of Fractal Geometry. MacTutor History of Mathematics (2009)

[38] Weinberger, E.D., Stadler, P.F.: Why some fitness landscapes are fractal. J. Theor. Biol. 163, 255–275 (1993)

[39] Zelinka, I., Celikovsky, S., Richter, H., Chen, G.: Evolutionary Algorithms and Chaotic Systems. Springer, Germany (2010)

[40] Zelinka, I., Senkerik, R., Navratil, E.: Investigation on evolutionary optimization of chaos control. Chaos, Solitons and Fractals 40, 111–129 (2009)

[41] Zmeskal, O., Nezadal, M., Buchnicek, M.: Fractal-Cantorian geometry, Hausdorff dimension and the fundamental laws of physics. Chaos, Solitons and Fractals 17, 113–119 (2003)

[42] Zmeskal, O., Nezadal, M., Buchnicek, M.: Fractal analysis of printed structures. J. Imaging Sci. Technol. 46, 453–456 (2002)

[43] Zmeskal, O., Bzatek, T., Nezadal, M.: HarFA - Harmonic and Fractal image Analyser (1997, 2012), http://www.fch.vutbr.cz/lectures/imagesci/

Chapter 16
Fitness Morphs and Nonlinear Projections of Agent–Case Embeddings to Characterize Fitness Landscapes

Daniel Ashlock, Justin Schonfeld, Wendy Ashlock, and Colin Lee

Abstract. The fitness landscape of an evolutionary computation system is a set of points defined by the representation of the potential solutions with a connectivity created by the variation operators. In real optimization, fitness landscape analysis can profitably confuse this connectivity with that provided by the usual metric structure for Euclidean space (they are actually very different). In this chapter we examine fitness landscapes for a variety of discrete problems including finding self avoiding walks, finding features for DNA sequence classification, the Tartarus AI test problem, and locating cellular automata rules. We also examine a novel real optimization problem connected with the Mandelbrot set. We use agent-case embeddings, fitness morphs, and nonlinear projection to explore the fitness landscapes of these problems in a series of extended examples. All of these techniques transform information about discrete fitness into real-valued spaces enabling both analysis and visualization.

16.1 Introduction

With the ever expanding library of problems being tackled by evolutionary computational tools, improving our ability to understand and classify problems is becoming more important than ever. One of the most helpful concepts available for determining how to approach a new problem with evolutionary computation is the *fitness landscape*.

Like many of the concepts in evolutionary computation, the idea of the fitness landscape was first introduced in the field of evolutionary biology [33], but the

Daniel Ashlock · Justin Schonfeld · Colin Lee
Department of Mathematics and Statistics, University of Guelph,
50 Stone Rd. E, Guelph, Ontario, Canada, N1G 2R4
e-mail: {dashlock,jschonfe,clee04}@uoguelph.ca

Wendy Ashlock
Department of Computer Science and Engineering,
York University, 4700 Keele Street Toronto, Ontario Canada, M3J 1P3
e-mail: washlock@sympatico.ca

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_16, © Springer-Verlag Berlin Heidelberg 2014

concept applies in an equally straightforward manner to evolutionary computation. A fitness landscape is simply a mapping, for a given problem, of all potential solutions to fitness values together with a topology created by the variation operator(s). The language of fitness landscapes is adapted from the language used to describe physical landscapes. The term *hill* is used to describe a region of high fitness (in a maximization problem), *valley* to describe a region of low fitness. Maximization problems look for hills; minimization problems look for valleys.

Understanding the fitness landscape associated with a problem or class of problems is important because it can often lead to improved performance by allowing the researcher to match the appropriate evolutionary computation method with the appropriate problem. Using the language of real world landscapes can often be deceptive due to the complexity of fitness landscapes encountered in evolutionary computation. Where a real world landscape is three dimensional, even a simple length n string gene over the alphabet $\{0,1\}$ describes an $n+1$ dimensional landscape.

It is important to realize that a fitness landscape is actually a product of three interacting elements: the representation (encoding) of potential solutions, the variability operators (commonly mutation and crossover), and the fitness function. As we will explore in this chapter, changing any of these three elements has the potential to dramatically alter the resulting fitness landscape.

Evolutionary computation encompasses a wide variety of algorithms and approaches including genetic algorithms, genetic programming, evolutionary strategies, and particle swarm optimization to name but a few. While many of these approaches share the idea of a fitness landscape, the radical variation in the representation of potential solutions as well as other algorithmic details often challenges the standard catalogue of fitness landscape analysis techniques.

16.1.1 Terminology

Much of the terminology used is borrowed from evolutionary biology as well. The encoded version of a potential solution is referred to as the *genotype*. The expressed (or evaluated) version of the potential solution is the *phenotype*. In the case where the genotype is equivalent to the phenotype, this is referred to as a *direct* encoding. If the genotype is instead expressed in a manner which requires decoding before or during evaluation, this is referred to as an *indirect* encoding.

Occasionally, when dealing with a string gene or other representation analogous to the biological reality of genetic material, we will refer to a potential solution as a *chromosome* and to the elements of that solution as *genes*. The position of the gene in the chromosome is its *locus*.

Epistasis is a measure of the degree to which the interaction between elements of the genotype influence the fitness of the potential solution. Consider two problems: onemax and an instance of the class of landscapes known as NK fitness landscapes. In the first problem, onemax, each potential solution is represented as a binary string s, and the fitness of s is simply the number of 1's in s. Thus for fitness f, we have

$f(00111) = 3$ and $f(11100) = 3$. On the other hand, in the case of the second problem, fitness is calculated by adding together functions from $\{0,1\} \to \{0,1\}$. These function are applied to loci $0 \dots N-1$ together with the loci following each. For example, let N = 5 and K = 1, and $g(0,0) = 1$, $g(0,1) = 0$, $g(1,0) = 0$, and $g(1,1) = 1$. Then the fitness $f(10011) = g(1,0) + g(0,0) + g(0,1) + g(1,1) = 0+1+0+1 = 2$, while $f(11100) = g(1,1) + g(1,1) + g(1,0) + g(0,0) = 1+1+0+1 = 3$. Consider these two functions. In the first a one is good and the goodness of a particular one is not affected by the value at any other location. In the NK-fitness landscape, the value of a one at a particular location depends strongly on the values at other loci. Understanding the role of epistasis for a particular problem/representation/operator set can often provide crucial insight into understanding the fitness landscape described by the set and thus determining the appropriate representation/operator pair to tackle the problem.

The *neighborhood* of a solution is the set of solutions which are within one application of the variation operator of the solution. Thus, the size of a neighborhood is highly dependent on both the representation and the variation operator.

A mutation which has no impact on fitness is referred to as a *neutral mutation*. A *neutral network* is a set of solutions which are all connected to each other by sequences of neutral mutations.

The remainder of the chapter is organized as follows. In Section 16.2 we will briefly describe a few standard analysis approaches as well as define the three tools focused on in this chapter. Section 16.3 uses the self-avoiding walk problem to introduce issues of representation and its impact on the character of fitness landscapes. Section 16.4 applies fitness landscape analysis techniques to a type of augmented finite state machine called a *side effect machine*. Section 16.5 demonstrates agent-case embeddings on the classical AI test problem *Tartarus*. A unique application of real parameter optimization is used in Section 16.6 to demonstrate the use of fitness morphs. Section 16.7 generalizes fitness morphs for a discrete representation encoding cellular automata rules. Finally, Section 16.8 concludes the chapter.

16.2 Analysis Technology

A great deal of research has been done concerning techniques for categorizing fitness landscapes. This includes methods for grading landscapes by their "hardness" [17], measuring the "ruggedness" of a landscape [31], or treating landscapes as network graphs [28]. As fitness landscapes are often high-dimensional and difficult to visualize or easily grasp, it is often useful to explore a number of different perspectives when attempting to understand them. The following lists give few of the more commonly examined perspectives phrased for maximization problems.

Geometric Features

- Barriers – A characterization of the valleys in a landscape.
- Local Optima – A characterization of the hills, number and frequency, in a fitness landscape.

- Neutrality - A characterization of the neutral spaces in a fitness landscape.
- Ruggedness - As with a real landscape the ruggedness of a fitness landscape is a characterization of how quickly and dramatically it changes from hills to valleys and back.

Behavioral Features

- Epistasis - Approaches for measuring the degree of epistasis in a particular fitness landscape.
- Evolvability - The extent to which a population of potential solutions can improve their fitness.
- Fitness Distance Correlation - The correlation between the fitness and the distance to the global optima.
- Walks – The behavior of walks, directed or undirected (random), through a landscape. One of the techniques discussed heavily in this chapter, *fitness morphs*, is based around a particular sort of walk, the shortest path from one optimum to another.

A more comprehensive review of each of these perspectives as well as additional approaches can be found in [24]. In this chapter we focus on three particular techniques: fitness morphs, agent-case embeddings, and nonlinear projection (multidimensional scaling).

16.2.1 Nonlinear Projection

A nonlinear projection (NLP) is an evolved projection of a set of points in a n-dimensional space onto a set of points in a 2-dimensional plane [13]. It is a variant of multidimensional scaling or ordination; a set of statistical techniques for reducing the dimensionality of high dimensional spaces while minimizing the information loss [18]. In nonlinear projection, each potential solution is a vector of (x,y) coordinates on a 2D plane. The fitness of the solution is the Pearson Correlation computed between a distance matrix on the points in the high dimensional space and a distance matrix on the vectors of (x,y) coordinates. Potential solutions are evolved by a standard evolutionary computation algorithm until the fitness converges. Nonlinear projection provides a straightforward method for visualizing high dimensional spaces in a plane where they can be easily interpreted by humans, or otherwise analyzed.

There are many software packages available for various types of ordination.[1] Among the easiest to use are those contained in the software environment R [25]. R is a special purpose language for statistics and graphics.[2] R allows users to extend its functionality by adding their own functions, which then are incorporated into packages that others can use. Two R packages contain functions useful for

[1] See `http://ordination.okstate.edu/software.htm` for a list of ordination software packages.

[2] R is free and available for download at `http://www.r-project.org`

ordination: MASS [30] and vegan [23]. MASS is a package designed to support a statistics textbook, and vegan is an R package for community ecologists.

There are two types of ordination available in R. Both take as input a distance or dissimilarity matrix. The first is classical multidimensional scaling, also known as principal coordinates analysis. Use the R function *cmdscale* for this method. It has the feature that the dimensions in the resulting diagram are ordered so that the first dimension incorporates the most variability in the data, the second the next, and so on. This means that a k+1 dimensional diagram is just a k dimensional diagram with another dimension added. This method makes the assumption that a linear projection of the data is optimal, and so is not always appropriate to use. It is analytical, so it always gives the same answer given the same input.

The second type of ordination available in R is non-metric multidimensional scaling. It makes no assumptions about the nature of the data, but requires a fixed choice of dimensions. The first two dimensions of a 3-dimensional ordination diagram will typically be different from a 2-dimensional ordination diagram built from the same data. This method can produce different results based on different initial configurations and different runtimes. The basic idea is to start with an initial configuration of the data points and then use gradient descent to maximize the goodness of fit of the distances between points with their distances in the input distance matrix. The *isoMDS* function in the MASS package and the *metaMDS* function in the vegan package implement this method. *metaMDS* incorporates *isoMDS* adding such features as attempting to find a stable solution by running the algorithm several times with different random initial configurations and doing a final rotation of the data so that the first axis reflects the principal source of variation as in classical multidimensional scaling.

In this chapter, we use both non-linear projection created from an evolutionary algorithm and non-linear projection using the R multidimensional scaling functions.

16.2.2 Fitness Morphs

A fitness morph is a plot of the fitness computed along a shortest path between two optima. The morph provides insight into the structure of the landscape. A flat fitness morph, for example, indicates the two optima are part of a neutral network. For a real-valued optimization problem, a fitness morph is relatively easy to compute by simply traveling along a straight line between the two optima and evaluating the fitness at regular intervals. When dealing with more complex representations computing the path between two optima can be more challenging. In Section 16.7 a method for computing the morph between the rules for two cellular automata is given.

16.2.3 Agent-Case Embeddings

A key task in investigating fitness landscapes is the location of optima. When fitness is sampled from a large number of cases, then even knowing you have an optimum is problematic. Cases are a general notion and can be as simple as points in a data set being modeled or as complex as individual robot path planning problems. *Agent case embeddings (ACEs)* [7] are an analysis tool based on comparative performance of agents (typically alternative solutions) on various cases. They allow us to cluster solutions and deduce information about the geometry of the solution space, if not the actual geometry of the fitness landscape.

ACEs are a general purpose tool for exploring the variety of solutions produced by evolutionary algorithms, as well the geometry of the space of problems the evolving population is attempting to solve. Multiple instances of the problems solved by the evolved agents must be available or contrivable in order to use agent-case embeddings. The motivating example for ACEs, predating the name, appears in [12].

Definition 16.1. Given a set $\mathscr{A} = \{A_1, A_2, \cdots, A_n\}$ of n agents and a set $\mathscr{C} = \{C_1, C_2, \ldots C_m\}$ of m cases, an **agent-case embedding** is a simultaneous embedding of $\mathscr{A}$ into $\mathbb{R}^m$ and of $\mathscr{C}$ into $\mathbb{R}^n$ obtained in the following manner. Let $f_{C_j}(A_i)$ be the fitness or score that agent A_i obtains on case C_j. Then the **agent-case matrix** M is the $n \times m$ matrix where $M_{i,j} = f_{C_j}(A_i)$. The embedding in $\mathbb{R}^m$ of agent A_i is the point whose coordinates are the i^{th} row of the matrix. The embedding in $\mathbb{R}^n$ of case C_j is the point whose coordinates are the j^{th} column of the matrix.

The agent-case matrix is a book-keeping device which contains both embeddings of agents and cases as rows and columns respectively. If n agents and m cases are used, then the embedding of an agent is the vector in R^m whose coordinates are the m scores of the agent on the m cases, and the embedding of a case is the vector in $\mathbb{R}^n$ whose coordinates are the scores the n agents got on the specified case.

A key property of agent-case embeddings is that they ignore genotypic differences that do not impact phenotype. Therefore, agent-case embeddings are not useful when neutral genotypic differences are of interest. However, they can be used to compare solutions evolved with different representations. The behavior of agents on problem cases is the only information relevant for an agent-case embedding and so phenotypically similar individuals will have similar embeddings even when they have very different genotypes or are encoded differently.

As mentioned earlier, agent-case embeddings are only useful when a variety of problem cases are available or contrivable. For an example where contrivance is useful, see Section 16.7. It is worth noting that this is not a very limiting restriction, but may require some creativity, since problem cases may be contrived in a variety of ways. When competitive agent strategies for a game are evolved, a fixed set of opponent strategies can be used as the set of cases where the score an agent receives playing against the case strategy is the agent-case score, this is the *agent-as-case* version of agent-case embeddings [5, 6]. When evolutionary computation is used to optimize solutions to a particular fitness function, a variety of other fitness functions may be used as the set of cases. When some form of initial conditions are evaluated as part of the fitness of an evolved structure a set of alternate initial conditions may be used as the problem cases [9]. The distance between agents in the embedding will be based on the difference in the agents' behavior on the problem cases; care must be taken so that the agent-case embedding derived distance between agents and cases is relevant for the particular task.

16.3 Self Avoiding Walks

The SAW problem is a good framework for discussing fitness landscapes because it changes a great deal when we change the mutation operator or the representation or potential solutions. It uses a string of commands to direct moves on a grid. For an application of this problem to designing RFID tags, see [21]. Fitness is evaluated by starting in the lower left corner of the grid and then making the moves specified by the chromosome. The sequence of moves made is referred to as the *walk*. If a move is made that would cause the walk to leave the grid, then that move is ignored. The walk is allowed to revisit cells of the grid. Fitness is equal to the number of squares visited when the walk is completed. An optimal walk has fitness equal to the number of squares in the grid minus one. The problem is called the *self-avoiding* walk problem because optimal solutions do not revisit squares; they are self-avoiding walks. Three examples of self-avoiding walks appear in Figure 16.1.

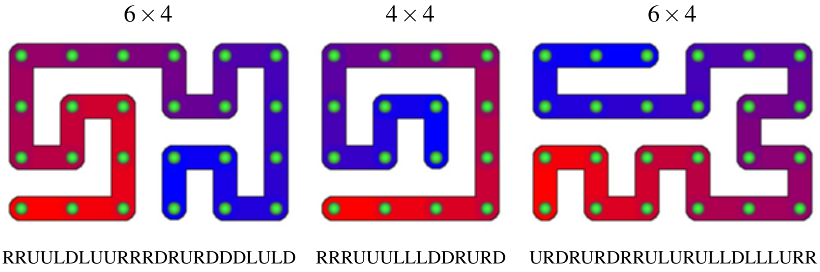


Fig. 16.1 Examples of optimal self avoiding walks and their genes in the direct representation

We now describe three different representations for the SAW problem together with their point mutation operators.

The **direct representation** is a string of appropriate length over the alphabet $\{U, D, L, R\}$. The gene is used directly to make moves on a test grid with fitness equal to the number of grid squares visited at least once. A point mutation consists of picking a position in the string and changing the character at that location.

The direct representation is the baseline. Most published work on the SAW problem [1, 2, 20] uses the direct representation. In [20] it was shown that the optimization behavior of the SAW problem changes substantially for different problem sizes (grid dimensions). The change in behavior is conjectured to be the result of a change in the ratio of local to global optima in the fitness landscape as the problem size changes.

The **state-conditioned** representation is a string of appropriate length over the alphabet $\{F, R, L\}$. In this case, interpretation of the walk requires state information of an agent performing the walk. The agent starts in the lower left corner facing to the right. The move F instructs the agent to move forward one square. The instruction R instructs the agent to turn right and then move forward one square. The instruction L similarly tells the agent to turn left and advance. The point mutation for this representation is, except for the different alphabet, the same as that for the direct representation.

The state-conditioned representation uses a smaller alphabet, giving it a much smaller fitness landscape, and also encodes an important piece of domain knowledge: going back to the square you just came from is a wasted move. This means that there are many local optima in the direct representation that cannot be encoded in the state-conditioned representation. We note that any *global* optima of the direct representation can be encoded in the state-conditioned representation. This means that using the state-conditioned representation sacrifices only local optima.

The **gene-expression representation** is a string of more than the appropriate length over the alphabet $\{U,D,L,R,u,d,l,r\}$. The representation has a parameter, e.g. 50% or 100%, that says how much longer than the number of required moves it is. The genetic loci beyond the minimum needed are called the *excess* of the representation. Upper case letters are treated as in the direct representation, while lower case letters are ignored (not expressed). In addition, if too many loci are expressed, those beyond the number of grids in the SAW less one are ignored: the representation never permits "extra" moves. The point mutation for this representation picks a random position and half the time changes the upper-lower case status of the letter at that position. The other half of the time it generates a new character uniformly at random.

The gene-expression representation has a far larger fitness landscape than the direct representation. It also encodes a larger number of phenotypes. Since a walk encoded in the gene-expression representation can have too few moves, this representation encodes all the full-length phenotypes from the direct representation as well as all too-short phenotypes. The ability to change between upper and lower case letters means that the gene-expression representation has the ability to insert and delete characters in its phenotype.

Fitness landscapes for the three representations are examined using a stochastic hill-climber. This search algorithm first generates a random gene and then performs 1,000,000 hill-climbing steps. A hill-climbing step consists of mutating the gene and retaining the mutation if it does not make the fitness worse. Table 16.1 shows the results of using the stochastic hill-climber on the 5×5 SAW problem using all three representations with the mutation consisting of one, two, or three point mutations. The hill-climber is run 1000 times for each representation and type of mutation. The number of global optima located are recorded. The large number of hill-climbing steps mean that the final gene has a very high probability of being an optimum.

Looking at Table 16.1 we see that the most efficient representation, with the smallest fitness landscape and incorporating domain-specific knowledge, exhibits the worst performance. In fact performance on the 5×5 SAW problem improves with the size of the fitness landscape and hence the size of the search space. This

Table 16.1 Number of times in 1000 trials with a stochastic hill-climber that a global optimum was located for the 5 × 5 SAW problem

Point	Representation			
Mutations	Direct	State	Expr. 50%	Expr. 100%
1	9	1	237	271
2	701	118	947	965
3	799	571	965	973

seems backwards, but is in fact easy to explain. The factor driving performance is the *neutral networks* possessed by each representation. Recall that a *neutral mutation* is one that does not change fitness and that a neutral network is the set of all genes reachable from a given gene by a series of neutral mutations.

The state-conditioned representation wastes the fewest genetic loci because it expresses all its loci and never moves back-and-forth. It thus has the smallest neutral networks. The gene-expression representation has huge neutral networks – any mutation of an unexpressed loci to another unexpressed loci is a neutral mutation. In addition, moves beyond those permitted are not used even when they are expressed. The larger the excess in the gene-expression representation, the more unused loci there are and hence the more neutral mutations. For all three mutation levels, the performance increases with the size of the neutral networks in the representation.

The second obvious pattern is that more point mutations are better, at least within the range of the experiments performed here. The explanation can be easily understood in fitness landscape terms. If we ignore neutral mutations, then a gene is on a hilltop if none of its mutants are better than it. When we use a mutation operator that changes more loci, then the number of genes one mutation away grows combinatorially. The effect of increasing the number of loci changed by the mutation operator is to merge local optima, either with one another or into global optima.

If we do not ignore neutrality, then the situation is more nuanced. Instead of having point optima, entire neutral networks become optima. At this point the definition of an optimum becomes the following. A neutral network is an optimum if no mutant of any member of the network has higher fitness. There are large neutral networks that are not optima. Demonstrating they are not optima can be tricky: there may only be a few members of the network that have mutants that are better. If we use probabilistic mutation, which has a positive probability of modifying each location, then, technically, there are no local optima and only one global optimum. This is because any gene may be reached in one mutation. In this situation, different definitions are required – one might call a gene a local optimum if the probability of improvement is less than some positive threshold.

We conclude by reiterating that the fitness landscape contains points defined by the representation and has a topology generated by the variation operators. The SAW experiments demonstrate that choice of representation and variation operator can have a large impact on the character of the fitness landscape and the resulting optimization performance.

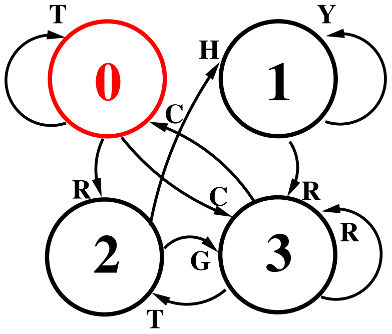
16.4 Side Effect Machines

Study of the evolution of side effect machines provides an example in which nonlinear projection and other multidimensional scaling techniques contribute to understanding the fitness landscape. Side effect machines (SEMs) create features for analysis of strings on a limited alphabet such as DNA sequences. They are a superset of k-mer features, features that count the number of occurrences in a string of a particular substring of length k, for example, the number of occurrences of the 3-mer "ATG" in a DNA sequence. They were first introduced in [11] and later used in [8, 14, 15, 26] for DNA sequence classification problems. They were used in [19] to classify words in an error correcting code and in [22] to find DNA sequence motifs. [16] provides a detailed analysis of their fitness landscapes.

A SEM is an augmented finite state machine. A finite state machine is a basic concept in the theory of computation that provides a mathematical model of computation. For more information, consult [27] or any similar textbook. A finite state machine consists of: a set of states, an alphabet, and a transition function. A start state and one or more accept states are designated. A string is passed through the machine, starting in the start state, transitioning to different states based on the transition function, and stopping when it transitions to an accept state or when the end of the string is reached. Its output is the accept state it reached or the fact that it did not reach an accept state. A SEM does not have accept states and augments each state with a counter. The counter keeps track of how many times the string passed through that state. The output of a SEM is the vector of values in the counters. Each counter is a SEM feature. An example of a SEM is shown in Figure 16.2 along with its representation for use in evolutionary computation.

The SEM search space is huge, growing super-exponentially with the number of states. There are n^{4n} n-state SEMs. This means there are, for example, 4.29 billion 4-state SEMs. Some sort of search algorithm is needed to select appropriate SEMs for a given problem. For DNA sequence classification problems, a genetic algorithm is used, with SEM fitness based on the accuracy of a classifier built from the features of the SEM using a given data set. Different classifiers have been tried, including k-means, k nearest neighbor, and random forest. Note that these fitness functions can produce fuzzy results, varying from evaluation to evaluation of the same SEM. The fitness landscapes are multi-modal. Diverse feature sets can be obtained by doing feature selection on the best SEMs from multiple evolutionary replicates. SEM features generated in this way produce gray-box classifiers that not only solve the classification problem, but also give biological insight into the sequences being classified.

The fitness landscape is, of course, different for every SEM classification problem. Visualizations of the landscapes can give insight into what type of search algorithm to use, how many optima there are, and how close together the optima are.



State	A	C	G	T
0	2	3	2	0
1	3	1	3	1
2	1	1	3	1
3	3	0	3	2

Fig. 16.2 On the left is an example of a SEM. To reduce clutter transitions are labeled using IUPAC codes, a 15-letter code to represent various combinations of DNA bases. R represents A or G; Y represents C or T; H represents A or C or T. The start state is shown in red. On the right is its representation for use in evolutionary computation. The SEM is stored as an 2D array with the rows representing states and the columns the DNA alphbet. The value in position (*i*,*j*) is the state to which to transition when base *j* is encountered in state *i*.

These visualizations are a guide to how long the algorithm needs to run and the quality and diversity of features to expect. Figure 16.3 shows multi-dimensional scalings of 500 randomly-generated 4-state SEMs as fitness landscapes explored by an algorithm using only mutation (no crossover) for three different DNA sequence classification problems. Details of these DNA sequence classification problems are not important for understanding the points made in this chapter, but interested readers can find them in [16]. The problem represented with blue figures is the IES problem in that paper; the problem represented in red is the RT problem; and the problem represented in green is the sLTR/SINE problem. The distance measure used is the mutation distance between the SEMs, the minimum number of mutations needed to transform one SEM into another. The points are colored based on their fitness, darker for better fitness. In all these landscapes, high and low fitness SEMs appear close together. This suggests that a single mutation can cause a large change in fitness. From this we know not to try gradient search techniques.

The red and the green fitness landscapes both have high quality optima (dots that appear black); fewer in the green than in the red landscape. The blue landscape has some good quality optima, but none as good as the best in the green and red landscapes. The best optima in the red landscape are scattered throughout the space, suggesting diversity, while those in the green landscape are more clustered.

SEM fitness landscapes are rich in neutral networks. SEMs with unused states have the same fitness as other SEMs with states identical to their used states, but different values for their unused states. A neutral network of a machine with one unused state has 4^n SEMs in it with each connected to $4n$ other SEMs. Figure 16.4 shows a multidimensional scaling of a part of the SEM fitness landscape that includes a neutral network created by an unused state. Larger or additional neutral networks can be created when states are technically used, but relative to a particular data set, are in practice unused. For example in an AT-rich organism a subsequence

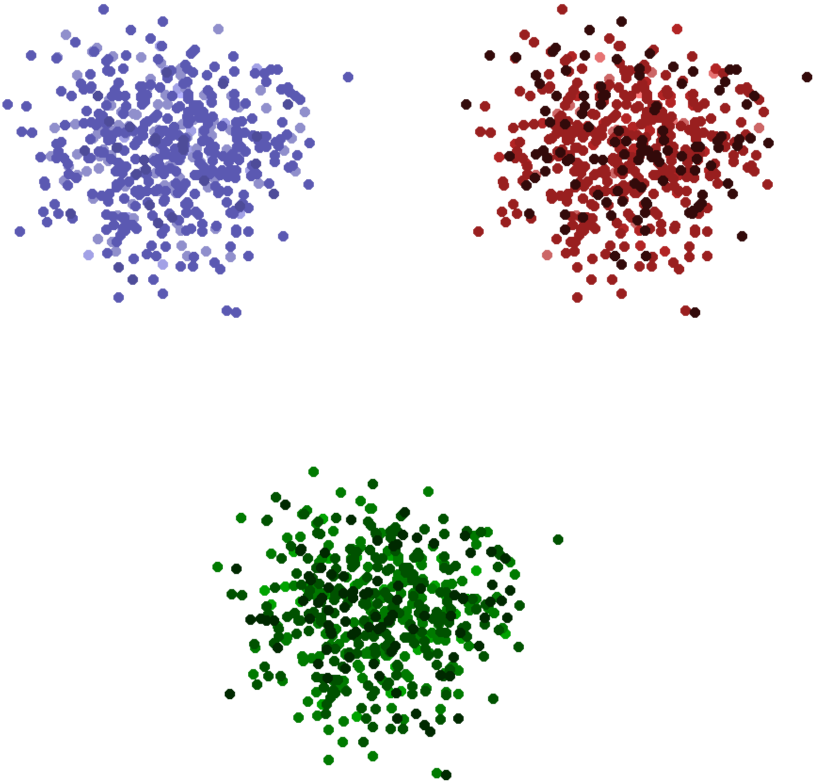


Fig. 16.3 Visualizations of samples of fitness landscapes based on multi-dimensional scaling for three different DNA sequence classification problems. Darker circles represent better fitness.

such as CGCGCGCGCG might never occur, so a state that could only be reached following such a sequence would be, in practice, unused. Also, in classifiers often features vary in importance. So, if a SEM contains several highly effective features, then the values of the other features do not matter much and mutations affecting them do not change the fitness of the SEM.

It is important to point out that the features created by SEMs separated by a large mutation distance are not necessarily qualitatively different. For example, if you take a SEM and renumber its states, you get another SEM that produces identical features. These two SEMs are far apart in terms of mutation distance, but close together in terms of useful behavior. For this reason, we also examine the phenotypic fitness landscape. This is a fitness landscape of SEM features created using correlation distance. Correlation distance is one minus the absolute value of the Pearson correlation of the features, based on their values for the data set in question.

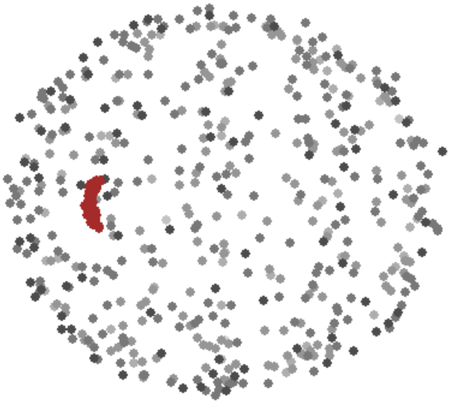


Fig. 16.4 Visualization of the SEM fitness landscape based on 500 randomly selected 4-state SEMs and 50 SEMs sampled from a neutral network created by an unused state. The random SEMs are colored based on fitness values for the green problem (darker circles represent SEMs with higher fitness) and the SEMs in the neutral network are brown.

Analysis of this fitness landscape also enables comparison of features generated in various different ways, like SEM features and *k*-mer features, and features generated to solve different problems.

Figure 16.5 shows the phenotypic fitness landscapes for the same three DNA sequence classification problems as in Figure 16.3. The circles represent high fitness SEM features found using 500 replicates of a genetic algorithm on 4-state SEMs. The diamonds represent the most effective 200 *k*-mer features for that problem, where $k \in \{1,2,3,4\}$. Examination of these phenotypic fitness landscapes gives us insight into the diversity of features found by the genetic algorithm as well as their similarity to *k*-mer features.

The evolved SEM features in the red fitness landscapes are similar to *k*-mer features. Also, there is a large group of features on the left side of the landscape, including both SEM features and *k*-mer features, that are similar to each other. The blue fitness landscape contains a more diverse collection of both SEM and *k*-mer features. There are SEM features that are very different from any *k*-mer feature and *k*-mer features that are very different from any SEM feature. The green fitness landscape shows an even greater disparity between the SEM and *k*-mer features. There are clumps of SEM features, suggesting that some features are found over and over again by the genetic algorithm. Comparison with the green fitness landscape in Figure 16.3 suggests that these are the highest quality features being found over and over again. If this phenotypic fitness landscape corresponded instead to the red fitness landscape in Figure 16.3, which has many high fitness points scattered throughout, it would instead suggest that the genetic algorithm was not efficiently searching the space.

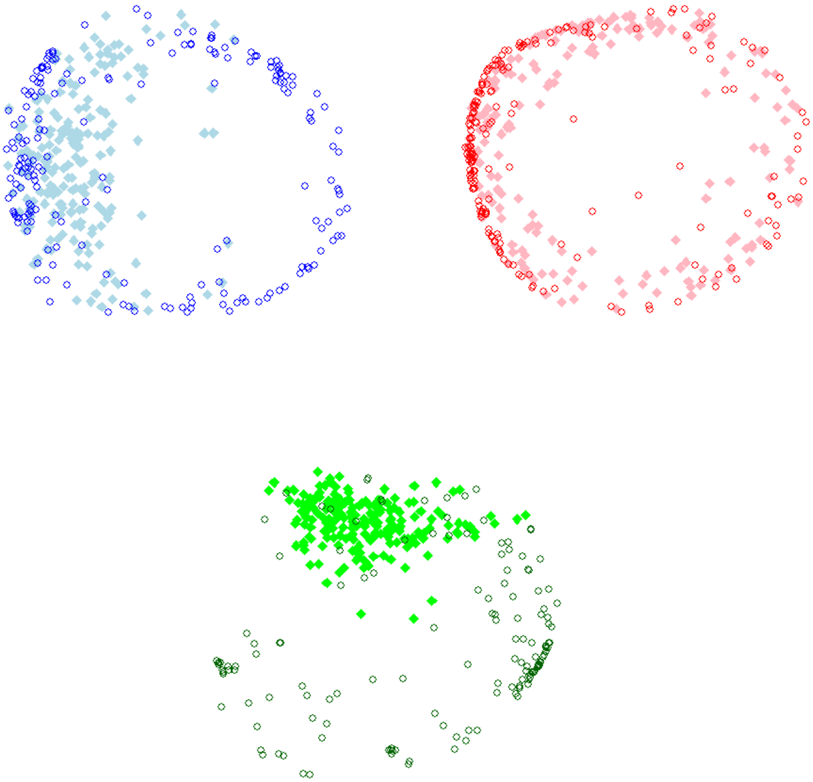


Fig. 16.5 Visualizations of phenotypic fitness landscapes based on multi-dimensional scaling for three different DNA sequence classification problems. The highest fitness SEM features found by a genetic algorithm are shown as circles; *k*-mer features are shown as diamonds.

Figure 16.6 shows a phenotypic fitness landscape for the best fitness SEM features found for each of the three DNA sequence classification problems. To create this figure, correlation distance was calculated using the three data sets combined. By looking at this figure, we can tell that the blue and red problems are related, with similar features performing well for both, while the green problem is distinct. This could be important in the biological interpretation of the features used in the resulting gray-box classifier.

This is the type of analysis that can be done with ACEs, which are applied more extensively to the analysis of the Tartarus problem in Section 16.5, the fractal real optimization problem in Section 16.6 and the apoptotic cellular automata problem in Section 16.7. In this case the SEMs represent agents and the data sets represent cases. Figure 16.6 compares the effective agents for the three different cases.

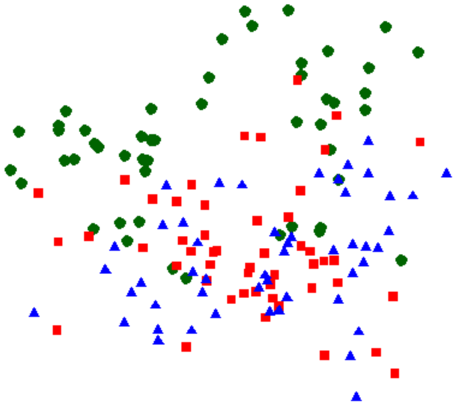


Fig. 16.6 Visualization of phenotypic fitness landscape containing SEM features evolved to solve three different DNA sequence classification problems

Figure 16.7 shows a fitness landscape containing the best fitness SEM features generated by genetic algorithms using four different fitness functions for the problem represented in blue in the other figures. This shows us the impact of the choice of fitness function. The fitness functions represented by the blue squares and the yellow triangles are locating similar features. The other two fitness functions are locating a comparatively more diverse set of features, and they are locating features significantly different from each other.

16.5 Tartarus

The Tartarus task is useful for demonstrating the utility of ACEs. It was defined in [29]. It takes place on a 6 × 6 grid bounded by walls. Figure 16.8 shows a valid initial state for the Tartarus task. Six blocks are placed so that they neither touch the walls nor form a close 2 × 2 group of blocks. The robot is given 80 moves in which it may advance, turn left, or turn right. Turning is done in place and so is always possible. The robot can push one box ahead of it but cannot push two boxes. The robot cannot advance through a wall nor can it push a box through a wall. After 80 moves the robot is rewarded for the number of box sides against a wall. A box in a corner is worth two points, and a box against a wall but not in a corner is worth one point. The maximum score for a board is thus 10. An agent-case embedding is developed by using 100 evolved robot controllers as agents and 100 randomly selected Tartarus boards as cases, with the agent-case score being the Tartarus score the agent receives on the case board.

The agents evolved for this demonstration use one of the simplest representations available, a string of 16 moves over the alphabet **F**orward, **L**eft, **R**ight. The robot

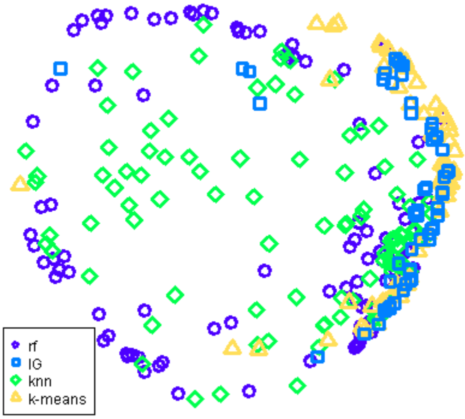


Fig. 16.7 Visualization of phenotypic fitness landscape containing SEM features evolved using four different fitness functions

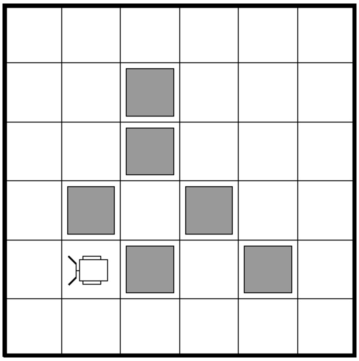


Fig. 16.8 A valid starting configuration for the Tartarus task

loops through the string five times to generate 80 moves. In [1] is it found that the shorter, looped string representation yields better solutions than a full 80-character representation. The algorithm used is a standard evolutionary algorithm with a population of 120 strings operated on with size four single tournament selection using two point crossover and 1-3 mutations per new structure. The number of mutations is selected uniformly at random and the operator replaces a character in the string with a new one. The algorithm is run for 200 generations.

A collection of 100 runs are performed and the resulting best-of-run agents are used to generate the ACE. Figure 16.9 replicates, for a different representation, an observation appearing in [12]. The ACE permits us to establish that there are no "typical" Tartarus boards by plotting the distribution of the distance of embeddings of sampled boards from the mean board embedding. The lack of boards at the mean position means that all boards have agents that perform both well and poorly on that board. This means that the fitness landscape for the robots is shaped by a tradeoff surface where skill at some boards is bought at the cost of skill on others.

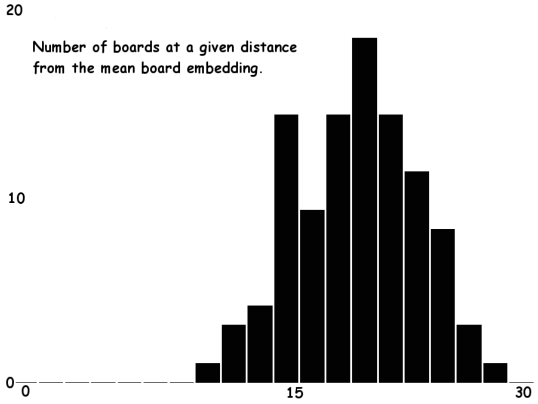


Fig. 16.9 This histogram bins 100 cases (Tartarus board) showing the distance of their embeddings from the mean position of all 100 embeddings

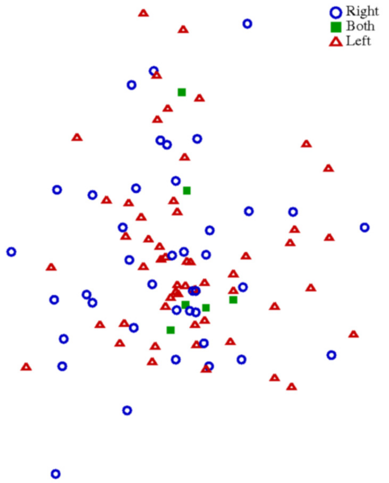


Fig. 16.10 Nonlinear projection of an ACE for Tartarus agents categorized as having an excess of left or right turns, or neither

When using the string based representation, evolution is likely to select control strategies with an excess of one type of turn, right or left. The 100 16-character strong controllers were categorized as having less than 40% left turns (category: right), 40%-60% left turns (category: middle), and more than 60% left turns (category: left). The same ACE used to produce Figure 16.9 was subjected to nonlinear projection. The result is shown in Figure 16.10.

This figure shows that relatively few robots (six) are in the middle category. Random expectation would place twenty in this category. The spatially mixed pattern of the robots in the left and right categories suggest that being left- or right-favoring does not grant a particular advantage with one or another subset of the 100 Tartarus boards used to generate the ACE. We expected clustering in accordance with the left-right categories, so this result was a surprise. The distribution does exhibit some clustering of cases, consistent with the idea that the ACE can be used to locate optima even when evolution is using samples to estimate fitness. A substantially larger set of samples would be needed to clearly establish this.

16.6 Fractal Real Optimization

Real optimization is perhaps the best studied subjects in computational intelligence and the fitness landscapes for real optimization are the best understood. We present a novel real optimization problem, that of locating views in the Mandelbrot set [4]. This is used to demonstrate the utility of fitness morphs in analysis. A rendering of the Mandelbrot set, showing the location of view located by optimization, appears in Figure 16.11.

Definition 16.2. The **Mandelbrot series** for a complex number z is given by:

$$z_0 = z \tag{16.1}$$

$$z_{n+1} = z_n^2 + z \tag{16.2}$$

Definition 16.3. The **divergence number** for a complex number z is the index n of the first member of its Mandelbrot series to have $|z_n| \geq 2$. If no such number n exists, the divergence number is infinite.

A *view* in the Mandelbrot set is a square in the complex plane defined by three real parameters (x,y,s). The complex number $z = x + yi$ is the upper left corner of the square and s is the log of the side length of the square. Several examples of views are shown in Figure 16.12. Views are located by optimization with a novel fractal representation [10] with a fitness function based on a *desired appearance mask*.

The *mask fitness function* used to locate views is defined as follows. Start with the hill-shaped function

$$f(x,y) = \frac{1}{x^2+y^2+1} \quad -3 \leq x,y, \leq 3 \tag{16.3}$$

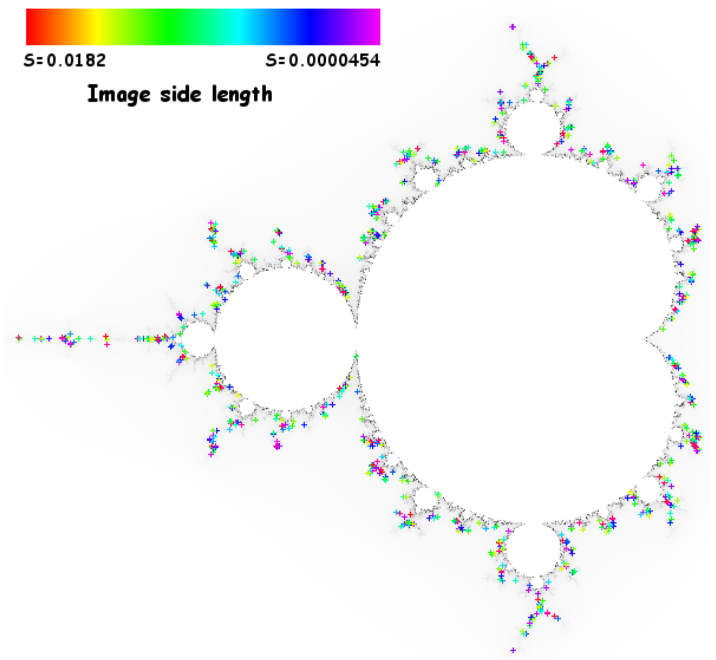


Fig. 16.11 Positions of 1000 high fitness views located in the Mandelbrot set

and extract a regularly spaced 11×11 grid of values within the square of definition. These values form the desired appearance mask. Fitness is the mean squared error (MSE) of these values with normalized divergence numbers for an analogous grid of points on a view. Divergence numbers are normalized by stopping computation of the divergence number at $L = 400$, if they are not smaller than that, and then dividing the divergence number by L. The fitness landscape of this fitness function is a version of the function $f(z) =$ divergence number of z, smoothed by use of the mask. This mask is good at locating *minibrots*, smaller copies of the Mandelbrot set which appear within the set.

The views shown in Figure 16.12 are enumerated with the multi-optima Sierpinski Searcher (MOSS) [3]. Figure 16.13 shows fitness morphs from this space. These morphs are graphs of the mask fitness function over one-dimensional transects of a three-dimensional space of views. The transect goes not only from one position in the Mandelbrot set to another but also changes the depth of zoom into the set log-linearly from the scale of the first view to the second.

Keeping in mind that lower fitness values are better, the fitness morphs in Figure 16.13 show a collection of different behaviors. Morph 1 shows a large number of intermediate optima between its endpoints; morph 2 is similar but without the spikes of very bad fitness visible in morph 1. Morph 3 shows large scale features of the landscape including a plateau of bad fitness between the values of 0.6 and 0.8 for

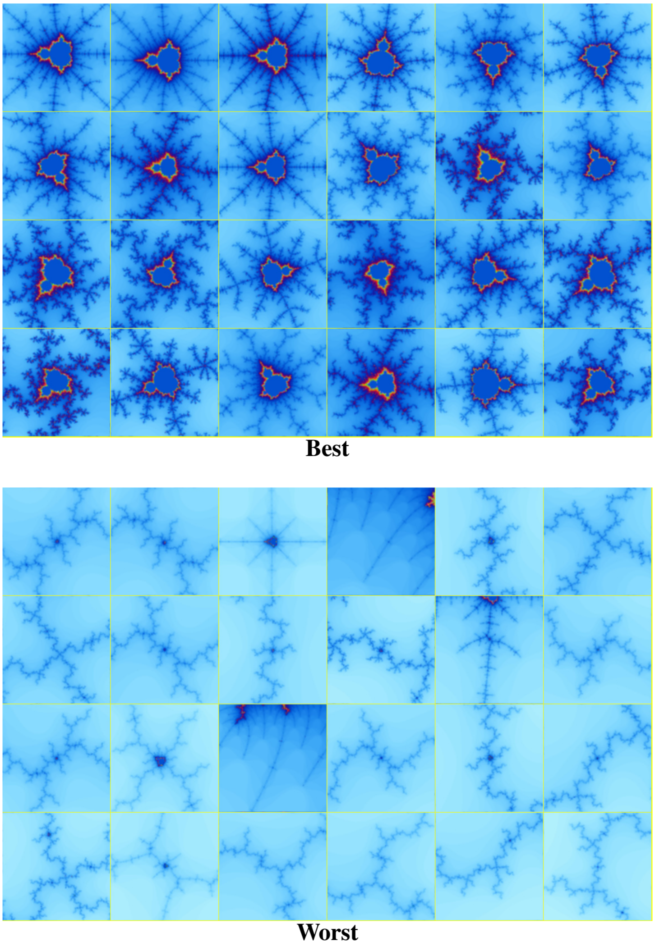


Fig. 16.12 The best and worst 24 minibrots located using the MOSS algorithm. These views are from a collection of 1000 distinct optima sorted by fitness.

the morphing parameter and large regions of similar fitness. Morph 4 crosses the interior of the Mandelbrot set, explaining the uniformly bad fitness away from the endpoints. Morph 5 brushes three limbs of the interior of the Mandelbrot set. Morph 6 is the most boring, displaying a low hill of inferior fitness.

Each of the downward spikes in the fitness morphs represents a region of the Mandelbrot set in which there is variation in the divergence numbers. Because the mask is searching for a view with high divergence numbers in the middle and low divergence numbers at the edges, both views in the interior of the Mandelbrot set *and* views well outside the set receive bad fitness values. The fitness morphs permit precise location of areas with variable divergence numbers and hence complex, interesting views.

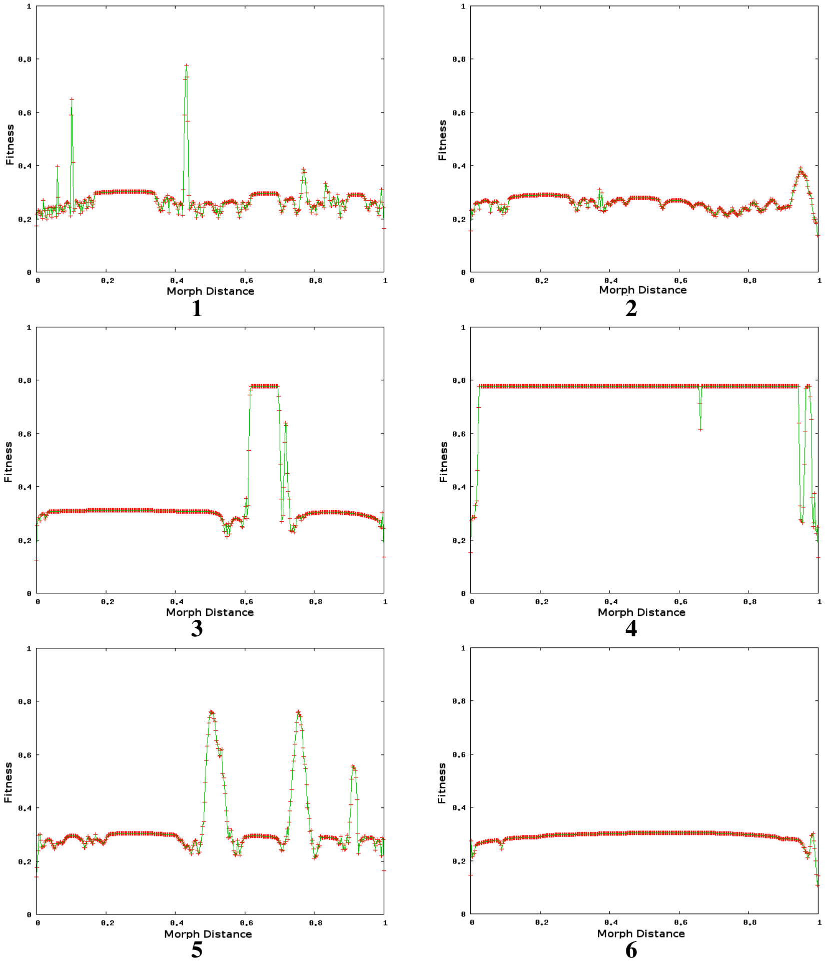


Fig. 16.13 Six example fitness morphs between pairs of optima of the mask fitness function on views in the Mandelbrot set. Each morph samples the fitness at 400 locations between the optima defining its endpoints.

16.7 Apoptotic Cellular Automata

Fitness morphs for a real function of real parameters are computed by performing repeated fitness evaluations along a shortest path from one optimum to another. The cellular automata described in this section are represented as arrays of integer values forming a lookup table for the automata's updating rule. The "path" from one rule to the other is not well defined. In this section we will show how to create morphs for this type of discrete representation by sampling many shortest paths between a pair of optima. We also show how to use ACEs to visualize the fitness landscape for cellular automata. Cellular automata have three parts:

(1) A collection of cells divided into neighborhoods of each cell,
(2) A set of states that cells can have,
(3) A rule that maps possible assignments of cell states of a neighborhood of a given cell c to a new state for c.

The automata used here are one-dimensional, an array of 401 cells in a line that wraps so that the first and last cells are adjacent. The neighborhood of each cell is the cell itself together with the two closest neighbors to the left and right. The possible cell states are the numbers 0-7. Updating rules are specified as lookup tables. The value of the cell states in a neighborhood are summed, yielding a number in the range 0-35. The lookup table is an array of 36 values in the range 0-7. The updated value of a given cell is the value of the lookup table indexed by the sum of the cell values in that cell's neighborhood. The cell state 0 is defined to be the *dead* state and we require that updating rules map an all-zero neighborhood to 0. This prevents spontaneous generation of living cells in a dead area.

Updating of the automata described here is *synchronous*. New state for all cells are computed and then they are simultaneously updated to their new values. The initial state of the automata consists of setting all but the three cells in the center of the cell array to 0. The middle cells are set to (1)(2)(1). The *time history* of a one-dimensional cellular automata is the sequence of cell states of the automata. Each synchronous updating is a time step. Examples of time histories of automata are given in Figure 16.14.

The fitness function used to evolve automata rules rewards the automata for growing as much as possible so long as they die before a particular time step. Since the death is programmed into the rule, we call such cellular automata *apoptotic* after the biological term for programmed cell death.

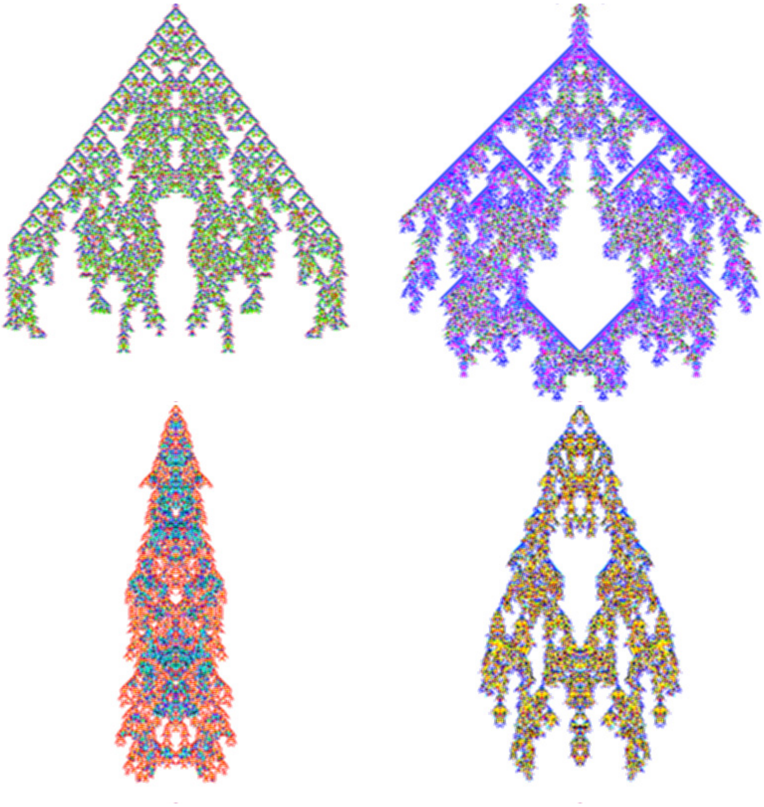
Definition 16.4. Apoptotic fitness is zero if the time history contains any living cells after the 401st time step. If no live cells remain in the final step, then the fitness is the number of live cells in the time history from time zero to the next to last step. We call an updating rule **apoptotic** if it has a nonzero apoptotic fitness.

We now describe the procedure for constructing fitness morphs for CA-rules. Each rule is a potential solution to the apoptotic rule location problem.

Definition 16.5. If σ is an order on the 36 positions in a CA-rule, then σ defines a shortest path between any two rules as follows. One end of the path is the first rule. The values in this rule are changed to the values in the second rule in the order given by σ. This is the σ-path between the rules.

If two rules differ in k positions then there are $k!$ different σ-paths between them. In order to get a fitness morph we sample 10,000 σ-paths and compute the mean and maximum fitness at each step. All possible fitness morphs for the four example automata given in Figure 16.14 are given in Figure 16.15.

The fitness morphs shown in Figure 16.15 tell us something interesting. The average fitness remains well above 10.0 for all six morphs. The average fitness of 1,000,000 randomly generated CA-rules is 0.0319. This means that the paths between optima are consistently well above random fitness. Similarly the maximum

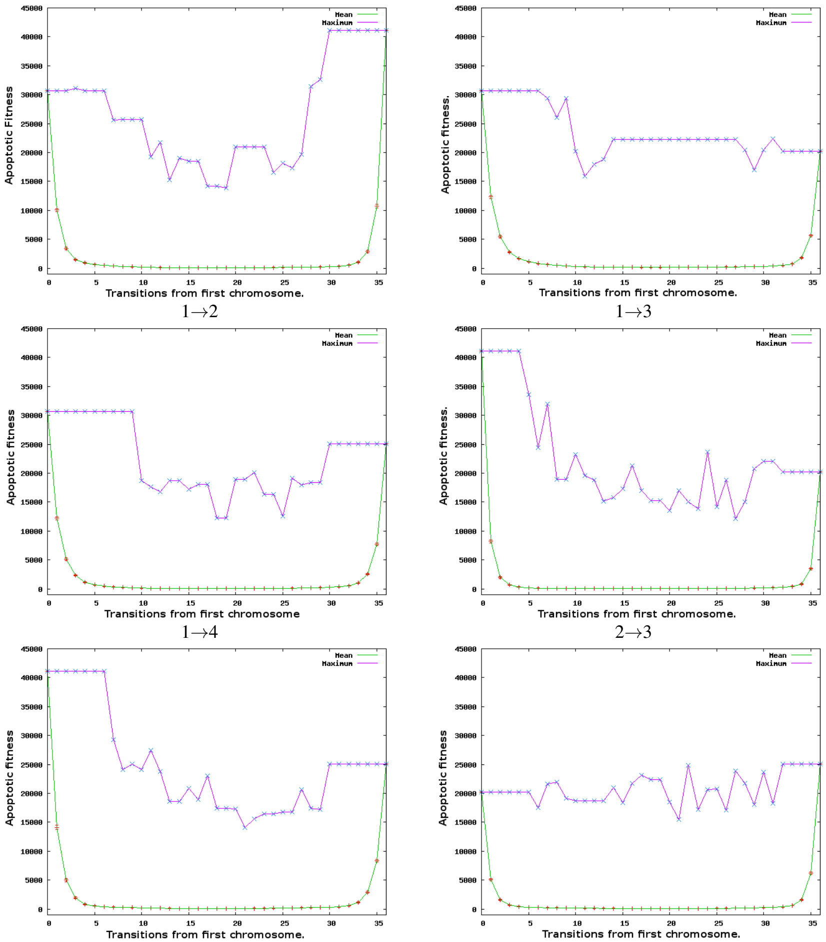


Rules:
0010220330105341070031023636354671 77 0300210304274060034045612524524230 67
0010501150102261436610667702467653 62 0200200060212571550055734122754266 21

Fig. 16.14 Time histories for four apoptotic cellular automata. Rules and automata are displayed in the same order.

fitness on sampled paths between optima stays well above 10,000. Taken together these observations suggest that the high-fitness rules are packed into a small part of the space. In [7] this hypothesis is explored more carefully and it is found that the apoptotic rules are indeed dense in a very small part of the rule space. This means the fitness landscape is a complex multi-level mesa in the middle of a very large flat plain.

It is also worth noting that many of the rules never use several of the positions in their rule. Among 390 high-fitness rules examined in detail in [9] (the standard apoptotic data set of SADS) some used only 20 of the 36 positions. Since changing the value of an unused position is a neutral mutation, this means that there are $8^{16} = 2.81E14$ members of the neutral network for those rules. Low fitness, apoptotic rules can easily use fewer positions in their rules giving them titanic neutral networks. In spite of the size of these neutral networks, direct comparison of the 390 rules in



1→2

1→3

1→4

2→3

2→4

3→4

Fig. 16.15 Six example discrete fitness morphs between pairs of optima of the apoptotic fitness function for apoptotic cellular automata. Each morph is the average of 10000 sample shortest paths. The endpoints are the rules for the automata shown in Figure 16.14.

[9] found zero duplicates. In the time since, tens of thousands of apoptotic rules representing different optima have been located, no two of which are members of the same neutral network. This means the fitness landscape has a remarkable number of hills.

Suppose that we characterize the neutral networks in the fitness landscape as “countries” and say that two countries share a border if a point in one can be transfered into a point in another by a single application of the mutation operator. Then what the fitness morphs tell us is that the apoptotic rules form a spike rising out of a flat plain of zero fitness. The spike has the structure of a complex mesa in which there are smaller and larger flat spaces with cliffs for borders. These cliffs result from the fact that fitness takes on only integer values and, as looking at traces of maximum fitness over the course of evolution [9] tells us, these cliffs are often quite high – hundreds of fitness units. The complex character of the fitness landscape for apoptotic cellular automata, with its large flat spaces and complex connectivity, prompted its inclusion here.

16.7.1 ACEs for Cellular Automata

In constructing an ACE for cellular automata rules we place the rules in the role of agents. While the rules were evolved with an initial condition of 0 save for a (1)(2)(1) in the center of the cell array, they *can* be run with any boundary condition. Given that we have 8 cell states, there are 63 initial conditions that replace (1)(2)(1) with $(a)(b)(a)$ where not both of a and b are 0. We treat these 63 initial conditions as cases and construct the ACE in the usual manner, with the agent-case score being the apoptotic fitness of the agent using the initial conditions of the case. Each rule is now mapped to a vector of 63 integer values. A nonlinear projection of this ACE is shown in Figure 16.16.

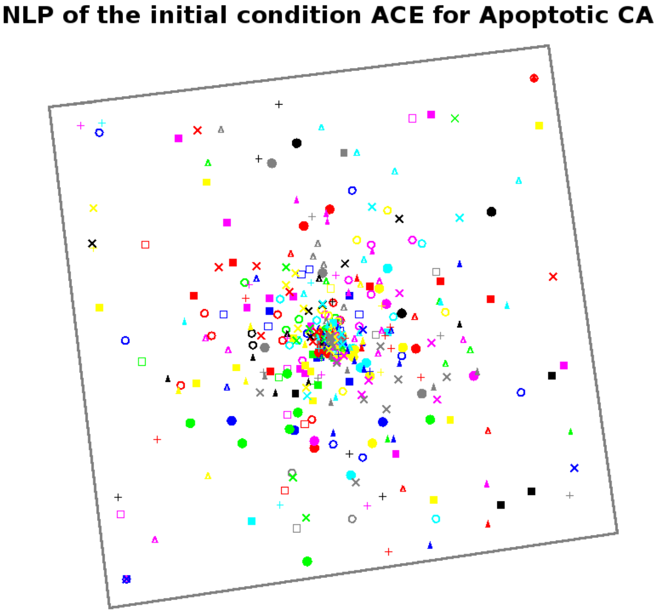


Fig. 16.16 A nonlinear projection of the ACE for cellular automata rules. A set of 390 rules appear in the projection. Glyph types are for thirteen experiments varying population size and mutation from [9]. A gray outline was added to make the square of variation more visible.

The nonlinear projection treats the rules from the SADS as points, one rule from each of 30 replicates in each of thirteen experiments. The thirteen experiments formed a parameter study in [9]. The main result that the NLP supports is this: the algorithm is quite robust to the choice of population size and variation operator. This result makes sense in terms of what we know about the fitness landscape – there is relatively little gradient information and there are enormous flat spaces. This means that the high fitness individuals in a population at a given time will probably be descended from a single population member that made a large jump up a fitness cliff in the recent past. The neutral networks (flat spaces) act to level the playing field between more and less aggressive mutation operators. Within a neutral network, evolution engages in random walks until if finds a cliff. More aggressive mutation operators random walk faster, but they also walk off the (more common) downward cliffs more often.

The central distribution of points in the NLP, as well as its square shape, can be explained by looking at individual coordinates of the ACE. There are two pairs of sets of initial conditions for which many rules score high in one but not the other. This creates two large sources of variation in four directions. The nonlinear projection maps the pair variations in the two pairs of initial conditions onto two orthogonal axes. The scatter in these two coordinates of the ACE yields the square. The centrality comes from the fact that a majority of rules have high fitness only for the initial conditions under which they were evolved. The ACE also demonstrates another sense in which the rules are robust. Their fitness is well above the random background for many initial conditions.

16.8 Conclusions

Fitness landscapes come in many shapes and sizes. A landscape can be as simple as a single hill, like that provided by the onemax problem [1] or as complex as the sampled interaction between agents and boards in Tartarus. When agents compete against other agents, as in algorithms training game playing agents, we can even achieve time-varying landscapes, an interesting topic beyond the scope of this chapter.

The SAW problem gave us an example of a family of rugged fitness landscapes for a simple problem, solvable by hand. We also saw that changing any of the components of the fitness landscape (representation or variation operators) can have a dramatic effect on its shape and character. This means that paying close attention to representation and variation operators is likely to pay benefits. This is a consequence of the famous no free lunch theorem [32] that, among other things, implies that using a generic optimizer is not the best strategy.

As we showed with the SEM example, it is beneficial to look at both the phenotypic fitness landscape as well as the genotypic landscape. The two may be quite distinct and offer very different perspectives on the optimization task of interest. Both give insight into the number and variety of optima, and the phenotypic fitness landscape provides a visual comparison with features created using other methods.

Both the fitness morph and the NLP methods are useful for understanding the nature of the more opaque high dimensional landscapes. Each techniques is applicable whenever you are able to calculate a distance between either the phenotypes or genotypes of potential solutions. They provide contrasting views of the same high-dimensional space and so using both can provide insight not otherwise available. The cellular automata example shows it is possible to create a meaningful distance measure even when dealing with complex discrete spaces. Here, by using initial conditions as additional problem cases, an ACE was constructed when a natural one was not available inside the original problem description.

When dealing with a tightly related class of problems such as Tartarus boards or apoptotic cellular automata, examining a single instance of a problem is insufficient to understand the landscape. Approaches for understanding multiple problem cases together, such as ACEs, are valuable. In particular ACEs permit the examination not only of the landscape of solution but the landscape of problems. This permits us to locate pathological instances of the problem as well as grouping similar problem instances.

As a whole, the diverse examples in this chapter suggest a vast array of problems for which evolutionary computation is an appropriate tool, and beyond that a vast array of representations and variation operators unexplored potential. We are just beginning to analyze the space of fitness landscapes, and it is strange indeed.

References

[1] Ashlock, D.: Evolutionary Computation for Optimization and Modeling. Springer, New York (2006)

[2] Ashlock, D., Bryden, K.M., Corns, S., Schonfeld, J.: An updated taxonomy of evolutionary computation problems using graph-based evolutionary algorithms. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 403–410. IEEE Press, Piscataway (2006)

[3] Ashlock, D., Bryden, K.M., Gent, S.: Multiscale feature location with a fractal representation. Intelligent Engineering Systems Through Artificial Neural Networks 19, 173–180 (2009)

[4] Ashlock, D., Jamieson, B.: Evolutionary computation to search mandelbrot sets for aesthetic images. Journal of Mathematics and Art 1(3), 147–158 (2008)

[5] Ashlock, D., Kuusela, C., Rogers, N.: Hormonal systems for prisoners dilemma agents. In: Proceedings of the 2011 Conference on Computational Intelligence in Games, pp. 63–70 (2011)

[6] Ashlock, D., Kuusela, C., Cojocaru, M.: Shopkeeper strategies in the iterated prisoner's dilemma. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 1063–1070. IEEE Press, Piscataway (2011)

[7] Ashlock, D., Lee, C.: Agent-case embeddings for the analysis of evolved systems. IEEE Transactions on Evolutionary Computation 17(2), 227–240 (2013)

[8] Ashlock, D., McEachern, A.: Ring optimization of side effect machines. Intelligent Engineering Systems Through Artificial Neural Networks 19, 165–172 (2009)

[9] Ashlock, D., McNicholas, S.: Fitness landscapes of evolved cellular automata. IEEE Transactions on Evolutionary Computation 17(2), 198–212 (2013)

[10] Ashlock, D., Schonfeld, J.: The Sierpinski representation for real optimization. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 87–94. IEEE Press, Piscataway (2007)

[11] Ashlock, D., Warner, E.: Classifying synthetic and biological DNA sequences with side effect machines. In: Proceedings of the 2008 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 22–29 (2008)

[12] Ashlock, D., Warner, E.: The geometry of tartarus fitness cases. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 1309–1316. IEEE Press, Piscataway (2008)

[13] Ashlock, D., Schonfeld, J.: Nonlinear projection for the display of high dimensional distance data. In: Proceedings of IEEE Congress on Evolutionary Computation, vol. 3, pp. 2776–2783. IEEE Press, Piscataway (2005)

[14] Ashlock, W., Datta, S.: Detecting retroviruses using reading frame information and side effect machines. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–8 (2010)

[15] Ashlock, W., Datta, S.: Distinguishing endogenous retroviral LTRs from SINE elements using features extracted from evolved side effect machines. IEEE/ACM Transactions on Computational Biology and Bioinformatics 9(6), 1676–1689 (2012)

[16] Ashlock, W., Datta, S.: Evolved features for DNA sequence classification and their fitness landscapes. IEEE Transactions on Evolutionary Computation 17(2), 185–197 (2013)

[17] Borenstein, Y., Poli, R.: Decomposition of fitness functions in random heuristic search. In: Stephens, C.R., Toussaint, M., Whitley, L.D., Stadler, P.F. (eds.) FOGA 2007. LNCS, vol. 4436, pp. 123–137. Springer, Heidelberg (2007)

[18] Borg, I., Groenen, P.: Modern Multidimensional Scaling: Theory and Applications, 2nd edn. Springer (2005)

[19] Brown, J., Houghten, S., Ashlock, D.: Side effect machines for quaternary edit metric decoding. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–8 (2010)

[20] Bryden, K.M., Ashlock, D.A., Corns, S., Willson, S.J.: Graph based evolutionary algorithms. IEEE Transactions on Evolutionary Computation 10, 550–567 (2006)

[21] Montgomery, J., Randall, M., Lewis, A.A.: Differential evolution for rfid antenna design: a comparison with ant colony optimisation. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011, pp. 673–680. ACM, New York (2011)

[22] Noori, F.A., Houghten, S.: A multi-objective genetic algorithm with side effect machines for motif discovery. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–8 (2012)

[23] Oksanen, J., Guillaume Blanchet, F., Kindt, R., Legendre, P., Minchin, P.R., O'Hara, R.B., Simpson, G.L., Solymos, P., Stevens, M.H.H., Wagner, H.: vegan: Community Ecology Package. R package version 2.0-1 (2011)

[24] Pitzer, E., Affenzeller, M.: A comprehensive survey on fitness landscape analysis. In: Fodor, J., Klempous, R., Suárez Araujo, C.P. (eds.) Recent Advances in Intelligent Engineering Systems. SCI, vol. 378, pp. 161–191. Springer, Heidelberg (2012)

[25] R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2011)

[26] Schonfeld, J., Ashlock, D.: Classifying Cytochrome c Oxidase subunit 1 by translation initiation mechanism using side effect machines. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–7 (2010)

[27] Sipser, M.: Introduction to the Theory of Computation, 2nd edn. Thomson (2006)

[28] Stadler, P.F.: Linear operators on correlated landscapes. Journal De Physique I 4, 681–696 (1994)

[29] Teller, A.: The evolution of mental models. In: Kinnear, K. (ed.) Advances in Genetic Programming, ch. 9. The MIT Press (1994)

[30] Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S, 4th edn. Springer, New York (2002)

[31] Weinberger, E.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cybernetics 63(5), 325–336 (1990)

[32] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)

[33] Wright, S.: The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In: Proceedings of the Sixth International Congress on Genetics, pp. 355–366 (1932)

Chapter 17
Visual Analysis of Discrete Particle Swarm Optimization Using Fitness Landscapes

Sebastian Volke, Simon Bin, Dirk Zeckzer, Martin Middendorf, and Gerik Scheuermann

Abstract. Particle swarm optimization (PSO) is a metaheuristic where a swarm of particles moves within a search space in order to find an optimal solution. PSO has been applied to continuous and combinatorial optimization problems in various application areas. As is typical for metaheuristics, it is also not easy for PSO for algorithm designers to understand in detail how and why changes in the design of a PSO algorithm influence its optimization behavior. It is shown in this chapter that a suitable visualization of the optimization process can be very helpful for understanding the optimization behavior of PSO algorithms. In particular, it is explained how the visualization tool dPSO-Vis can be used to analyze the optimization behavior of PSO algorithms. The two example PSO algorithms that are used are the SetPSO and the HelixPSO. Both algorithms can be used for solving the RNA secondary structure prediction problem.

17.1 Introduction

Many optimization problems that occur in science or in economical applications can be modeled as combinatorial optimization problems. Unfortunately, it is often very hard to find optimal solutions to such problems. Therefore different types of heuristics are used to solve such problems. Metaheurstics, e.g., genetic algorithms, ant colony optimization, and particle swarm optimization (PSO), have the

Sebastian Volke · Simon Bin · Martin Middendorf · Gerik Scheuermann
Department of Computer Science, University of Leipzig, Augustusplatz 10,
04109 Leipzig, Germany
e-mail: {volke,sbin,middendorf,
scheuermann}@informatik.uni-leipzig.de

Dirk Zeckzer
Department of Computer Science, University of Kaiserslautern, Postfach 3049,
67653 Kaiserslautern, Germany
e-mail: zeckzer@informatik.uni-kl.de

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_17, © Springer-Verlag Berlin Heidelberg 2014

advantage that they can be applied to different combinatorial optimization problems. A common principle that is employed by several metaheuristics is to keep a population of good solutions and use it to create new and hopefully better solutions. A metaheuristic usually has several parameters that influence its optimization behavior, e.g., the size of the population. The best parameter values for a metaheuristic depend on the problem instance. A general problem with metaheuristics is that it is very difficult to analyze their optimization behavior. Thus, it is difficult to understand why some metaheuristics work better than others on certain problems and how this is influenced by the different types of the optimization problems.

In this chapter, we show that a suitable visualization of the optimization process can be very helpful for gaining a better understanding of metaheuristics. In particular, we consider PSO algorithms for discrete optimization problems using a visualization tool – called dPSO-Vis – that has been presented very recently for such algorithms [15]. The dPSO-Vis visualization tool leverages the fact that the movement behavior of the particles in a discrete PSO algorithm defines a fitness landscape on the set of possible solutions. The idea is then to visualize the fitness landscape topologically and show the particles within that landscape.

In this chapter, dPSO-Vis is used to visualize the optimization behavior of two PSO algorithms, namely SetPSO [12] and HelixPSO [6, 7, 8], for solving the RNA secondary structure prediction problem.

17.2 Discrete Optimization Problems and RNA Folding

Different types of combinatorial optimization problems occur in various application domains, e.g., assignment problems, routing problems, subset problems, or permutation problems. Often, it is possible to exploit special properties of the particular problem to find solutions more efficiently. But nevertheless, many of these problems are NP-hard, which means that it is unlikely that an optimal solution can be found in polynomial time [5].

Formally, a combinatorial optimization problem consists of a finite set of configurations X and a cost function $f : X \to \mathbb{R}$ that assigns a real value to each configuration. The task is to find an optimal solution, i.e., a configuration that minimizes the cost function (resp. maximizes the fitness function). For some combinatorial optimization problems a neighborhood relation $\chi \subset X \times X$ exists. In any case, the heuristic algorithm has to define such a relation in order to operate. We will later use this fact for the analysis of such algorithms. This neighborhood relation associates each configuration $c \in X$ with a set of neighbors $N(c) = \{x \in X : (c,x) \in \chi\}$. Thus, we have a configuration space (X,χ). Within this space, concepts such as locality exist, which can also be transferred to the cost function f.

In this chapter the RNA folding problem is considered as an example problem. In this problem, an RNA molecule is given that can be described as a sequence of four different types of nucleotides. Each type of nucleotide is characterized by one of the bases adenine (A), guanine (G), cytosine (C), or uracil (U). The 3-dimensional structure of an RNA molecule depends on hydrogen bonds that are formed between some pairs of nucleotides that are not neighbors in the sequence. A hydrogen bond can occur only between G and C, or A and U, or G and U. Two such connected nucleotides are called a *base pair*. There are a few additional restrictions that a set of base pairs has to conform to in order to be feasible (for details see, e.g., [8]). The RNA folding problem is to find a feasible set of base pairs, i.e., a secondary structure, with maximum free energy (with respect to some energy model, for details see [8]). Both algorithms that are considered in this chapter use helices as the elements that define a secondary structure where a (maximal) helix is a (maximal) set of two or more adjacent base pairs that form a ladder-like structure. Thus, a secondary structure is a set of helices that determines a feasible set of base pairs.

The RNA folding problem is a subset problem. For this type of combinatorial optimization problem, a set of objects S and a conflict relation $C \subset S \times S$ that lists mutually exclusive elements of S are given. The set of configurations is the set of all feasible subsets of S, i.e., $X = \{\hat{S} \subset S | \forall x, y \in \hat{S} : (x,y) \notin C\}$. There is a cost value associated with each such subset, and the task is to find the optimal subset (i.e., one with minimal total costs in case of minimization problems). In the case of RNA folding, S is the set of (maximal) helices and the conflict relation C contains pairs of helices that cannot occur within the same secondary structure.

Table 17.1 lists the nucleotide sequences of three example RNAs that are used throughout this chapter. Table 17.2 additionally lists the number of helices and how many admissible subsets are possible.

Table 17.1 RNA molecules from different species with their nucleotide sequence

Name	Sequence
Bacillus anthracis str.	UCCGCAGUAGCUCAGUGGUAGAGCUAUCGGCUGUUAACCGAUCGGUCGUAGGUUCGAGUCCUACCUGCKGAGCCA
Aeropyrum pernix	GGGCCCGUAGCUCAGCCAGGACAGAGCGCCGGCCUUCUAAGCCGGUGGUCCCGGGUUCAAAUCCCGGCGGGCCCGCC
Toxoplasma gondii	AGCACCGUAGCUCAGUGGGAGAGUGGGGGGCUCAUAACCCCCAGGUCCAUGGAUCGAAACCAUGCGGUGCUA

Table 17.2 RNA molecules from different species with their number of helices #H and their number of admissible subsets of helices #A

Name	#H	#A
Bacillus anthracis str.	35	1162
Aeropyrum pernix	60	7876
Toxoplasma gondii	39	1828

17.3 Particle Swarm Optimization

The particle swarm optimization (PSO) metaheuristic was introduced by Kennedy and Eberhart [10] and was inspired by the behavior of a flock of birds when searching for food. Originally, PSO was used to solve continuous optimization problems but meanwhile it has also been applied to various discrete optimization problems. The basic idea is to simulate a swarm of n particles moving within a search space in order to find an optimum. At each time step t, the position $\mathbf{p}_i(t)$ of particle i is updated:

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{v}_i(t+1)$$

where $\mathbf{v}_i(t)$ is the velocity of i. The interesting part is the velocity update of particle i,

$$\begin{aligned} \mathbf{v}_i(t+1) =& \omega_p \cdot \mathbf{v}_i(t) + \underbrace{\omega_c \cdot \mathbf{R}_1 * (\mathbf{y}_i(t) - \mathbf{p}_i(t))}_{\text{cognitive component}} + \\ & \underbrace{\omega_s \cdot \mathbf{R}_2 * (\hat{\mathbf{p}}(t) - \mathbf{p}_i(t))}_{\text{social component}}, \end{aligned} \tag{17.1}$$

where ω_p, ω_c, and ω_s are parameters called the inertia weight, cognitive acceleration constant and social acceleration constant, respectively. $\mathbf{R}_1$ and $\mathbf{R}_2$ are vectors of random numbers in $[0,1]$, and $*$ denotes piecewise multiplication of vectors. It can be observed that the velocity is adjusted to stay near good positions that the particle has found so far (personal best position (*pbest*) $\mathbf{y}_i(t)$), and to move towards the best position that the swarm has found so far (the global best position (*gbest*) $\hat{\mathbf{p}}(t)$). The former influence is called the cognitive component of the velocity update and the latter is called the social component. There exist many variants of PSO algorithms, e.g., instead of using the global best solution during the velocity update, a neighborhood topology can be defined for the swarm and the neighborhood best position (*lbest*) is used for velocity update (for more details see, e.g., [1] or [2]).

This formulation requires the optimization problem to be continuous, i.e., the configuration space needs to be an affine space or a vector space. Nevertheless, PSO algorithms have been proposed that can be applied to a discrete search space. In the case of RNA folding problems, there are the SetPSO [12] and the HelixPSO, proposed in [7]. Extensions and variants of HelixPSO have been investigated in [6, 8].

17.3.1 SetPSO

SetPSO is the first PSO algorithm for the RNA folding problem and has been proposed by Neethling and Engelbrecht [12]. As other PSO algorithms for discrete optimization problems, SetPSO differs from the standard PSO scheme that has been described above. It searches on the set of helices of a given RNA sequence and represents secondary structures as feasible sets of helices. Hence, the current position $p(t) \in X$ of a particle is characterized by its set of helices. Direction is represented with target configurations or target solutions, usually a combination of the particle's *pbest* and *lbest* solution, i.e., $lbest \cup pbest$.

Movement of a particle is defined as a change in the particle's current set of helices. This is done by the addition and by the removal of helices from the set of helices that corresponds to the current position of the particle. A set $O \subset S$ of helices which is removed from the particle's position is computed from the empty set by adding a helix with probability $p_I > 0$ if it is neither in the *pbest* solution nor in the *lbest* solution. A candidate set P of helices that might be added to the particle's position is computed from the empty set by adding each helix of the target solution with probability $p_C > 0$ and each helix from the set of all helices with probability $p_R > 0$. To avoid base pair conflicts, all helices in the set O are removed before those of the set P are added (if feasible). The velocity update in SetPSO is the computation of the sets O and P. The position update is the actual computation of the new solution, i.e., the computation of the new subset of helices. More details of SetPSO can be found in [12].

In this work, we use a swarm topology where all particles are connected to each other. Thus, all particles are drawn to the globally best solution. It is not specified whether this should be the historical or the current best solution. Later, we will refer to the "global memory" variant of SetPSO when the historical best solution is used, and to the "no global memory" variant when only temporally local data is used.

17.3.2 HelixPSO

Different from the SetPSO algorithms, HelixPSO uses maximal helices as the basic elements to form a solution to the RNA folding problem. Thus, for HelixPSO, S is the set of maximal helices. The position of a particle is represented as a permutation of the elements of S. This permutation can be converted to a solution of the RNA folding problem by computing a maximal feasible subset of S as follows: starting with the first helix of the permutation and iteratively always selecting the next feasible helix. Note that different particle positions (permutations) might lead to the same RNA structure (maximal subset of maximal helices).

Each particle i has a set of *candidate target positions* T_i and for each $t \in T_i$ a weight $w(t) > 0$. The particle tries to move closer to the target positions by swapping elements of the particle's current permutation such that it becomes more similar to a target permutation. The relative weight of a position in T_i determines the probability that it is chosen as a target. T_i is initialized with a single random position of weight 1.0, i.e., a permutation that is generated randomly so that each permutation has the same probability. After each iteration of HelixPSO, the weight of each target position is decreased by multiplication with a parameter ρ, $0 < \rho < 1$. Then the personal best position and either the global best position or the cluster best position (details are given later) are added to T_i with weight $\omega_c \cdot R_1$ and $\omega_s \cdot R_2$, respectively, where R_1 and R_2 are random numbers chosen uniformly in $[0,1]$. The constants $\omega_c > 0$ and $\omega_s > 0$ refer to the cognitive and to the social component, respectively. Observe that this is very similar to the impact of *pbest* and *gbest* (or *lbest*) in the standard PSO scheme.

The concern of the position update is to make a particle more similar to some target permutation by swapping positions in its helices list. Since not every swap in the permutation might actually change the represented RNA structure, a series of α

swaps is performed. Thus, α is the first important parameter of the algorithm. With a probability of $\beta > 0$, a directed swap, i.e., a swap towards a target chosen from T_i, is done, otherwise a random swap of two elements in the permutation is performed. More details about directed swaps can be found in [8].

HelixPSO uses a swarm topology that consists of clusters of particles. Every cluster is associated with a cluster best particle, i.e., a historical best position is not considered, but the current best position within the cluster. For every particle except the cluster best particle, the cluster best is used as the *lbest* position. The cluster best particle uses the *gbest* position instead. The idea of this behavior is to have multiple clusters that explore the search space more or less independently. The particles within a cluster should stay close to each other by orienting themselves towards the cluster best. On the other hand, the clusters should collaborate and join each other at the exploration of known good positions in the search space. To achieve this, the cluster best moves in the direction of the currently known global best position.

HelixPSO also contains two optimization variants: i) usage of 1-elitism and ii) a simulated annealing based heuristic to control desirable follow-up particle positions. 1-elitism means that after each simulation step, the position of the worst particle is reset to the global best position. In doing so, the target vector of this particle is not affected, though. The original version of the algorithm selected follow-up positions with a greedy strategy: the position was only accepted if it possessed better fitness. The simulated annealing-like version relaxes this constraint by allowing poorer solutions depending on the temperature. At the beginning, almost every solution is accepted, leading to a large swarm diversity. During the cooling process, the probability to allow changes for the worse is decreased and the behavior becomes more greedy. For more details on Helix-PSO see [6, 8].

17.4 Fitness Landscape Induced by PSO Algorithms

A difficulty with discrete PSO algorithms is that it is not clear how to define a suitable neighborhood relation that can be used to define a fitness landscape. An idea is to use the possible transitions between the configurations that particles can make during the run of the PSO (following certain rules that are determined by the algorithm) to define a neighborhood relation on the set of configurations. By this definition a fitness landscape results that allows to analyze the algorithm's optimization behavior for the given problem. But there is also a problem with this idea since for typical discrete PSO algorithms, e.g., the PSOs that are considered in this chapter, the particle movement contains a random component. Then it can happen that, in principle, every configuration can be reached by a particle from every other configuration within one time step.

A SetPSO particle can in principle open all helices at once and close an arbitrary set of new helices. Even though such an incident has a very small probability, it is possible. Similarly, considering a particle of HelixPSO, it is possible that it completely changes the resulting RNA conformation in one time step by swapping just two elements in the permutation. In some cases, there exist very few such swaps, making the transition from one configuration to certain other ones very improbable. But it can happen.

Consequently, if every transition that is possible for a particle results in a neighborhood relation for the corresponding configurations, the neighborhood graph would be the complete graph. This is not desirable, because the fitness landscape would be topologically trivial and the neighborhood graph would show transition events that are very unlikely. In order to be able to properly analyze the optimization behavior of discrete PSO algorithms, we propose to omit all edges with low transition probabilities and to include only those edges that correspond to single elementary operations, which are used by the particles to change their current configuration.

For SetPSO, the elementary operations to define a transition from one configuration into another are the removal, addition, or exchange of an element of S, i.e., of one helix. Thus, it is required for the neighborhood relation that two adjacent configurations differ by only one element. Since such transitions are very likely to happen, the neighborhood graph shows the most important transitions without becoming too cluttered.

For HelixPSO, the elementary operation is a swap of two maximal helices in the permutation of all maximal helices. Thus, it is required for the neighborhood relation that only one swap is necessary to transform a permutation corresponding to one configuration into a permutation corresponding to an adjacent configuration. Observe that we do not use the permutations themselves as configurations, but the maximal sets of helices that are derived from them. In this manner, the size of the fitness landscape can be reduced drastically, in exchange to becoming much denser. Table 17.3 shows the sizes of the resulting fitness landscapes for both SetPSO and HelixPSO for three example RNA molecules.

Table 17.3 Sizes of the example fitness landscapes for SetPSO and HelixPSO. X: set of configurations, χ: neighborhood relation, i.e., $|\chi|$ is the number of edges in the neighborhood graph.

Name	SetPSO		HelixPSO	
	$\lvert X\rvert$	$\lvert\chi\rvert$	$\lvert X\rvert$	$\lvert\chi\rvert$
Bacillus anthracis str.	1162	15851	407	75581
Aeropyrum pernix	7876	146279	1786	1432126
Toxoplasma gondii	1828	28015	653	184888

17.5 Visual Analysis Method: dPSO-Vis

The software tool *discrete Particle Swarm Optimization-Visualization* (dPSO-Vis) implements a new visualization method for discrete PSO algorithms that has been proposed very recently by Volke et al. [15]. The method defines a fitness landscape that is associated with a discrete PSO and a corresponding optimization problem as described above to provide a visual analysis of the optimization behavior of the

PSO. The dPSO-Vis tool uses a terrain metaphor to visualize this fitness landscape. The images that can be created with dPSO-Vis can also show information about the locations of the particles and their movement within the fitness landscape.

In the following subsections, dPSO-Vis is described. At first, the topological methods for understanding the fitness landscapes are explained. Then, the computation and visualization of the topological landscape is shown. Finally, the visualization of the particles within the topological landscape is outlined.

17.5.1 Topology of Discrete Fitness Landscapes

Starting from the set of configurations X and the fitness function $f : X \to \mathbb{R}$, that are given by the optimization problem (see Section 17.2), and the neighborhood relation χ that reflects the search behavior of the swarm (see Section 17.4), we have a *discrete search space* (X,χ) and a *discrete fitness landscape* (X,χ,f). Because we consider discrete optimization problems, the search space can be represented as a simple, undirected graph $G = (V,E)$, where $V = X$ and E is determined by χ. A fitness landscape can then also be described by (G,f). For the case studies in this chapter, G is connected and called the *neighborhood graph*. Note that we consider minimization problems in this chapter and that the function f assigns a cost value (energy value) to every configuration. Hence, even though we call the landscapes fitness landscapes they are actually cost landscapes and therefore low points in the landscape correspond to good solutions, i.e., solutions with a high fitness.

To represent the topology of the landscape, the *barrier tree* data structure can be used. It has been proposed by Flamm et al. [4] as a result of their topological analysis of fitness landscapes (which they called *energy landscapes*). Using the results of this analysis and the barrier tree for our visualization method has the advantage that it can be applied to a wide range of optimization problems and their PSO algorithms, even when the resulting fitness landscapes are highly degenerate. The main results from Flamm's work are briefly described in the following paragraphs.

Among the vertices of the neighborhood graph, which are of special interest here, are the local minima. The set of local minima, $\mathscr{M}$, is denoted as

$$\mathscr{M} = \{v \in V : \forall w \in N(v) : f(v) \le f(w)\} ,$$

where $N(x) = \{y \in V : \{x,y\} \in E\}$ is the set of neighbors of x. It is useful to identify local minima that are directly connected in G. Thus, $\mathfrak{M}$ is defined as the set of all connected components of $G[\mathscr{M}]$, i.e., the subgraph of G with vertex set $\mathscr{M}$.

A central concept in Flamm's work is the notion of cycles. A cycle of $x \in V$ with $f(x) \ge \eta$ at height η is the connected component of the corresponding level set $\{y \in V | f(y) \le \eta\}$ that contains x. The component is denoted with $C_\eta(x)$. Additionally, the open cycle $C^\circ_\eta(x)$ is basically $C_\eta(x)$ without the points that have cost η, formally:

$$C^{\circ}_{\eta}(x) = \begin{cases} \bigcup_{\eta' < \eta} C_{\eta'}(x) & \text{if } \eta > f(x) \\ \{x\} & \text{if } \eta = f(x) \text{ and } x \in \mathscr{M} \\ \emptyset & \text{otherwise} \end{cases}$$

With a gradually increasing value of η, the corresponding cycles around the local minima become larger and eventually merge. The cost value η_m that leads to a merge event of two or more cycles is called the *barrier* between the corresponding local minima. The reason for choosing this name is that every path connecting two such minima contains at least one vertex with a cost value at least as high as the barrier. Beside the barriers, i.e., the cost values, the actual vertices, where two cycles contact, are of interest. These vertices are called saddle points. To formally define them, let $\mathbb{P}_{xy}$ be the set of all graph-theoretic walks between x and y in G. Then, a saddle point between two local minima $x, y \in \mathscr{M}$ is a point $s \in V$ such that

(1) $\forall \eta < f(s) : C_{\eta}(x) \cap C_{\eta}(y) = \emptyset$,
(2) $C_{f(s)}(x) = C_{f(s)}(y)$, and
(3) $\exists \mathbf{p} \in \mathbb{P}_{xy} : s \in \mathbf{p}$ and $\forall m \in \mathscr{M} : G[C^{\circ}_{f(s)}(m) \cap \mathbf{p}]$ is connected.

Similar to local minima, it is useful to define weak equivalence classes of saddle points. Flamm et al. considered two saddle points p and q to be weakly equivalent, if $f(p) = f(q)$ and $C_{f(p)}(p) = C_{f(p)}(q)$. The set of weak equivalence classes of saddle points is denoted with $\mathfrak{S}$.

Now, the *barrier tree* $\mathfrak{B}(G, f) = (\mathfrak{U}, \mathfrak{E})$ of a fitness landscape can be defined as a directed tree. The vertices are the equivalence classes of local minima and saddle points, i.e., $\mathfrak{U} = \mathfrak{M} \cup \mathfrak{S}$. The barrier tree reflects the inclusion relation between the corresponding level sets. The arc set of the barrier tree can be defined as follows:

$$(x,y) \in \mathfrak{E} \subset \mathfrak{U} \times \mathfrak{U} \iff \begin{array}{l} C_{f(x)}(x) \supset C_{f(y)}(y) \text{ and} \\ \nexists z \in \mathfrak{U} : C_{f(x)}(x) \supset C_{f(z)}(z) \supset C_{f(y)}(y) \end{array} \tag{17.2}$$

Flamm et al. [4] gave a different definition based on paths in the neighborhood graph, which is equivalent to the definition given here. Additionally, their work contains a proof that the barrier tree is a rooted tree, given that G is connected.

The barrier tree allows to easily find the minima within the search space and also shows the barriers between them. This makes it especially useful for the field of computational biology, where it has been applied several times, e.g., [3, 9].

In order to have the barrier tree reflecting the whole search space, Volke et al. [15] proposed to add the global maximum to the set of vertices and to insert an arc from the global maximum to the root of the barrier tree. Then, every arc represents a certain interval of the cost function (i.e., the interval between the costs of the vertices that are connected by the arc) and altogether the whole cost range between the maximum cost and minimum cost is covered by the arcs. Even though the barrier tree already contains important structural information, it is useful to associate the actual configurations of the optimization problem with the corresponding parts of the tree to support further analysis. Volke et al. [15] defined a partition of the configuration space into so called *basins* and associated every barrier tree arc with one basin. Given an arc $a = (v_1, v_2)$ of G, the basin $B(a)$ is basically the connected component of $f^{-1}([f(v_2), f(v_1)])$ that contains v_2. Here $f^{-1}([f(v_2), f(v_1)])$ is the set of all configurations with a cost value between between $f(v_2)$ and $f(v_1)$. More formally,

$$B((v_1, v_2)) = \left(C^{\circ}_{f(v_1)}(v_2) \setminus C_{f(v_2)}(v_2)\right) \cup v_2. \tag{17.3}$$

The barrier tree itself can be computed with the flooding algorithm [3, 4]. This algorithm can be easily extended to also compute the partitioning of the configuration space.

17.5.2 Visualizing the Topology Using a Landscape Metaphor

An obvious approach to visually understand the topology of a fitness landscape would be a layout of the barrier tree. However, since we are interested in more than the structure of the search space and thus need to visualize additional information, a simple tree layout is not appropriate. One problem that has to be solved is that a goal of a good graph layout is optimal usage of the screen space. But this goal collides with the idea to use uncovered screen space for showing additional data.

Another approach is to generate a terrain from the barrier tree that is topologically equivalent. This has first been proposed by Weber et al. [16] for visualizing scalar field topology in scientific visualization with 3D landscapes. Oesterling et al. [13] applied the problem to nD-point cloud data and improved visual clarity by using a 2D landscape profile.

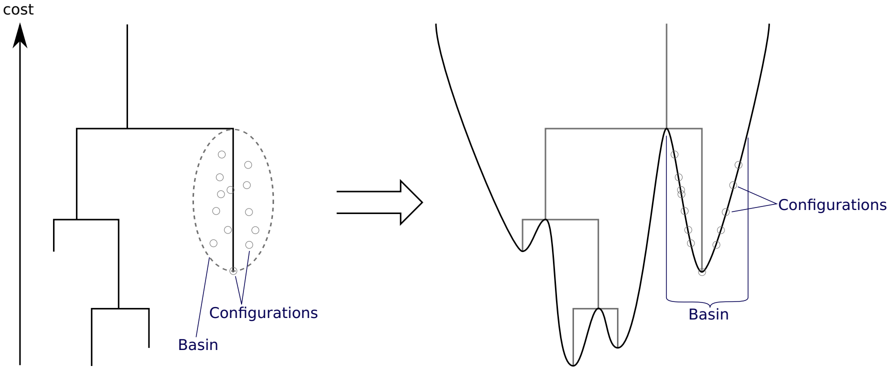


Fig. 17.1 Starting from a barrier tree (left part) a topologically equivalent barrier landscape (right part) is computed. The image shows, how structural properties such as minima or saddles within the barrier landscape correspond to vertices in the barrier tree. Every valley in the barrier landscape has a size proportional to the size of the associated basin. Parts of the landscape profile can be associated with respective configurations.

A similar approach is used in dPSO-Vis. The goal is to compute a 2D landscape profile, in the form of a height graph over a 1D line, that is topologically equivalent to the fitness landscape of the optimization problem and therefore has the same barrier tree. We call such a profile the *barrier landscape*. It contains a valley for every leaf node of the barrier tree and nested valleys for every subtree of it. If two subtrees are joined by an inner node, then there is a corresponding mountain pass within the landscape profile that joins the corresponding valleys. An overview of this correspondence is shown in Figure 17.1.

To generate the barrier landscape, a simple recursive algorithm can be used. Every part of the landscape profile corresponds to an edge in the barrier tree, so that the landscape can be constructed by traversing the tree in depth-first or breadth-first order. Considering an edge e of the barrier tree, we generate the height profile of the edge itself and the subtree below it on a given horizontal interval I. We have a maximal and a minimal cost value, i.e., the cost values of the start- respectively the end-node of e, and a volume in terms of the size of the basin associated with e. The part of I, which is proportional to the volume associated with e, is used to generate the left and right *slopes* of e, as shown in Figure 17.2. The remaining part of I is split between the child edges of e and used when recursively generating the corresponding landscape parts. Between the child valleys, a saddle point has to be inserted to preserve the topological structure.

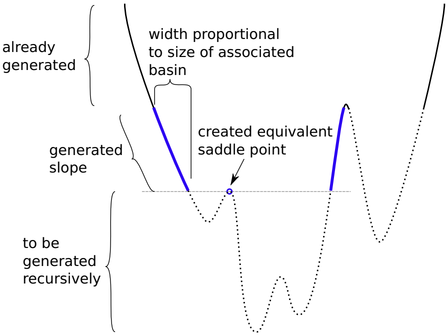

already generated
width proportional to size of associated basin
generated slope
created equivalent saddle point
to be generated recursively

Fig. 17.2 Recursive generation of the barrier landscape. The blue slopes are just being generated, as well as the saddle point between the areas for the child valleys.

The slopes are created by dPSO-Vis such that the "shape" of the search space, i.e., the distribution of the configurations over the cost values, is visible. The creation of the slopes is done by starting at a height that corresponds to the maximum cost value and that vertically drops down in general. Plateaus, i.e., short horizontal shelves, are inserted at certain height levels that have a size according to the number of configurations at the corresponding cost level. We use a logarithmic scaling here to stress lower valleys that often are of low volume. The result of this process is visible in the landscape profile shown in Figure 17.3.

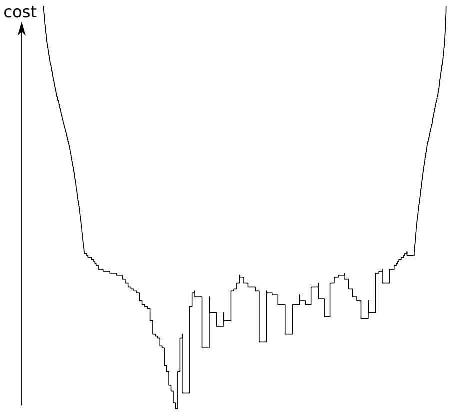
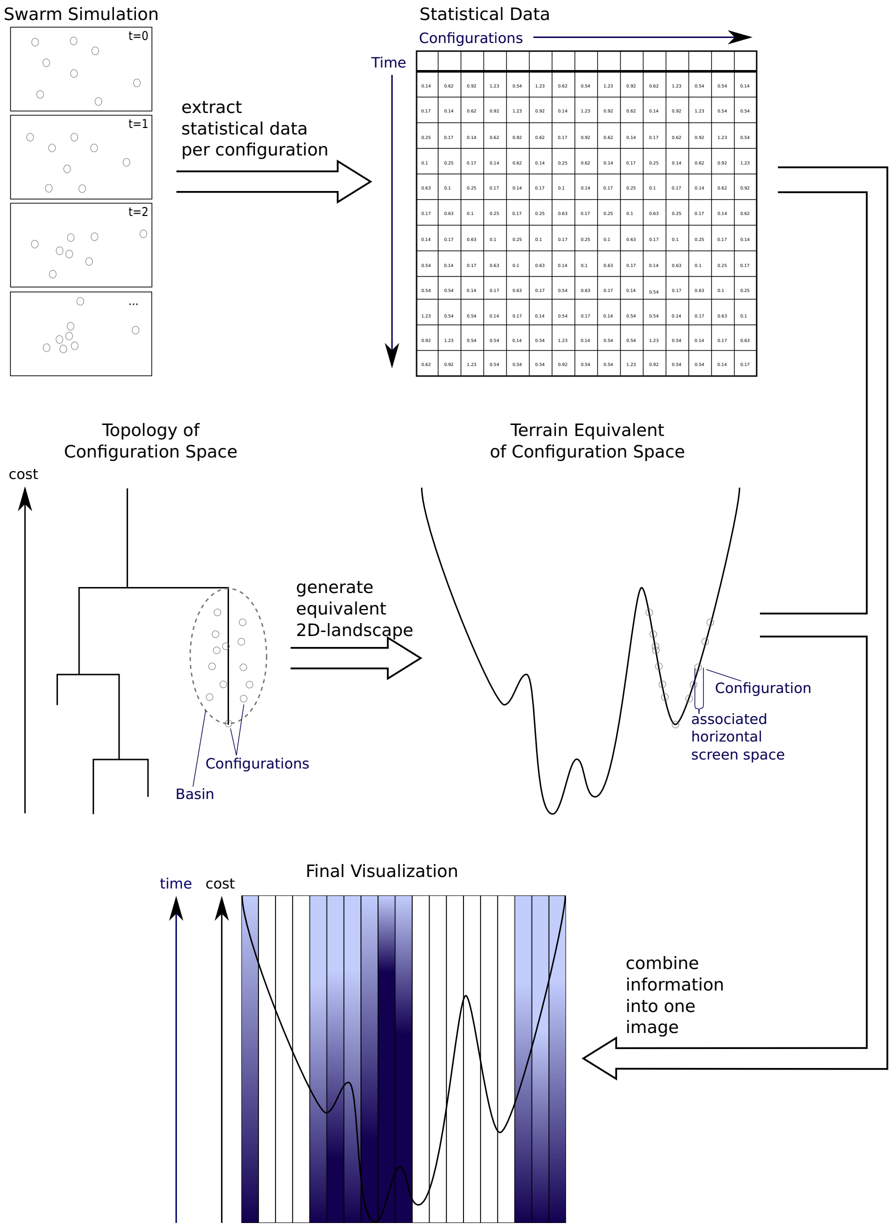


Fig. 17.3 A plot of an example barrier landscape with logarithmic scaling. The width of the plateaus corresponds to the number of configurations at that fitness level.

17.5.3 Visualizing the Swarm Movement within the Landscape

The use of visualization for analyzing the behavior of swarm algorithms for combinatorial optimization is not very common so far. One reason is that most existing visualizations for swarm algorithms are for continuous spaces to plot particle trajectories. But these methods usually require that the configuration space is a vector space which is not the case for discrete problems. Moreover, trajectories can not represent the random nature of the particle positions [14] and thus might not be a very appropriate means to understand the behavior of swarm algorithms. By far the most papers on swarm algorithms for combinatorial optimization use statistical measures on simulation runs to analyze the optimization behavior. Examples are the quality of the best solution found or measures of swarm diversity. Such measures can be used to compare different algorithms but they do not give many hints about the causes of the observed differences.

The visualization method of dPSO-Vis proposed in [15] shows time-dependent and configuration-local statistical data about the particles at all configurations during swarm simulation runs based on the barrier landscape. That landscape represents the topological structure, so that direct adjacency within the barrier landscape does not necessarily imply direct adjacency of configurations in the search space. However, it allows linking data with interesting regions of the search space. For example, it gives information about the basins or local minima that have been found by an algorithm. It also shows how often the basins have been found and how long the particles stay in different basins.



0.14	0.62	0.92	1.23	0.54	1.23	0.62	0.54	1.23	0.92	0.62	1.23	0.54	0.54	0.14
0.17	0.14	0.62	0.92	1.23	0.92	0.14	1.23	0.92	0.62	0.14	0.92	1.23	0.54	0.54
0.25	0.17	0.14	0.62	0.92	0.62	0.17	0.92	0.62	0.14	0.17	0.62	0.92	1.23	0.54
0.1	0.25	0.17	0.14	0.62	0.14	0.25	0.62	0.14	0.17	0.25	0.14	0.62	0.92	1.23
0.63	0.1	0.25	0.17	0.14	0.17	0.1	0.14	0.17	0.25	0.1	0.17	0.14	0.62	0.92
0.17	0.63	0.1	0.25	0.17	0.25	0.63	0.17	0.25	0.1	0.63	0.25	0.17	0.14	0.62
0.14	0.17	0.63	0.1	0.25	0.1	0.17	0.25	0.1	0.63	0.17	0.1	0.25	0.17	0.14
0.54	0.14	0.17	0.63	0.1	0.63	0.14	0.1	0.63	0.17	0.14	0.63	0.1	0.25	0.17
0.54	0.54	0.14	0.17	0.63	0.17	0.54	0.63	0.17	0.14	0.54	0.17	0.63	0.1	0.25
1.23	0.54	0.54	0.14	0.17	0.14	0.54	0.17	0.14	0.54	0.54	0.14	0.17	0.63	0.1
0.92	1.23	0.54	0.54	0.14	0.54	1.23	0.14	0.54	0.54	1.23	0.54	0.14	0.17	0.63
0.62	0.92	1.23	0.54	0.54	0.54	0.92	0.54	0.54	1.23	0.92	0.54	0.54	0.14	0.17

Fig. 17.4 dPSO-Vis pipeline: From the particle swarm simulation run, statistical data is computed, resulting in a time-series of different statistical measures for every configuration of the search space. From the barrier tree the barrier landscape is computed. Thus, every configuration is associated with a certain horizontal interval on the screen. Reusing the vertical axis as time-axis, the time-series for every configuration is plotted whereby the measured values are color-coded.

The dPSO-Vis tool shows four different measures for every configuration (mostly time-dependent):

(1) the *covering measure* shows if the configuration has been visited by the swarm during the simulation,
(2) the *accumulated visitor count* shows how often a configuration has been visited until the current time-step,
(3) the *current visitor count* shows how many particles are located at a configuration at the current time-step, and
(4) the *last visitor age* shows how many time steps ago the configuration has lastly been visited by a particle.

When computing the barrier landscape, every configuration is assigned a certain horizontal interval on the screen. As shown in Figure 17.4, this interval and the height of the image define a rectangle that can be used to plot the statistical data for the respective configuration. The vertical axis is reused as time-axis, so that the position on the screen characterizes the shown data in terms of configuration and time-step. To use the screen space efficiently, the measured value is represented by its color.

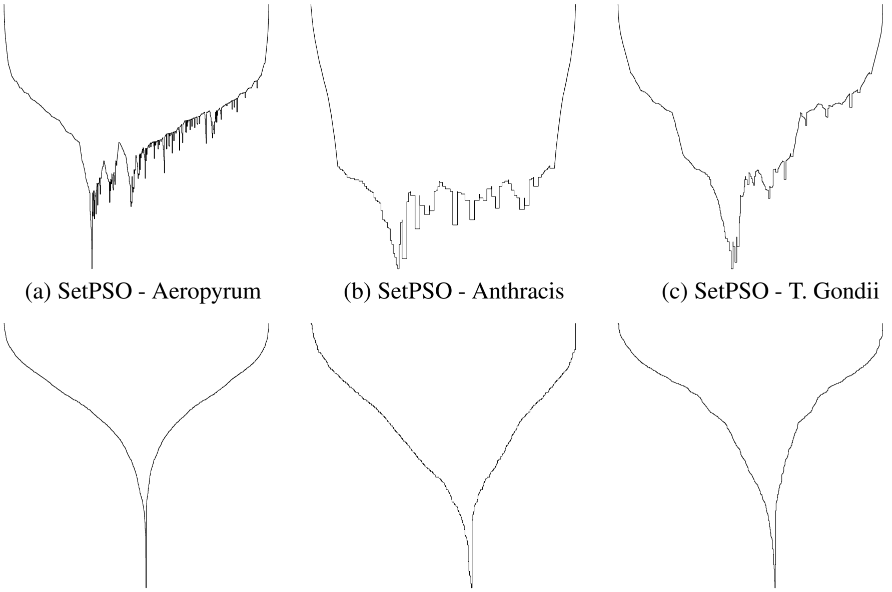
Two different color models are used to encode the values that vary the hue and the luminance, respectively. The latter is used where the actual value is of interest, e.g., when showing visitor counts, because the human visual system is well-trained to distinguish and compare different luminance. Variation of the color tone is used, where the focus is on qualitative differences, e.g., when contrasting "old" basins versus "new" ones. Here, the human ability to distinguish different colors without assigning absolute, numeric values with it, is leveraged.

Using dPSO-Vis, it is possible to comprehend both global properties of the swarm and local behavior in the context of the search space. Thus, the algorithm designer is able to analyze the benefits and penalties of design decisions in greater detail and more in depth than by plotting global statistical measures. In the following section, we show how dPSO-Vis can be used to analyze and compare the optimization behavior of PSO algorithms.

17.6 Application to SetPSO and HelixPSO

In this section, dPSO-Vis is used to visually analyze SetPSO and HelixPSO. This is only a brief overview to show the applicability of the method, but does not suffice as an in-depth analysis. All test runs have been done with a population of size 10 over 100 iterations.

A comparison of two algorithms can be done by comparing their fitness landscapes. Figure 17.5 shows plots of the barrier landscapes for SetPSO and HelixPSO for the three example RNA molecules from Tables 17.1 to 17.3. There are interesting differences visible. The landscapes of SetPSO are rather jagged and contain many subvalleys. The most interesting valley, that contains the global minimum, is small compared to the sizes of other deep, but non-optimal valleys (compare [17] where it was shown that relative basin size is an important factor for the difficulty of a



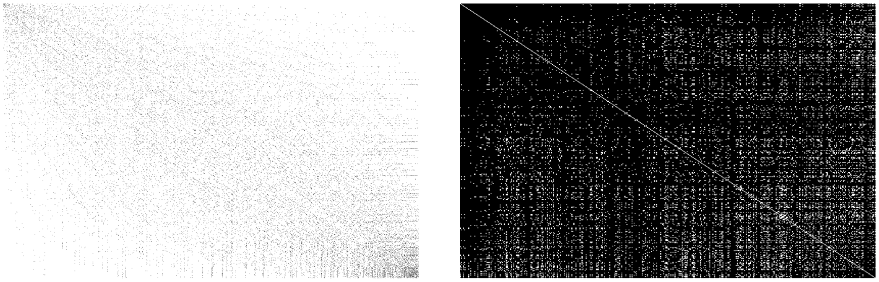
(a) SetPSO - Aeropyrum (b) SetPSO - Anthracis (c) SetPSO - T. Gondii

(d) HelixPSO - Aeropyrum (e) HelixPSO - Anthracis (f) HelixPSO - T. Gondii

Fig. 17.5 Comparison of Barrier Landscapes of SetPSO and HelixPSO for three example RNAs. Valley sizes are scaled logarithmically, so that smaller valleys are emphasized.

problem for PSO). This implies that SetPSO particles are prone to become locked in undesirable parts of the search space. SetPSO parameters should be adjusted, such that the particles can easily leave local minima and re-explore less optimal parts of the search space. The barrier landscapes of HelixPSO contain only one valley in all cases. There appears to be a very good overall connectivity within the HelixPSO search space.

To further investigate this, Figure 17.6 shows the connectedness matrix between the possible configurations for the Anthracis RNA for SetPSO and HelixPSO. These pictures have to be interpreted carefully, because they show only the connectedness with respect to a single basic operation as has been explained in Section 17.4. Since SetPSO and HelixPSO use different basic operations, the figures cannot be compared directly. Moreover recall, that a particle can perform several basic operations within a time step. Nevertheless, it is quite obvious that the search space of HelixPSO has a much higher connectedness. A sparse neighborhood graph can be a problem for a PSO algorithm because there might be only few paths leaving a local minimum and it can be difficult to achieve enough diversity within the swarm. But a too high connectedness might also be a problem because it is difficult for the algorithm to get good hints about beneficial search areas. More transition possibilities imply higher probability to re-search certain areas multiple times and thus larger search times. According to the figures, the first issue could potentially become a problem of SetPSO, whereas the latter could appear as a problem of HelixPSO.



(a) SetPSO (b) HelixPSO

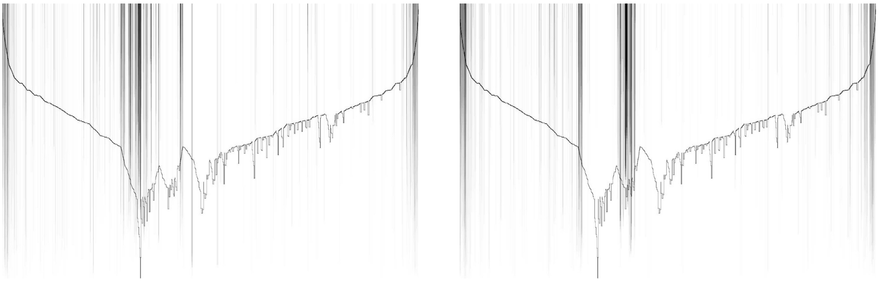
Fig. 17.6 Adjacency matrices of SetPSO/HelixPSO landscapes for Anthracis RNA. A black dot means, there is a connection between the configurations corresponding to the row and column, respectively.

A reason for this difference is the representation of particle state within HelixPSO. The number of RNA secondary structures is bounded by 2^n for n Helices and might be much smaller, because of conflicts between the Helices. SetPSO directly uses sets of Helices to encode particle positions. Thus, the number of SetPSO configurations is also bounded by 2^n. On the other hand, HelixPSO uses permutations of the helices as internal configurations, leading to a number of $n!$ different particle states, that are mapped to 2^n RNA configurations. This introduces a high degree of artificial symmetry, that increases factorial with n. Therefore, it is probable that HelixPSO landscapes always consist of a single valley, especially for large RNA sequences.

Next, the influence of global memory on SetPSO will be considered. Figure 17.7 shows the accumulated visitor count of two example runs for the Aeropyrum RNA. Since PSOs are inherently random, rerunning the algorithm could show different results, but the pictures have been chosen to represent our overall findings of several runs. It can be seen, that the two variants searched different areas of the configuration space during the first iterations. In both cases, one valley is searched intensively. It is generally a good sign, that the algorithm explores the space around a good solution in great detail. However, without the global memory, SetPSO often just forgets about the optimal valley, even if it finds it once or twice.

The visualization tool dPSO-Vis is very useful when investigating the influence of different parameters on the optimization behavior of an algorithm. Finding the best parameter values is a principle problem of metaheuristic algorithms and typically requires much testing effort. Visualization can help to largely reduce the necessary test effort because it helps to understand why certain parameter value combinations are good whereas others are not.

In Figure 17.8, the influence of the probabilities P_I and P_C on SetPSO, i.e., the probability to add a helix to the open set and the probability to add a helix to the closing set, respectively, are visualized for the Aeropyrum RNA.



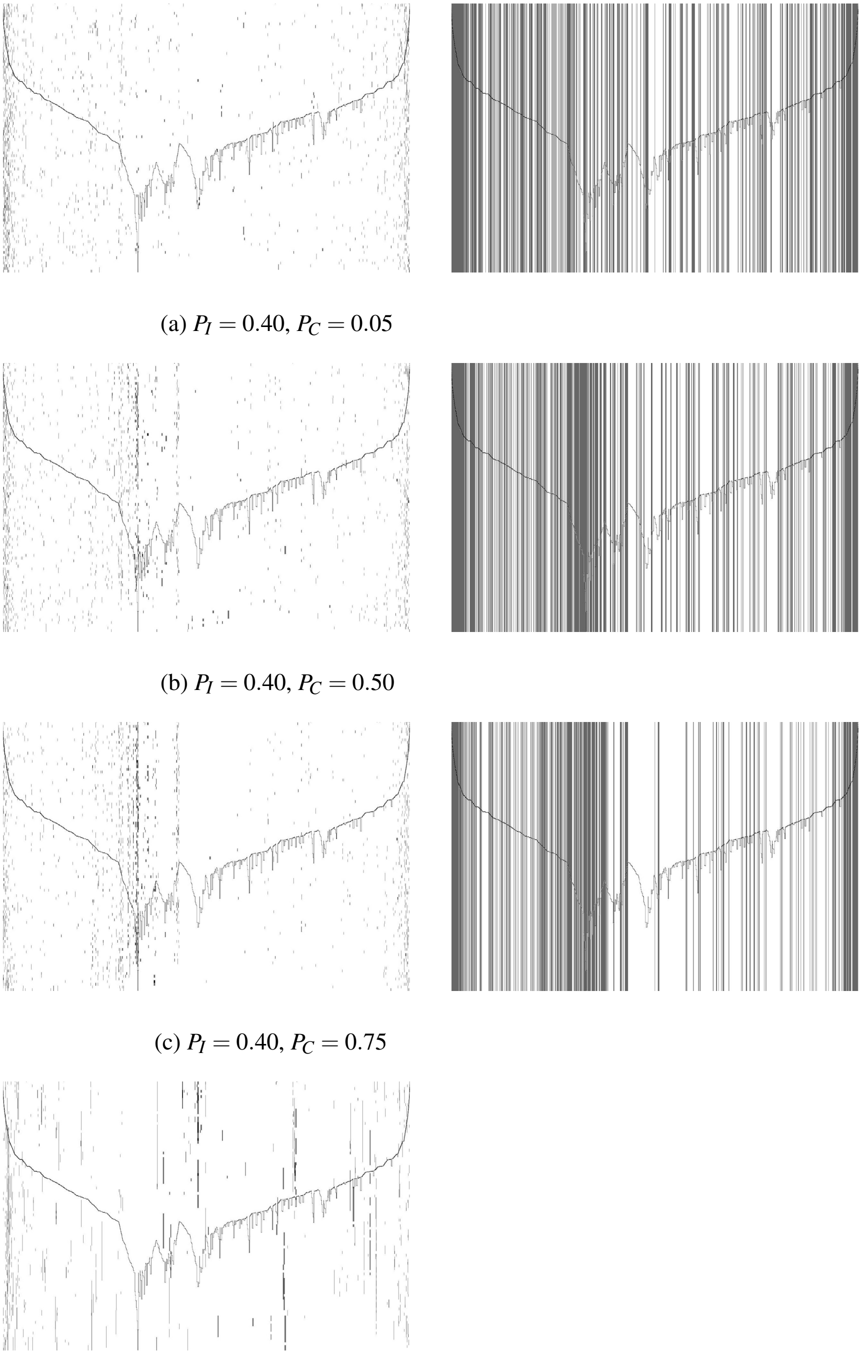
(a) With global memory (b) Without global memory

Fig. 17.7 Accumulated visitor count for SetPSO ($P_I = 0.4$, $P_C = 0.75$, $P_R = 0.15$), Aeropyrum-RNA

In Figures 17.8(e) and 17.8(g), the influence of parameter P_I can be seen. Both subfigures show the current visitor count. For a very small value of $P_I = 0.05$, the swarm cannot find the configurations with a low energy and does not converge in the first 100 iterations. For values higher than $P_I = 0.4$, the swarm concentrates the search around the configuration with minimum energy. This confirms the findings of Neethling and Engelbrecht [11] that the algorithm performs poorly with small P_I. A certain entropy is necessary for good results.

The influence of parameter P_C on SetPSO is shown in Figures 17.8(a) to 17.8(f). The left column shows the current visitor count and the right column shows the covering. In accordance with Neethling's results, the covering depictions show that the value of P_C does not influence much, whether the global optimum is found or not. However, that does not mean that P_C has no influence at all, as can be seen in the visitor count images. For a very small value of $P_C = 0.05$, the swarm does not concentrate around the minimum. The covering measure in Subfigure 17.8(b) shows that the areas around local minima are often slightly better covered than other areas. But around the global minimum, there is not a particular strong concentration of visits. This is different for higher values of $P_C = 0.4$, where Subfigures 17.8(c) and 17.8(d) clearly show a concentration of the search around the global minimum and a convergence of the swarm.

For HelixPSO three variants – greedy strategy with 1-elitism, simulated annealing strategy with 1-elitism, and simulated annealing strategy without 1-elitism – are compared in Figure 17.9 for the Aeropyrum RNA. The current visitor count in the first row of Figure 17.9 shows that both variants of HelixPSO with 1-elitism converge very fast to the area around the global minimum. Also simulated annealing HelixPSO without 1-elitism finds the area of the global minimum fast. It concentrates its search around the global minimum after about 70 iterations. The covering in the second row of Figure 17.9 shows that HelixPSO with simulated annealing but no 1-elitism has covered a much larger area of the configuration space than the other two variants. The other two variants have not searched large parts of the



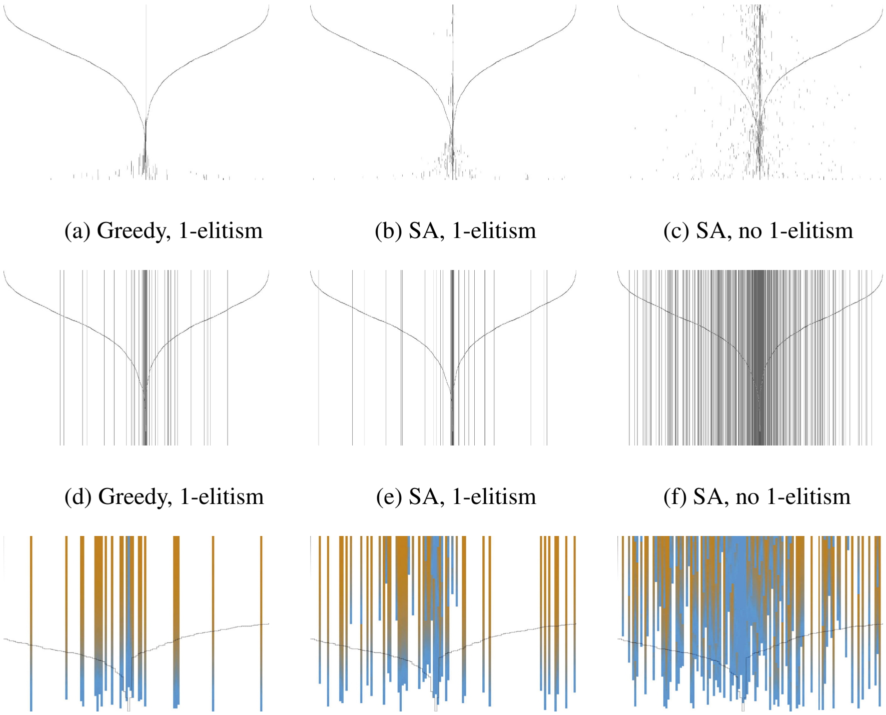
(a) $P_I = 0.40, P_C = 0.05$

(b) $P_I = 0.40, P_C = 0.50$

(c) $P_I = 0.40, P_C = 0.75$

(d) $P_I = 0.05, P_C = 0.75$

Fig. 17.8 SetPSO with varying entropy weight and closing probability (Aeropyrum RNA). In the left column the current visitor count is shown and in the right column the covering.



(a) Greedy, 1-elitism (b) SA, 1-elitism (c) SA, no 1-elitism

(d) Greedy, 1-elitism (e) SA, 1-elitism (f) SA, no 1-elitism

(g) Greedy, 1-elitism (h) SA, 1-elitism (i) SA, no 1-elitism

Fig. 17.9 HelixPSO on Aeropyrum RNA. In the first row the current visitor count is shown, in the second row the covering and in the last row the last visitor age is shown.

search space but have concentrated their search very early to the areas around the very small energy configurations.

The last visitor age measure that is shown in the bottom row of Figure 17.9 shows a difference between the Greedy version and the simulated annealing version of HelixPSO with 1-elitism. The Greedy version does never search outside the area around the global minimum in the second half of the run (i.e., after 50 iterations). This is different for the simulated annealing version, where the swarm searches in larger areas of the configuration space also in the second half of the search space. Clearly, the annealing variant without 1-elitism does not restrict itself to the area around the global minimum but searches all parts of the configuration space during the whole run. It can be concluded that the greedy variant might get problems with more complicated search spaces because the swarm concentrates too fast. A 1-elitism strategy seems useful when it is necessary to find good solutions fast.

17.7 Conclusions

We have discussed the problem to visualize the optimization behavior of particle swarm optimization (PSO) algorithms for discrete optimization problems. Typically, it is difficult to analyze the optimization behavior of such algorithms and visualization might help. However, for discrete PSO algorithms it is often hard to even characterize the locations of the swarm particles within the configuration space. Therefore, some simple visualization approaches that directly map the locations of the particles into a 2D or 3D space are not applicable.

An interesting visual approach that combines a terrain representation of the fitness landscape topology with configuration-local, time-dependent statistical measures of PSO runs is implemented in the visualization tool dPSO-Vis. The principle of dPSO-Vis is to combine the configuration set and the fitness function of the optimization problem with the transition behavior of the particles of a PSO algorithm to define a fitness landscape, which represents the search behavior of the algorithm. This fitness landscape is called the barrier landscape. It was demonstrated in this chapter how dPSO-Vis can be used to analyze and compare the optimization behavior of PSO algorithms. The two example PSO algorithms that have been used in this chapter were SetPSO and HelixPSO. Both algorithms have been designed for solving the RNA folding problem.

Acknowledgements. This work was supported by European Social Fund (ESF App.No. 100098248 and App.No.100098251) and by the DFG IRTG 1131.

References

[1] Eberhart, R.C., Kennedy, J.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)

[2] Engelbrecht, A.P.: Computational Intelligence: An Introduction. John Wiley and Sons, Chichester (2002)

[3] Flamm, C., Fontana, W., Hofacker, I.L., Schuster, P.: RNA folding at elementary step resolution. RNA 6, 325–338 (2000)

[4] Flamm, C., Hofacker, I.L., Stadler, P.F., Wolfinger, M.T.: Barrier trees of degenerate landscapes. Z. Phys. Chem. 216, 1–19 (2002)

[5] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)

[6] Geis, M.: Secondary Structure Prediction of Large RNAs. Ph.D. thesis, Universität Leipzig (2008)

[7] Geis, M., Middendorf, M.: A particle swarm optimizer for finding minimum free energy RNA secondary structures. In: Proc. IEEE Swarm Intelligence Symposium, pp. 1–8 (2007)

[8] Geis, M., Middendorf, M.: Particle swarm optimization for finding RNA secondary structures. International Journal of Intelligent Computing and Cybernetics 4, 160–186 (2011)

[9] Heine, C., Scheuermann, G., Flamm, C., Hofacker, I.L., Stadler, P.F.: Visualization of barrier tree sequences. IEEE Transactions on Visualization and Computer Graphics 12, 781–788 (2006)

[10] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the International Conference on Neural Networks, pp. 1942–1948 (1995)

[11] Neethling, C.M.: Using SetPSO to determine RNA secondary structure. Ph.D. thesis, University of Pretoria (2008)

[12] Neethling, M., Engelbrecht, A.: Determining RNA secondary structure using set-based particle swarm optimization. In: Proc. IEEE Congress on Evolutionary Computation, CEC 2006, pp. 1670 –1677 (2006)

[13] Oesterling, P., Heine, C., Jänicke, H., Scheuermann, G., Heyer, G.: Visualization of high-dimensional point clouds using their density distribution's topology. IEEE Transactions on Visualization and Computer Graphics 17, 1547–1559 (2011)

[14] Secrest, B.R., Lamont, G.B.: Visualizing particle swarm optimization - Gaussian particle swarm optimization. In: Proc. IEEE Swarm Intelligence Symposium, pp. 198–204 (2003)

[15] Volke, S., Middendorf, M., Hlawitschka, M., Kasten, J., Zeckzer, D., Scheuermann, G.: dPSO-Vis: Topology-based visualization of discrete particle swarm optimization. Computer Graphics Forum 32, 351–360 (2013)

[16] Weber, G., Bremer, P.T., Pascucci, V.: Topological landscapes: A terrain metaphor for scientific data. IEEE Transactions on Visualization and Computer Graphics 13, 1416–1423 (2007)

[17] Xin, B., Chen, J., Pan, F.: Problem difficulty analysis for particle swarm optimization: deception and modality. In: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 623–630 (2009)

Chapter 18
Predicting Evolution and Visualizing High–Dimensional Fitness Landscapes

Bjørn Østman and Christoph Adami

Abstract. The tempo and mode of an adaptive process is strongly determined by the structure of the fitness landscape that underlies it. In order to be able to predict evolutionary outcomes (even on the short term), we must know more about the nature of realistic fitness landscapes than we do today. For example, in order to know whether evolution is predominantly taking paths that move upwards in fitness and along neutral ridges, or else entails a significant number of valley crossings, we need to be able to *visualize* these landscapes: we must determine whether there are peaks in the landscape, where these peaks are located with respect to one another, and whether evolutionary paths can connect them. This is a difficult task because genetic fitness landscapes (as opposed to those based on traits) are high-dimensional, and tools for visualizing such landscapes are lacking. In this contribution, we focus on the predictability of evolution on rugged genetic fitness landscapes, and determine that peaks in such landscapes are highly clustered: high peaks are predominantly close to other high peaks. As a consequence, the valleys separating such peaks are shallow and narrow, such that evolutionary trajectories towards the highest peak in the landscape can be achieved via a series of valley crossings.

18.1 Usefulness of the Metaphor

The structure of realistic fitness landscapes continues to be an active area of research. When discussing the ‘metaphor’ of the fitness landscape, researchers typically mean the one- or two-dimensional representation of fitness as a function of either genotype or phenotype. This is a powerful image that immediately evokes how an evolving population may change, given population size and mutation rate: the familiar image of a landscape with hills and valleys initially described by

Bjørn Østman · Christoph Adami
Department Microbiology and Molecular Genetics & BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI 48824, U.S.A.
e-mail: {ostman,adami}@msu.edu

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_18, © Springer-Verlag Berlin Heidelberg 2014

Wright [16, 22, 25, 47, 52, 53] affects our intuition about evolutionary dynamics. However, as genes or traits do not affect fitness in isolation, considering the fitness contribution of only one or two genes may focus on too small a part of the fitness landscape to make relevant predictions about how populations actually evolve. Instead, it is possible that many more dimensions are required to understand adaptation, which makes a straightforward visualization of the landscape impossible. Is the fitness landscape metaphor still instructive if many genes contribute to fitness, that is, if the fitness landscape is high-dimensional? This concern has been raised multiple times [24, 40, 46]. In particular, it has been questioned whether the peaks separated by valleys commonly seen in one- and two-dimensional fitness landscapes exist at all in higher dimensions. An argument by Gavrilets [17, 18] states that when the number of dimensions is very high, peaks disappear because they become connected by neutral ridges. This mathematical argument was presented as a solution to the problem of valley-crossing [16, 22, 25, 47, 53]. Indeed, the discussion about the structure of fitness landscapes is intimately linked to the dynamics of adaptation. After all, Wright's shifting balance theory [51, 52]–which suggest that peak shifts occur via random drift during times when population sizes are small–is only relevant if valleys actually need to be crossed. Fisher, on the other hand, figured that landscapes change often enough that populations can simply wait on a local peak until beneficial mutations became possible [15], after which the new peak will be climbed via a series of slightly beneficial mutations. If, on the other hand, all high-fitness genotypes are actually connected by ridges of genotypes with equal fitness, populations could then readily evolve along these ridges, and thus the problem of having to cope with deleterious mutations disappears.

However, two important arguments exist against this solution. First, even if these ridges exist, it is not clear whether they are numerous enough that they are preferred by an evolving population. As these ridges appear only once the fitness landscape is high-dimensional, the number of possible paths for the population to take is exponentially large. Only a very small fraction of these may be neutral ridges [18], and when the mutation supply is sufficient, it is more likely that the population will traverse a short (i.e., single-step) as well as shallow valley, rather than taking a longer neutral path (see results below). Secondly, when all high-fitness peaks are connected by neutral ridges as in Gavrilets′s Holey Landscape model, the high-fitness genotypes must permeate all areas of genotype space, and as a consequence are not clustered in genotype space [18]. However, it is not at all clear whether we should expect such a homogenous distribution of high fitness peaks. Indeed, just as Kauffman originally hypothesized [26], we have recently shown that—at least in the NK model of epistatic fitness landscapes—peaks are not evenly distributed, but that they instead cluster so that high fitness peaks are closer to each other than they are to other low fitness peaks [38]. In that work we studied sequences (haplotypes) with $N = 20$ loci only, which does not meet Gavrilets's criterion for a high-dimensional space. However, it is possible to look for the same phenomenon in landscapes with a much higher dimensionality, such as the landscape created by digital evolution environments such as Avida [1, 2, 36].

As for empirical landscapes, Whitlock et al. [46] asserted that it may not be possible to conclude beyond a shadow of a doubt that complex biological fitness landscapes are epistatic and contain multiple peaks, but that the evidence is very strong that they do. Since that review, many more empirical landscapes have been reconstructed, and the evidence for multiple peaks, strong epistatic interactions between genes, and local optima continues to increase [4, 13, 21, 27, 28, 29, 31, 41]. If biological fitness landscapes contain multiple peaks, the problem of just how populations manage to cross fitness valleys is real. Besides the now standard solutions of Fisher and of Wright, we now know that a high mutation-supply rate (product of population size and mutation rate) enables populations to cross valleys via compensatory mutations, despite the transient reduction in fitness [39, 45]. The more mutations the population incurs, the higher the level of standing genetic variation, and the higher the chance that inferior genotypes will give rise to offspring that are lucky enough to move closer to an adjacent peak. Of course, if the mutation-supply rate is low, then deleterious mutations are rarely tolerated, and in the extreme case of mutations being so rare that each mutation goes to fixation before the next is available, valleys of even moderate fitness loss cannot be crossed. But many biological systems have mutation-supply rates high enough that there are always plenty of opportunities for organisms of lower fitness to have offspring with mutations. In fact, some populations behave like quasispecies [11, 12]: typical examples are viruses [6] and even bacteria [9]. Such populations are characterized by extended clouds of genotypes containing many genetic variants, and these can easily cross valleys as the fixation probability of a deleterious mutant connected to the quasispecies is nonvanishing [50]. No matter how valleys are crossed during adaptation, we can now be almost certain that fitness landscapes contain distinct peaks, even in the multidimensional space of biological organisms. Therefore, we contend that the fitness landscape metaphor, exemplified by the image of hills and valleys in two dimensions, is useful in guiding our intuitions about evolutionary dynamics. While a high-dimensional landscape may differ quantitatively from the familiar picture of rolling hills and valleys, we believe that the idea that there are peaks and valleys in a biological fitness landscape is unlikely to be refuted.

18.2 Visualizing Fitness Landscapes

Even though low-dimensional fitness landscapes can sometimes serve as an appropriate visual for landscapes of higher dimensions, the use of other methods for visualizing multidimensional fitness landscapes is currently a topic of much interest. Techniques for visualization include different methods for creating two-dimensional representations of high-dimensional space [33, 49]. These methods are applied when complete knowledge of the fitness of all genotypes exists. Due to the inherent multidimensionality of genotypes, obtaining full information for biological systems is not feasible. The fitness landscapes of limited areas of genotype space have recently been constructed for *E. coli* [8, 30], RNA [41], HIV-1 [32], and *Aspergillus niger* [16]. This interest in reconstructing and visualizing fitness landscapes stems from the promise that knowledge of this function (i.e., fitness) holds, namely that in combination with population size and mutation rate, evolutionary dynamics becomes predictable. But what exactly does this predictability achieve? Which evolutionary events can we predict? Given the stochasticity inherent in evolutionary dynamics, does knowledge of the three core evolutionary parameters—population size, mutation rate, and fitness landscape—actually lead to testable predictions? Do we currently know enough about fitness landscapes that they can be used to make predictions about the likely future paths?

The most obvious benefit from knowing the structure of a fitness landscape is a better understanding of adaptation. For example, just knowing the topology of the local fitness landscape is enough to tell whether the population will experience adaptation or not. If the population is already located on a fitness peak, then it will remain there provided that the local structure is such that mutation-supply rate is too low for adjacent peaks to be reached. In the event that there are adjacent peaks of higher fitness, crossing the valleys in between would result in an adaptive relocation. As mentioned above, valley-crossing is only a hindrance to adaptation when the combination of low mutation-supply rate and deep and/or long valleys between peaks prohibits the population from taking advantage of the potential of deleterious mutations.

As speciation is a process intimately tied to evolutionary dynamics, it is also impacted by the structure of the fitness landscape [18]. Genetic differences underlying multiple species necessarily result in those species occupying different areas of genotype space. If these areas are not separated by valleys, then intermediate genotypes (hybrids, in the case of sexually reproducing species) are of high fitness, and both initiation and maintenance of speciation will be less likely. A multi-peaked landscape therefore promotes speciation [10]. Given two adjacent fitness peaks, a population in the vicinity of both can adapt to either optima. If there are no physical or genetic barriers to competition within the population, the population will end up on one or the other (most likely the highest of the two, but not necessarily so). However, if such barriers to competition do exist, the population may permanently split into two, eventually leading to ecological speciation [35]. If the peaks correspond to genomic regions or traits that directly affect reproduction, speciation will be more likely to occur. Additionally, the more rugged the fitness landscape is with respect to traits that affect resource use and reproductive isolation, the higher the chance that evolutionary branching will lead to speciation. Consequently, the more dimensions relevant to fitness that are under scrutiny, the more precisely we can predict the evolutionary outcome with respect to speciation. Both adaptation and speciation are thus affected by how rugged the fitness landscape is. The more epistatic interactions between loci, the more rugged the landscape is, and the more peaks there are. Not only does the number of peaks matter, spatial correlations between peak genotypes could also affect adaptation and speciation. If peaks are not evenly distributed in genotype space, but are clustered in some fashion, close proximity of peaks can facilitate peak shifts and thereby increase the likelihood of adaptive shifts and ecological speciation. For sexually reproducing species, speciation may require that valleys in between peaks are deep and wide enough to cause hybrid sterility or breakdown [44]. As we shall see below, there are reasons to believe that peak clustering is a generic feature of multi-peaked, epistatic fitness landscapes.

18.3 Landscape Structure and Peaks

Studying the large-scale structure of biological fitness landscapes is difficult due to several factors. Measuring fitness is not trivial and experimentally manipulating organisms to explore large areas of genotype space is an arduous task. However, some numerical models have fitness landscapes where all genotypes can be enumerated. The NK model is such a landscape [25, 26], in which the amount of epistatic interactions can be adjusted and ruggedness thereby varied in a controlled manner. Briefly, N is the number of bi-allelic loci in the genome, and K is the number of neighboring loci that each locus interacts with. Fitness is calculated as the average fitness contribution of each locus, which in turn is taken from a table of uniform random numbers. In a K=0 landscape loci do not interact, and as a consequence the landscape contains only a single peak. Adaptation in such a landscape is predictable, as there is no other outcome than the population ending up on the peak (or around it, in case of a high mutation rate). In contrast, for $K = N - 1$, all loci interact with all other, and the fitness of neighboring genotypes are completely uncorrelated, creating a maximally rugged landscape. Both of these extremes have been mathematically investigated, but neither have much bearing on real fitness landscapes, which are rarely non-epistatic, and never completely lacking in fitness correlations [16, 25]. The intermediate range of K, which creates varying degrees of epistasis and ruggedness, is difficult to study analytically, but lends itself easily to computational methods [39]. Here we present several landscape measures [42] that characterize the large-scale structure of the NK landscape, and whose specifics in turn affect adaptation and evolutionary dynamics. In accordance with the massif central hypothesis [26], peaks are unevenly distributed in genotype space, and tend to form clusters. Several lines of evidence support this inference. First, defining peaks as genotypes whose one–mutant neighbors are all of lower fitness, we look at the dependence of the number of peaks on landscape dimensionality. As the dimensionality of the NK model increases with the number of loci, N, the number of peaks increases exponentially (Figure 18.1). There is thus no indication that neutral ridges appear as the dimensionality of the NK model in increased. Looking at the spatial distribution of peaks (genotypes for which all N mutational neighbors have a lower fitness), we found that peaks are located closer to each other than expected if they were distributed randomly [38]. The Hamming distance between peaks is lower than between a set of random genotypes (Figure 18.2A), where the random control displaces peaks by randomly assigning them new genotypes without replacement. This correlation disappears for high K, but reappears when focusing on at the highest peaks only (Figures 18.2B-D). Moreover, the higher the peaks are, the more likely they are only located near each other. This can be seen from the strong association between the fitness of a randomly selected peak and the peaks in its near vicinity. Looking at all peaks and comparing their fitness to the average fitness of peaks close

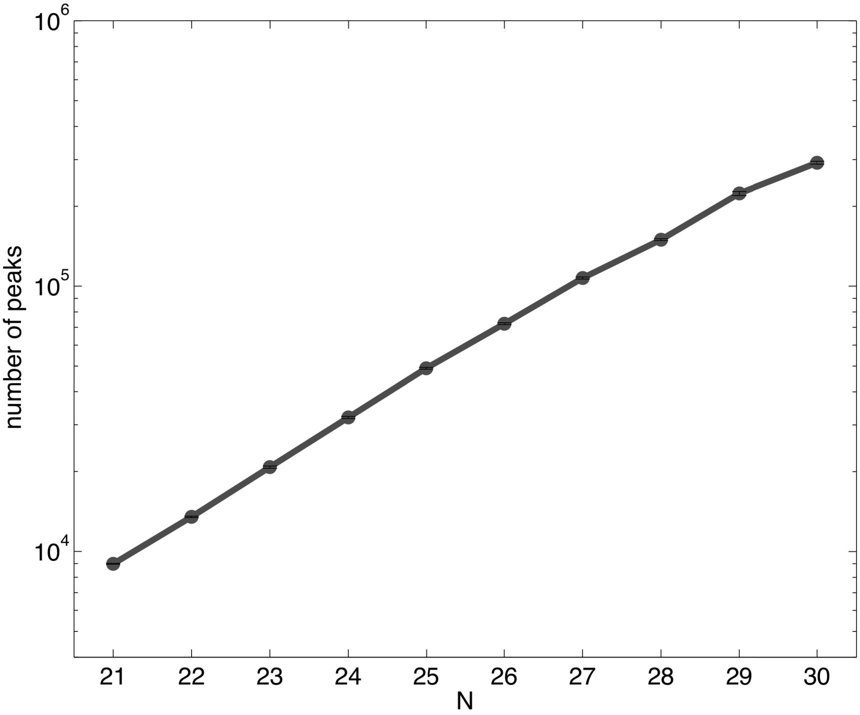


Fig. 18.1 In the NK landscape, the number of peaks increases exponentially as a function of the number of loci. Here shown for a highly rugged landscape, K=8, and N between 21 and 30. Peaks are here defined as genotypes whose N one-mutant neighbors all have lower fitness. Every datum is the average over five different landscapes, and the s.e.m. error bars in black are smaller than the markers. To the extent that it has been investigated, there is no indication that peaks cease to exist in multidimensional fitness landscapes, contra Gavrilets [17, 18].

by (Hamming distance of 2), a very strong correlation is evident (Figure 18.3). The strength of the correlation depends on the ruggedness of the landscape, but even though it is weaker for K=4 than for K=2, it is evident from this analysis that peaks are more likely to be found near peaks of a comparable rather than much different height.

The observed effect of a strong correlation between fitness of the highest peaks (and the watering down of that correlation when the peaks of lower fitness are included in the analysis) suggests an interesting feature of multidimensional rugged landscapes. Peaks are more likely to be located near other peaks of similar fitness, so not only is there a massif central, but there also exist clusters of peaks across the NK genotype space. If we look only at peaks of fitness above a threshold θ (Figure 18.2), the spatial correlation in genotype space is strong, but as we lower that threshold, more peaks appear and are spread out over more of genotype space. Additionally, as θ is lowered, more and more peaks become connected in a network where peaks are no further apart than a Hamming distance of two. We can observe a percolation phase transition for peak networks, where peak networks change from being disconnected to forming large networks that include nearly all peaks (Figure 18.4). (Because a network of peaks connected by a Hamming distance of exactly two cannot be connected to peaks separated by a Hamming distance of three, there are two large networks in Figure 18.4, which is why less than half of all peaks are found in the largest network.) If such rugged landscapes contain peaks in large areas of genotype space, we may then ask just how much of genotype space the clusters of peaks inhabit. Are peaks distributed evenly such that all points in genotype space are equally likely to be near a peak, or are there large areas of genotype space that are devoid of peaks? To test whether rugged landscapes contain peaks of low fitness distributed evenly in genotype space, we translated the bit-string NK genotypes into decimal numbers in the following way. Denote the binary string by the vector **s**. The corresponding decimal genotype is then

$$d(\mathbf{s}) = \sum_i s_i 2^{i-1}, \tag{18.1}$$

where the summation is over all indices of **s**. For example, the genotype 10010 is represented by the vector $\mathbf{s} = \{1,0,0,1,0\}$, and has the decimal genotype

$$d(\mathbf{s}) = 1\times 2^0 + 0\times 2^1 + 0\times 2^2 + 1\times 2^3 + 0\times 2^4 = 9. \tag{18.2}$$

This formalism can easily be extended to non-binary strings, e.g., genotypes consisting of more alleles. For DNA,

$$d(\mathbf{s}) = \sum_i s_i 4^{i-1}, \tag{18.3}$$

so for example (with A=0, T=1, C=2, G=3, reading right to left) TATA corresponds to

$$d(\mathbf{s}) = 0\times 4^0 + 1\times 4^1 + 0\times 4^2 + 1\times 4^4 = 68, \tag{18.4}$$

and the one-mutant GATA to

$$d(\mathbf{s}) = 0\times 4^0 + 1\times 4^1 + 0\times 4^2 + 3\times 4^4 = 196. \tag{18.5}$$

As we can see from this example, neighboring genotypes with Hamming distance one are not necessarily neighbors in this decimal genotype space. A genotype with two mutations can be closer than a single-mutant, as with TCTT, which has

$$d(\mathbf{s}) = 1\times 4^0 + 1\times 4^1 + 2\times 4^2 + 1\times 4^4 = 101. \tag{18.6}$$

If there were no structure to how peaks are distributed in genotype space, the distribution of peaks in decimal genotype space is expected to appear even. However, despite the disconnect between neighboring genotypes in decimal space, plotting peak fitness as a function of the decimal genotype reveals an intriguing order not observed among random genotypes (Figure 18.5). Peaks are irregularly distributed and large areas have no peaks. These basins of attraction [14] clearly show that peaks are not spread evenly in genotype space, but cluster together in agreement with the analysis above (Figures 18.2 and 18.3).

The question of whether populations would be able to utilize the neutral ridges in a Holey Landscape [17] can be investigated using a simple computational model. We begin with a homogenous population at the same genotype having a fitness of 1. The object is to find a genotype two mutations away either through a valley or a larger number of mutations away via a neutral ridge (Figure 18.6). Each mutation takes the offspring to either one of two neighboring genotypes, such that they can reach the target genotype through two mutations where the first has fitness 0.7, or

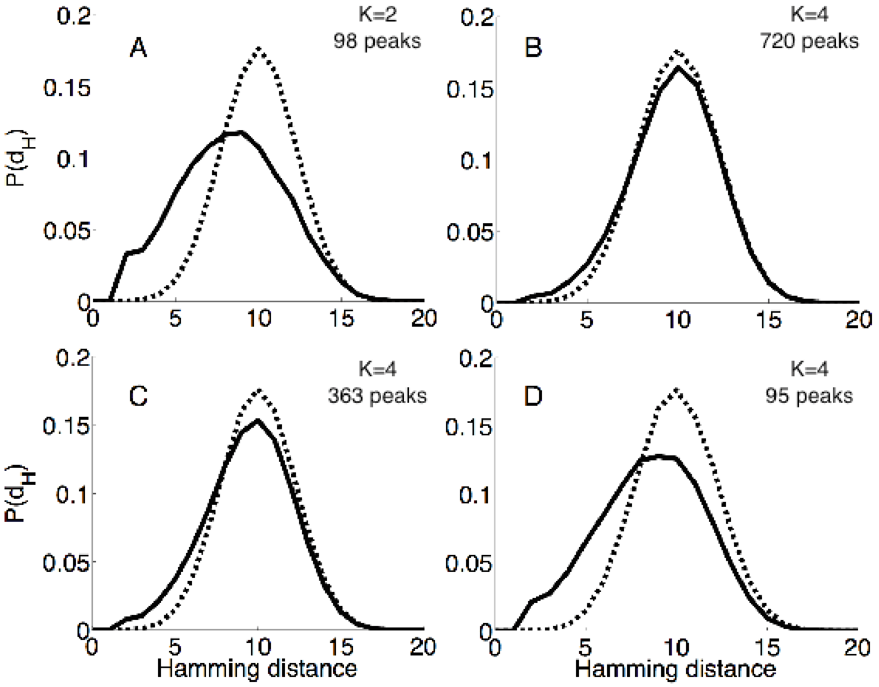


Fig. 18.2 Distributions of pairwise Hamming distances d_H between all peaks (solid) and between random control genotypes (dashed) in the NK landscape. The distributions shown are the averages of 50 different landscapes with genomes of length $N = 20$. (A) $K = 2$ landscapes containing an average of 98 peaks. (B) $K = 4$ landscapes containing an average of 720 peaks. (C) $K = 4$ landscapes including only an average of 363 peaks with a fitness above a threshold: $W \geq \Theta = 0.60$. (D) $K = 4$ landscapes including only an average of 95 peaks with a fitness above a threshold of $\Theta = 0.66$. As the samples include only the highest peaks, the pairwise distributions of $K = 4$ landscapes begin to resemble that of the $K = 2$ landscapes, suggesting that the highest peaks do cluster in genotype space, whereas the distribution of lower peaks is less biased. The control assigns genotypes to peaks by randomly sampling from all genotypes without replacement. Adapted from [38].

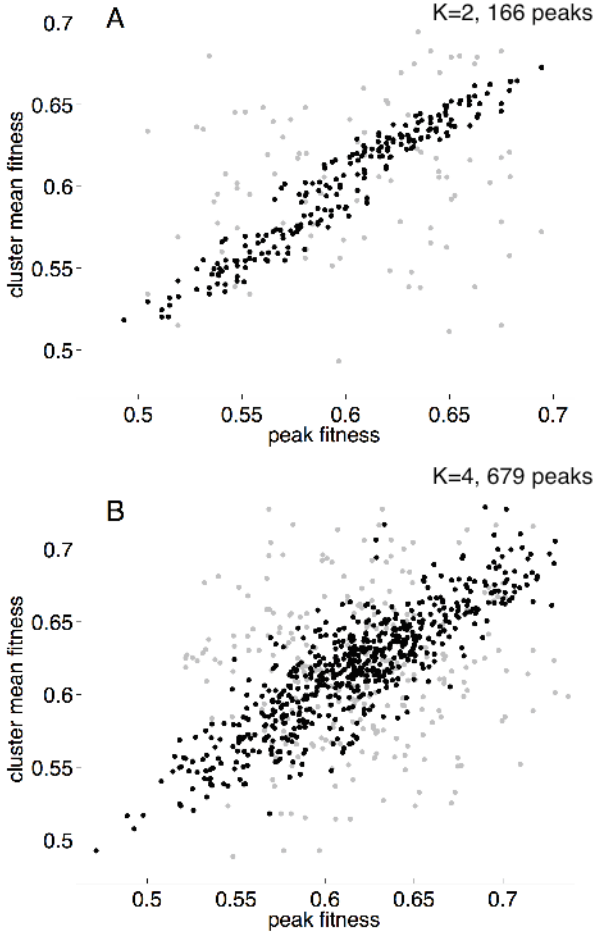


Fig. 18.3 Mean fitness of clusters of peaks as a function of peak fitness. The data is obtained comparing the fitness of each peaks with the average fitness of all other peaks within a Hamming distance of $d = 2$. (A) Landscape of $K = 2$ with 166 peaks (black dots). All landscapes show a strong correlation between cluster mean fitness and peak fitness, while the same analysis of assigning random genotypes without replacement to the peaks (but keeping the fitness) shows no such correlation (gray dots). The random data are from ten samplings. (B) One landscape of $K = 4$ with 679 peaks (black dots), and random genotypes (gray dots) obtained by sampling four times. Adapted from [38].

through four neutral mutations with genotypes of fitness 1. Results for a population size of 100 are shown in Figure 18.7, and clearly show that if the mutation-supply rate is not prohibitively low (as in the strong selection/weak mutation regime where mutations go to fixation individually) then populations can easily endure deleterious mutations necessary to cross valleys, and will do so rather than utilizing neutral ridges that require a larger number of neutral mutations.

18.4 Landscape Structure and Prediction

Predicting evolution is a notoriously difficult task [7], and the success of such an endeavor depends crucially on the parameters of the evolutionary process (e.g., mutation rate and population size for a non-recombining population) as well as the structure of the fitness landscape. If the mutation supply rate is very low and selection is strong, predicting evolution is comparably easier because the only stochastic component of the process lies in calculating the probability that a beneficial mutation (the one closest to the dominating type) achieves a sufficiently large clone size so that the fixation of that mutation becomes deterministic [19, 23, 37]. However, recent evidence suggests that this limit is rarely if ever achieved in nature [5] and that instead many beneficial, neutral, and deleterious mutations exist within realistic populations at the same time. Furthermore, selection on the deleterious mutations appears to be weak enough that such mutations can persist in large populations for hundreds or even thousands of generations [45] so that they can serve as stepping stones for further adaptive progress. In this limit, analytical methods fail to provide closed form solutions for the probability of fixation, in part because valley-crossing now becomes an important component in predicting evolutionary paths. It is precisely in this limit where landscape structure becomes a crucial element in predicting evolution.

The results described above suggest not only that high fitness peaks are more likely to be found near peaks of similar fitness, but they also imply that peaks of lower fitness permeate genetic space, so that it is possible to traverse the entire genetic space via paths that connect peaks separated only by a single deleterious mutation. While we do not suggest that such paths are more likely than those that involve possible neutral and beneficial mutations, the observation implies that the areas of fitness with high-fitness peaks are accessible from anywhere in the landscape. The picture painted by Gavrilets is in a sense complementary, where large neutral networks of low fitness provide access to the large neutral networks of high fitness peaks. But such a picture is neither necessary to understand adaptive progress (because even in the limit of strong mutation and weak selection deleterious mutations can persist for a very long time in the population), nor is such a picture likely, because evolutionary paths consisting only of neutral and beneficial mutations are vastly under-represented (compared to those that include valley-crossings) when peaks are clustered. Thus, while the possibility of valley crossings makes evolution much more difficult to predict, prediction is considerably helped if the local structure of the landscape is such that high peaks are near other high peaks, and that peak networks percolate genetic space (albeit only at low fitness).

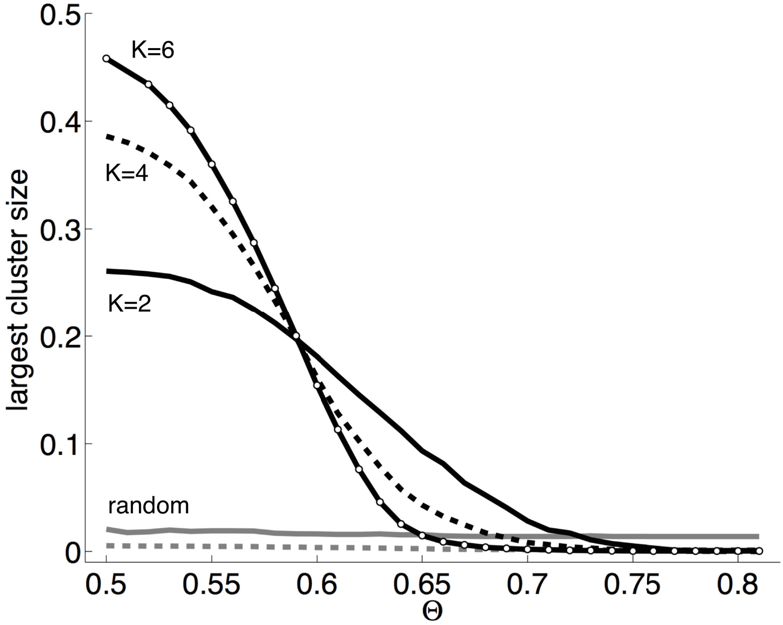


Fig. 18.4 Size of the largest network cluster in the landscape averaged over 50 landscapes for $K = 2$, 4, and 6 as a function of fitness threshold, Θ. Clusters consist of networks of peaks no further than Hamming distance $d = 2$ from each other. Only peaks with fitness above Θ are included in each sample. The more rugged the landscapes are, the more abrupt the transition is from small network clusters to one cluster dominating the landscape. Random genotypes for $K = 2$ (solid gray line) and $K = 4$ (dashed gray line) show no increase in cluster size as Θ is lowered, indicating that peaks cluster in a non-random manner. Adapted from [38].

18.5 Future Directions

While the concept and in particular the metaphor of the fitness landscape now has a long and distinguished history, we still have a lot to learn about the global and local structure of realistic landscapes with epistasis and pleiotropy. This is certainly true for computational models in which the fitness landscape is not explicitly designed, but it is particularly evident for empirical fitness landscapes, which have only recently begun to be investigated in the simplest cases. Because fitness landscape structure remains largely unknown, the predictive power that knowledge of that structure could bring has not yet been explored.

Here we have proposed that together with population size and mutation rate, fitness landscape structure is another key factor in enabling prediction of evolutionary outcomes. Despite the natural stochasticity of the evolutionary process, to a first approximation the trajectories that evolving populations take can be predicted with some degree of accuracy. The success of adaptation and the likelihood of speciation can be approximated given information about the aforementioned three key

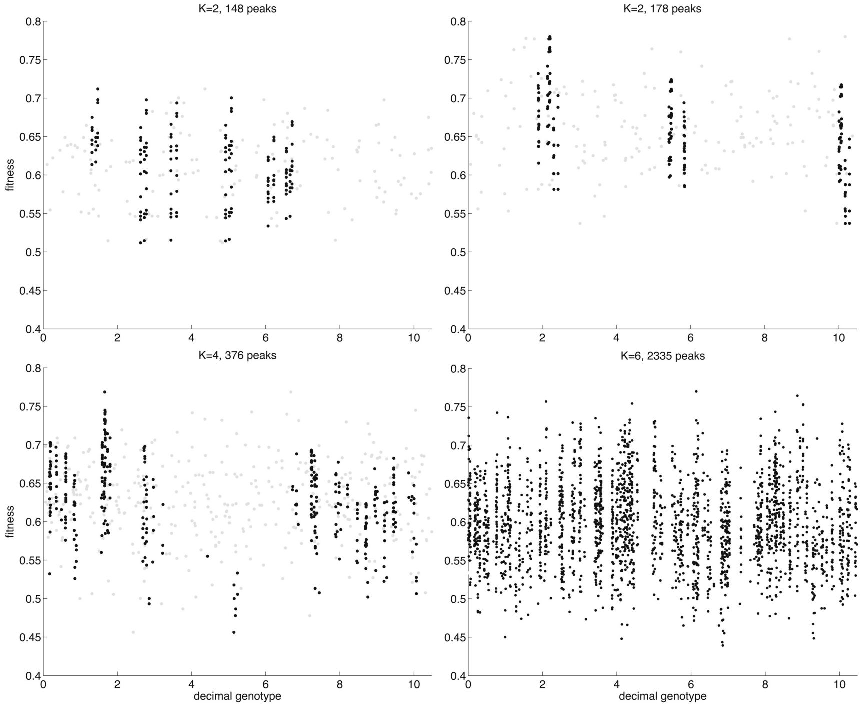


Fig. 18.5 Basins of attraction in the NK landscape. The bit-string genome of an NK organism can be translated into a decimal number, and peaks are here plotted with fitness as a function of this decimal genotype. Four different landscapes for $K = 2$ (top row) and $K = 4$ and 6 (bottom row) show that peaks cluster and huge voids exist with no peaks. Because neighboring genotypes can have very different decimal genotypes, not all adjacent peaks appear close to each other in decimal space. A single mutation at one locus can add a very high number to the decimal genotype, and so it is not immediately clear which peaks are close to each other in actual genotype space. Random reassignment of genotypes to peaks results in a random distribution in decimal genotype space (gray points in $K = 2$ and 4). Other kinds of fitness landscapes can be visualized in a similar fashion, even when the number of alleles is higher than two.

evolutionary parameters. Because of this prospect, various methods for visualizing critical properties of fitness landscapes are of great interest. The distribution of genotypes within populations depends not only on the supply of mutations, but also on the local structure of the fitness landscape; flat landscapes result in a wider distribution and thus a higher level of genetic variation than populations located on a high peak with steep slopes. The amount of genetic variation in turn affects the evolvability of the population, which is thus intimately tied to the structure of the fitness landscape.

In order to make predictions about adaptation and speciation using fitness landscapes, several features are of particular interest. As outlined above, the spatial

Fig. 18.6 Model for computing the probability that a population will traverse a valley rather than taking the path of a neutral ridge. The population starts at genotype S with fitness 1 and can either take the path via genotypes n–which all have fitness 1–or go through the valley via genotype V (with fitness 0.7). The simulation is stopped when the first individual reaches the target genotype T (we record which path the population took to get there).

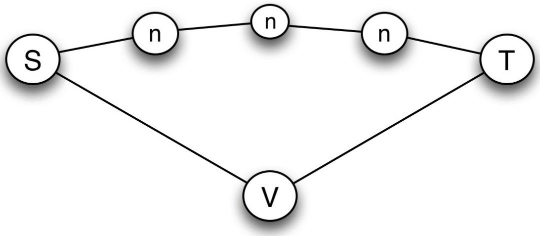
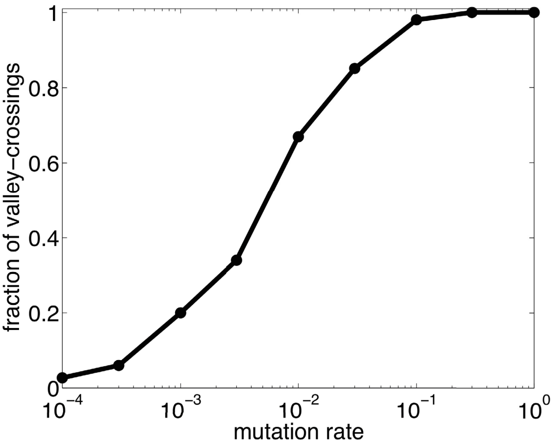


Fig. 18.7 Fraction of 100 simulations where the population took the path through a valley instead of a neutral ridge. When the mutation rate is not too low, valley-crossings are frequent. A population of 100 individuals is started with a genotype of fitness 1, and can evolve to the target genotype by taking a neutral path of length 4 or by going through a valley with fitness 0.7. The neutral ridge genotypes all have fitness 1, and traversing the valley requires only two mutations.



distribution of peaks in genotype space will affect evolutionary trajectories, so species-specific knowledge of the local structure must be obtained. While for some organisms it is evident that there are peaks in the local genotype or phenotype neighborhood, it is unclear if this is a general feature of all landscapes. It is possible that some or most organisms find themselves at a peak that is prohibitively far from other peaks. In such cases, if the landscape changes deprecating the occupied peak, the environmental change could lead to extinction before the population can respond adaptively. The extent of peak clustering in the local neighborhood thus anticipates the evolutionary outcome. Given that speciation can be driven by adaptation to specialized niches represented by distinct peaks in fitness landscapes [43], it follows that the ruggedness of the landscape and the number of peaks and their proximity to each other influences the likelihood of speciation. From this we can hypothesize a correlation between landscape ruggedness and the rate of speciation. We expect that the more rugged the landscape is, the higher the rate of speciation will be. It may be that too much ruggedness diminishes this effect when peaks are too numerous and close to each other, in which case valleys occupied by hybrids may not prohibit interbreeding, and thus lead to a breakdown of reproductive isolation in sexual species. In this case there would exist an optimum degree of ruggedness that maximizes the rate of speciation. If ruggedness could be estimated, it could be compared between clades that have different rates of speciation. Our expectation would then be that the rate of speciation is an increasing function of ruggedness up to a point where peaks become too numerous, leading to a breakdown in reproductive isolation. However, we currently have no way to estimate local fitness landscape ruggedness without genotyping and measuring fitness on a massive scale.

We investigated here the question of whether populations would be able to utilize the neutral ridges in a Holey Landscape [17] using a computational model (Figure 18.6), and show that even a large reduction in fitness (up to 30%) is tolerated if the mutation rate is not very low. An important variable here is the population size: If the population size is small, the mutation-supply rate may be so small that neutral paths cannot reliably be found. If the population size is large, then even very small fitness differences can be detected by selection, and thus even fewer paths are going to be neutral. It is possible that the available neutral paths in the limit of very high-dimensional genotype space are too few and far between so that bridging peaks along neutral ridges becomes much less likely than valley-crossing. Further investigation into the effect of the length of the neutral pathway is also warranted.

It is now clear [39, 45] that deleterious mutations can exist and segregate in populations for extended periods of time. We also know from experiments with microbes that populations experience beneficial mutations over long periods of time. Indeed, in over 50,000 generations of evolution in the Long Term Evolution Experiment (LTEE) [3], populations of *E. coli* continue to adapt with no sign of reaching a fitness plateau [48]. It Is not clear whether this evidence is consistent with the Holey Landscape model of fitness landscapes, in which beneficial mutations are abundant when the population is not at the highest fitness. If beneficial mutations were that easy to find, why are beneficial mutants still being found after over 50,000 generations of evolution? The evidence from the LTEE suggests, instead, that beneficial mutations are rare but continue to be available to lead to evolutionary progress.

Empirical fitness landscapes may turn out to have less in common with computational landscapes than we would have hoped, and it is therefore critical that empirical landscapes are investigated in greater detail. The common approach is to measure fitness of many genotypes a couple of mutations away from the peak, but this is unlikely to yield information about the ruggedness of the landscape. We therefore suggest that it would be valuable to measure the fitness of genotypes that are significantly removed from the wild-type (up to, say, 10 mutations away). This is impossible to do exhaustively because of the very large set of mutants, but selecting a subset to probe the fitness landscape in this manner could provide information about the fitness landscape pertinent to predicting evolutionary dynamics.

Finally, we acknowledge that real biological landscapes are not static entities as assumed in this work, but can change in both time and space. Evolutionary dynamics in non-static landscapes may be quite different from that of static landscapes [20, 34]. For example, pathogenic microbes experience changing fitness landscapes as the host continuously attempts to fight off the infection, and all organisms likewise may at times find certain traits to change from being advantageous to being sub-optimal because of environmental change. In these cases adaptation may be reduced to simply climbing the closest peak, as the previously optimal genotype has shifted, and the population finds itself in a position of low fitness. However, it is not known how frequently such change occurs, or how much it actually affects the fitness landscape. It may be that extreme cases of qualitative changes where peaks shift in genotype space are rare, and that most environmental change results in smaller deprecations of peaks, which has a less dramatic effect on evolution. It may be that many populations find themselves in effectively static landscapes for most of their evolutionary history. We therefore contend that knowledge of evolution in static fitness landscapes will still be relevant for our understanding of evolutionary dynamics.

Acknowledgements. We would like to thank Anurag Pakanati and Charles Ofria for extensive discussions on the structure of fitness landscapes. This work was was supported in part by the National Science Foundation's Frontier in Integrative Biological Research Grant No. FIBR-0527023 and NSF's BEACON Center for the Study of Evolution in Action, under Contract No. DBI-0939454. We wish to acknowledge the support of the Michigan State University High Performance Computing Center and the Institute for Cyber Enabled Research.

References

[1] Adami, C.: Introduction to Artificial Life. TELOS Springer Verlag, New York (1998)

[2] Adami, C.: Digital genetics: Unraveling the genetic basis of evolution. Nature Reviews Genetics 7(2), 109–118 (2006)

[3] Barrick, J.E., Yu, D.S., Yoon, S.H., Jeong, H., Oh, T.K., Schneider, D., Lenski, R.E., Kim, J.F.: Genome evolution and adaptation in a long–term experiment with Escherichia coli. Nature 461(7268), 1243–1247 (2009)

[4] Beerenwinkel, N., Pachter, L., Sturmfels, B., Elena, S.F., Lenski, R.E.: Analysis of epistatic interactions and fitness landscapes using a new geometric approach. BMC Evolutionary Biology 7(1), 60:1–60:12 (2007)

[5] Blount, Z.D., Barrick, J.E., Davidson, C.J., Lenski, R.E.: Genomic analysis of a key innovation in an experimental Escherichia coli population. Nature 489(7417), 513–518 (2012)

[6] Burch, C.L., Chao, L.: Evolvability of an RNA virus is determined by its mutational neighbourhood. Nature 406(6796), 625–628 (2000)

[7] Bush, R.M.: Predicting adaptive evolution. Nature Reviews 2, 387–392 (2001)

[8] Chou, H., Chiu, H., Delaney, N., Segrè, D., Marx, C.J.: Diminishing returns epistasis among beneficial mutations decelerates adaptation. Science 332, 1190–1192 (2011)

[9] Covacci, A., Rappuoli, R.: Helicobacter pylori: molecular evolution of a bacterial quasispecies. Current Opinion in Microbiology 1, 96–102 (1998)

[10] Doebeli, M., Dieckmann, U.: Evolutionary branching and sympatric speciation caused by different types of ecological interactions. The American Naturalist 156, S77–S101 (2000)

[11] Eigen, M.: Selforganization of matter and the evolution of biological macromolecules. Die Naturwissenschaften 58(10), 465–523 (1971)

[12] Eigen, M., McCaskill, J., Schuster, P.: Molecular quasi–species. The Journal of Physical Chemistry 92(24), 6881–6891 (1988)

[13] Elena, S.F., Lenski, R.E.: Test of synergistic interactions among deleterious mutations in bacteria. Nature 390, 395–398 (1997)

[14] Ellison, G.: Basins of attraction, long–run stochastic stability, and the speed of step–by–step evolution. Review of Economic Studies 67(1), 17–45 (2000)

[15] Fisher, R.A.: The Genetical Theory of Natural Selection. Oxford University Press, Oxford (1930)

[16] Franke, J., Klözer, A., de Visser, J.A.G.M., Krug, J.: Evolutionary accessibility of mutational pathways. PLoS Comput. Biol. 7(8), e1002134 (2011)

[17] Gavrilets, S.: Evolution and speciation on holey adaptive landscapes. Trends in Ecology & Evolution 12(8), 307–312 (1997)

[18] Gavrilets, S.: Fitness Landscapes and the Origin of Species. Princeton University Press, Princeton (2004)

[19] Gerrish, P.J., Lenski, R.E.: The fate of competing beneficial mutations in an asexual population. Genetica 102/103, 127–144 (1998)

[20] Hayden, E.J., Wagner, A.: Environmental change exposes beneficial epistatic interactions in a catalytic RNA. Proceedings of the Royal Society B: Biological Sciences 279(1742), 3418–3425 (2012)

[21] Hinkley, T., Martins, J., Chappey, C., Haddad, M., Stawiski, E., Whitcomb, J.M., Petropoulos, C.J., Bonhoeffer, S.: A systems analysis of mutational effects in HIV-1 protease and reverse transcriptase. Nature Genetics 43(5), 487–489 (2011)

[22] Jain, K.: Deterministic and stochastic regimes of asexual evolution on rugged fitness landscapes. Genetics 175, 1275–1288 (2007)

[23] Johnson, T., Barton, N.H.: The effect of deleterious alleles on adaptation in asexual organisms. Genetics 162, 395–411 (2002)

[24] Kaplan, J.: The end of the adaptive landscape metaphor? Biology & Philosophy 23(5), 625–638 (2008)

[25] Kauffman, S., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. Journal of Theoretical Biology 128(1), 11–45 (1987)

[26] Kauffman, S.: The Origins of Order. Oxford University Press, New York (1993)

[27] Kouyos, R.D., Leventhal, G.E., Hinkley, T., Haddad, M., Whitcomb, J.M., Petropoulos, C.J., Bonhoeffer, S.: Exploring the complexity of the HIV-1 fitness landscape. PLoS Genetics 8(3), e1002551 (2012)

[28] Kryazhimskiy, S., Dushoff, J., Bazykin, G.A., Plotkin, J.B.: Prevalence of epistasis in the evolution of influenza a surface proteins. PLoS Genetics 7, e1001301 (2011)

[29] Kvitek, D.J., Sherlock, G.: Reciprocal sign epistasis between frequently experimentally evolved adaptive mutations causes a rugged fitness landscape. PLoS Genetics 7(4), e1002056 (2011)

[30] Khan, A.I., Dinh, D.M., Schneider, D., Lenski, R.E., Cooper, T.F.: Negative epistasis between beneficial mutations in an evolving bacterial population. Science 332, 1193–1196 (2011)

[31] Lunzer, M., Miller, S.P., Felsheim, R., Dean, A.M.: The biochemical architecture of an ancient adaptive landscape. Science 310, 499–501 (2005)

[32] Martins, J.Z.R.: The exploration of HIV fitness landscapes. Ph.D. thesis, Universiy of Zurich (2012)

[33] McCandlish, D.: Visualizing fitness landscapes. Evolution 65(6), 1544–1558 (2011)

[34] Mustonen, V., Lässig, M.: From fitness landscapes to seascapes: non-equilibrium dynamics of selection and adaptation. Trends in Genetics 25(3), 111–119 (2009)

[35] Nosil, P., Harmon, L.: Ecological explanations for (incomplete) speciation. Trends in Ecology & Evolution 24(3), 145–156 (2009)

[36] Ofria, C., Wilke, C.: Avida: A software platform for research in computational evolutionary biology. Artificial Life 10, 191–229 (2004)

[37] Orr, H.A.: The rate of adaptation in asexuals. Genetics 155, 961–968 (2000)

[38] Østman, B., Hintze, A., Adami, C.: Critical properties of complex fitness landscapes. In: Fellerman, H., Dörr, M., Hanczyc, M.M., Ladegaard Laursen, L., Maurer, S., Merkle, D., Monnard, P.A., Stoy, K., Rasmussen, S. (eds.) Proc. of the ALife XII Conference, pp. 126–132. MIT Press (2010)

[39] Østman, B., Hintze, A., Adami, C.: Impact of epistasis and pleiotropy on evolutionary adaptation. Proceedings of the Royal Society B: Biological Sciences 279, 247–256 (2012)

[40] Pigliucci, M.: Adaptive landscapes, phenotypic space, and the power of metaphors. The Quarterly Review of Biology 83(3), 283–287 (2008)

[41] Pitt, J., Ferre-D'Amare, A.: Rapid construction of empirical RNA fitness landscapes. Science 330, 376–379 (2010)

[42] Richter, H.: Coupled map lattices as spatio–temporal fitness functions: Landscape measures and evolutionary optimization. Physica D 237(2), 167–186 (2008)

[43] Schluter, D.: Ecology and the origin of species. Trends in Ecology & Evolution 16(7), 372–380 (2001)

[44] Turelli, M., Barton, N.H., Coyne, J.A.: Theory and speciation. Trends in Ecology & Evolution 16(7), 330–343 (2001)

[45] Weissman, D., Desai, M., Fisher, D.: The rate at which asexual populations cross fitness valleys. Theoretical Population Biology 75, 286–300 (2009)

[46] Whitlock, M.C., Phillips, P.C., Moore, F.B.G., Tonsor, S.J.: Multiple fitness peaks and epistasis. Annual Review of Ecology and Systematics 26, 601–629 (1995)

[47] Whitlock, M.C.: Founder effects and peak shifts without genetic drift: Adaptive peak shifts occur easily when environments fluctuate slightly. Evolution 51(4), 1044–1048 (1997)

[48] Wielgoss, S., Barrick, J.E., Tenaillon, O., Wiser, M.J., Dittmar, W.J., Cruveiller, S., Chane-Woon-Ming, B., Médigue, C., Lenski, R.E., Schneider, D.: Mutation rate dynamics in a bacterial population reflect tension between adaptation and genetic load. Proceedings of the National Academy of Sciences 110(1), 222–227 (2013)

[49] Wiles, J., Tonkes, B.: Hyperspace geography: Visualizing fitness landscapes beyond 4D. Artificial Life 12, 211–216 (2006)

[50] Wilke, C.O.: Probability of fixation of an advantageous mutant in a viral quasispecies. Genetics 163, 467–474 (2003)

[51] Wright, S.: Evolution in Mendelian populations. Genetics 16(2), 97–159 (1931)

[52] Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones, D. (ed.) Proceedings of the 6th International Congress on Genetics, vol. 1, pp. 356–366 (1932)

[53] Wright, S.: The shifting balance theory and macroevolution. Annual Review of Genetics 16, 1–19 (1982)

Part V
Outlook and Afterthoughts

Chapter 19
Frontiers of Fitness Landscapes: A Summary of Open Questions

Hendrik Richter

Abstract. In this chapter we briefly look at possible future issues in fitness landscape research. We discuss challenges to a fitness landscape approach that result from recent experimental and theoretical findings about the information transfer in biological systems. In addition, we set out opportunities these results may open up and speculate about directions that landscape research may take. We summarize the discussion by laying out eight open questions.

19.1 Challenges and Limitations

In the other chapters of this book recent advances in the theory and application of fitness landscapes are presented. These chapters review each with its own focus and perspective the current state of understanding and also point at open problems and possible future directions. This chapter does not aim at repeating these detailed discussions. Accompanying these discussions, we here rather intend to focus on issues and ideas that were not explicitly addressed in other chapters of this book and give a more general summary of potential fields in future fitness landscape research. In the following, we will slightly deviate from the convention to initiate the discussion of what we wish (or need) to find out by laying out what we actually do know. Acknowledging that the current understanding is given in the other chapters of this book, we will start this outlook on open questions with a concise discussion about the current and fundamental limits and challenges of the fitness landscape approach. This goes along with our conviction that reporting research without identifying promises and possibilities as well as limitations and impossibilities is incomplete.

Hendrik Richter
HTWK Leipzig University of Applied Sciences, Faculty of Electrical Engineering
and Information Technology, D–04251 Leipzig, Germany
e-mail: richter@eit.htwk-leipzig.de

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_19, © Springer-Verlag Berlin Heidelberg 2014

Fitness landscapes are built by the relations between the genetically coded possibility (genotype), the actually realized biological system (phenotype) and the reproduction and survival success (fitness). These relations evoke genotype–to–fitness maps with imprinted potentials for evolutionary dynamics. Such maps are the immediate result of combining the principles of Darwinian evolution with the fundamentals of genetics in the so–called modern synthesis. As the understanding of evolutionary aspects of molecular biology has meanwhile moved significantly beyond the modern synthesis, the fitness landscape approach needs to reassess its defining ingredients: genotype and fitness. Broadly speaking, there are two fundamental questions. One is what the genotype actually codes for, the other is what fitness really signifies. We intend to approach these two questions in the light of recent findings.

Since the middle of last century we saw a deepening in the understanding of the biochemical mechanisms of information transfer and inheritance in biological systems that led to the genetic code of molecular carbon and protein based life. Starting with solving the DNA structure by Watson and Crick in 1953, this development came to a supposed culminating point about 50 years later with a complete decoding of the human genome (and genomes of other life forms as well). It seemed that by obtaining the entire genome, the presumed building plan of the actually realized biological system, we can have a match between the genetic code and corresponding biological structures and processes. Translating these structures and processes to abilities, traits and features and these features, traits and abilities to reproduction and survival success would, in principle, construct complete phenotype–to–fitness maps. Generalizing all possible genetic codings would, in turn, result in genotype–to–fitness maps as employed by the fitness landscape approach.

However, along the accumulation of experimental and theoretical findings came the growing recognition that sequencing the DNA alone is not giving the insight anticipated about the dynamical process of rendering a phenotypic realization (a life form with prescribed features and abilities) out of a genotypic specification (its DNA). It rather appears that the genes coded in the DNA do not statically define and specify phenotypic realizations themselves, but the interactions that the genes enable and evoke are dynamically creating developmental possibilities (and probabilities) for phenotypic realizations. A main argument for the interactions enabled by the DNA (which can be seen as algorithmic processes [69]) being more important than the DNA data themselves is the realization that the information content of the DNA (expressible as the number of base pairs) does not correlate with the complexity of the biological system that the DNA describes [14, 22, 23]. For instance, the data stored in mammals' genomes (including the human genome which has about 3.2 billion base pairs) is much smaller than the genomic size of some flowering plants and amphibians. Generally we find that mammals' genome size does not vary very dramatically. If we take the genome size of humans as scale, the dolphin's genome, for example, has 80.3% of the humans' genome size or the elephant's 101.9%. Compare to this today's largest known animal genome, which is that of an amphibian called the marbled lung fish (*protopterus aethiopicus*) and is 44.3 times the length of the human genome and the currently largest known genome, which belongs to the Japanese canopy plant (*paris japonica*) and is 50.7 times as large [23, 42]. Nevertheless, it seems to be absurd to argue that these plants or amphibians have biological structures and behaviors 40 or 50 times as complex as that of mammals.

The view that the DNA gives a unique specification of phenoypic realizations is summarized by the so–called central dogma of molecular biology [9]. The dogma states that the flow of genetic information is exclusive and unidirectional from DNA via RNA to protein (and hence also specifies the upper organizational levels of complex biological systems from RNA networks and sub–cellular machinery via cells, tissues and organs to the organism itself). There are further implications of this view. Information–transferring and –processing functionality can be generally described as to emerge from data plus an algorithm acting on these data (or an algorithm plus data processed by the algorithm). As biological information transfer is central to the understanding of evolutionary dynamics, this functionality also applies to biological systems. Taking this view, the central dogma implies that the DNA incorporates not only the data required to specify a protein, but also the algorithm that processes these data and brings the protein into being. In other words, the DNA does contain all the information that is needed to reconstruct an organism. It turned out that this view is at least an oversimplification [37, 38, 56, 69]. Over the last 20 years several microbiological information–processing activities have been discovered and studied that do not fit into this view of DNA being both data and processing information and are even outrightly forbidden according to the central dogma.

Amongst these "classically forbidden" processes the different forms of so–called mobile genetic elements are particularly interesting [56, 57, 64, 70]. Two forms, horizontal gene transfer and the effect of transposable elements (also known as transposons) have impact on the understanding of genotype–to–fitness maps beyond the modern synthesis. Horizontal transfer refers to the process of transmission of genes or genetic material from one organism to another in nonhereditary mechanisms. Horizontal transfer complements vertical transfer which is a generational transmission of genes from parents to offspring by (sexual or asexual) reproduction. The horizontal transmission is subsequently expressed, meaning that the exchange on the level of the genotype also alters the phenotype of the receiving organism. It is known since a rather long time that such an exchange of genetic material during the life time (and hence nonhereditary) occurs in bacteria and other simple single–celled micro–organisms [63]. Lately, it became apparent that horizontal gene transfer also plays an important role in more complex multi–cellular organisms [5, 31, 81]. In addition, horizontal transfer intermingles with the effect of transposons.

An established consequence of the fact that complexity of the biological system and the information content of its DNA (as measured by the number of base pairs) does not correlate is that the genes that actually code for proteins are only a (more or less large, or small) fraction of the whole genome. Next to coding DNA we have what is called non–coding DNA, sometimes (inappropriately) dubbed as junk DNA. The DNA further consists of sequences whose elements are transposable. These transposable elements are also non–coding and have the property to be able to change position within the genome, which inspired to name them jumping genes. Transposable elements can have the effect of causing or reversing mutations on the DNA and also altering the genome's size. Also, transposable elements can be horizontally transfered [52, 64]. Moreover, there are strong hints that the horizontal transfer of transposable elements is a major force in genome evolution and drives the spread of genetic innovation [17, 64, 70]. In other words, nonhereditary processes can cause moves in the genotypic space during the lifetime of a phenotypic realization. If these moves affect reproduction and survival success, the distribution of fitness over genotypic space is no longer constant.

Apart from these classically forbidden processes, there is another complication to the exclusive and unidirectional information flow from DNA to organism which comes into being by the role the RNA plays in this process. Meanwhile, we know that the multitude of different RNAs is so large and their functions are so diverse that these RNAs are sometimes playfully called as belonging to an RNA zoo [66]. The translation process from DNA via RNA to protein is highly complex. According to the central dogma of molecular biology, the main function of RNA is to transform the information stored in the DNA into proteins in a rather passive process. The RNA is mainly seen as a carrier of coding information. Again, this appears to be overly simplistic [15, 66, 77]. Next to the function of reading, decoding and converting of information stored in the DNA, the RNAs perform a significant number of regulation processes, for instance to determine which genomic regions are accessible, to activate or silence promoters to control the activity of genes, to switch on and off the transcription and translation process of proteins, or to repair the genome by compensatory mutations. These regulation processes are influenced by and interact with the molecular environment in which the translation process from DNA via RNA to protein takes place. The molecular environment, in turn, is stimulated by the organism it belongs to and ecological processes the organism is exposed to. As a consequence, there is a coupling between evolutionary processes on a molecular level and ecological processes on a macroscopic level. The way the organism experiences its ecological environment influences which DNA–to–protein translation processes occur. The results of the translation processes, on the other hand, drive phenotypic features and potentially alter the organism's abilities to cope with the ecological environment. Recent experimental and theoretical results suggest that these features of the RNAs promote the merger of evolutionary and ecological timescales and are responsible for the rapid occurrence of evolutionary novelty in biological systems [3, 24, 41, 53, 67]. We also refer to Chapter 12 of this book by Richard Watson and Marc Ebner for further discussion of eco–evolutionary dynamics.

For a fitness landscape view the versatile functions of RNAs and the coupling between evolutionary and ecological processes means that the same genotype may have different phenotypic realizations with possibly different fitness values. This also means that it may be useful to consider that fitness has spatial and temporal granularity [61]. The spatial scale (with given granularity) is in phenotypic space and reflects the fact that a genotype may be polysemous and hence implies equivocal but summable fitness. The temporal scale reflects the interdependencies that result from the merger of evolutionary and ecological processes in specifying a phenotype and its fitness.

To conclude the concept that the DNA uniquely specifies a phenotypic realization is at least incomplete and most likely a gross oversimplification. This also implies that a fitness landscape approach in the simple form as sketched at the beginning of this section is questionable. Provocatively, it could be asked if there is a genotype–to–fitness map at all. We next review how these results may open up chances for future developments in fitness landscape research.

19.2 Chances and Directions

In the last section we have discussed some recent findings about the way genetic information is processed in the lower organizational levels of complex biological systems. From these findings it can be concluded that neither the central dogma of molecular biology nor the modern synthesis of evolutionary biology are sufficient theoretical frameworks for truly understanding evolutionary processes [20, 21, 56, 57]. On a more general level, these deficits may also be responsible for alternatives to a (bio–)chemistry–centered view of evolutionary processes gaining in importance. According to these alternatives, evolution is foremost a physical process and life is but one manifestation of this process [12, 21]. Evolution defines life, but life is an exemplification of evolution. To understand the physical process of evolution it is necessary to describe its dynamics and hence to identify the driving forces that cause this dynamics. In physics, on a fundamental level, the forces that enable dynamics are the result of symmetry breaking [1, 6]. The same could (or even should) apply for the forces underlying evolutionary dynamics. There are some proposes to formulate the emergence of topological features in fitness landscapes as a symmetry breaking process, or instance for equivocal genotype–to–phenotype maps using the concept of effective fitness [61] or in describing how in evolutionary computation crossover exploits certain search space properties [45, 48]. However, these proposes do only start to include some of the highly complex processes that specify phenotypic realizations as considered in the last section, see for example [49, 50, 51]. To extend these ideas appears to be promising and we next briefly outline the link between symmetry breaking and fitness landscapes.

From the perspective of a phenotype, several genotypes may be suitable to dynamically provide genotypic traits that enhance survival and reproduction success in a given ecological environment [2]. From the perspective of a genotype, several phenotypes might be the outcome from the way the complex transcription and translation process of proteins interacts with environmental conditions. Moreover, the ecological environment itself undergoes its own dynamic processes. In the absence of all environmental conditions, all genotypes may have the same survival and reproduction fitness. The genotypes can be seen as symmetric. The environmental conditions and their dynamics set what fitness means and signifies; the genotypic symmetry is broken. The symmetry breaking hence cast (or recast) a non–trivial fitness landscape and formulates the driving forces of evolutionary dynamics. This, however, at the same time defines (or redefines) the genotype–to–phenotype map in a complex process that has its own dynamics and is assumed to be subject to driving forces other than pure selective pressure; see as an example protein evolution [80]. There is evidence that changing environmental conditions may trigger these transcription and translation processes (and hence set the genotype–to–phenotype map) [3, 24, 67]. However, the transcription and translation processes are not mappings of environmental dynamics in any trivial way and follow their own developmental principles, for instance they obeys the laws of physics and chemistry [55, 80] and are subject to self–organization processes [27, 29, 75].

Despite all that, it remains an undisputed fact that the genes of an organism play a defining (but not the sole) role in causing biological reality and driving evolutionary dynamics. The recent findings discussed in the last section mainly highlight that the genome encoded in DNA and hereditary information in general are not alone sufficient to reconstruct an organism [37, 38, 69]. Hence, we need to specify mechanisms other than random mutation, genetic recombination, natural selection and drift to explain evolutionary changes. There is no doubt that whatever the evolutionary variation processes are, success or failure of an evolutionary innovation is decided by adaption to the environment. To be adapted well foremost means to have traits and abilities to compete and/or cooperate to maximize the utilization of resources, but also to cope with shifts in climate or other physical attributes of the living space, and to react on alterations in the food and/or prey abundance, or predator and/or parasites existence. All this creates pressure that in the end is realized by natural and sexual selection mechanisms. This is fitness. Clearly, for this process there must be a relation between the genetically possible (genotype) and fitness. However, this relation is characterized by the complex processes to render phenotypic traits and their effects on fitness. Having rebuffed doubts about the existence of genotype–to–fitness maps, the more interesting question is what their mathematical structures and properties are. We think that the ideas of symmetry breaking might be helpful for approaching this question.

In a way this discussion is closely related to some of the points made by Stuart Kauffman in his foreword of this book. Roughly summarized, he said that there are no statable laws for describing the evolution of the biosphere, and hence all attempts to model evolutionary dynamics, for instance by fitness landscapes, are likely to come to nothing. In some ways, we think that Kauffman's arguments are strong. In all likelihood, there is no underlying prestatable plan for evolution, let alone for describing evolution spatially and temporally complete. We personally think that having and solving such laws were to border creationism. Besides, even if we restrict ourselves to non–living matter, we know that even if we have all the laws of physics, and all positions and velocities of particles, we cannot predict their future development spatially and temporally complete. There are several physical limits that reject this Laplacian idea. However, the limitation can be partly remedied by relaxing spatial and temporal range. Spatially and temporally local, physics and chemistry can describe and predict processes in non–living matter rather well. It is an interesting question to what extend this approach is transferable to evolutionary processes.

Evolution is open–ended, evolution has no goal or solution, and evolution is not optimum seeking in any strict mathematical sense. In such a sense, we were to have clearly defined input–output–relations between what can possibly be coded by an information carrier genetically propagable forward in time (input) and what advantages in terms of survival and reproduction success a specific coded instance has in a given (and all possible) biospherical environments (output). Evolution as a strict optimum seeking process would mean to find (and hang on to) genetic specifications that yield highest advantage. The discussions so far about what we know now suggest that such input–output–relations are seemingly impossible to set up for evolution of a sufficiently complex biosphere. Even the starting and the end point of this input–output–relation is fuzzy. To enumerate all possible codings over a DNA string (just to remain with what we know now) would exceed the information processing capacity of the universe. To complicate matters even more, there are nonhereditary mechanisms of gene transfer and the complex and only superficially understood rules for the members of the RNA zoo to read the DNA and to specify proteins out of the reading process. This leads to a combinatorial exploding space of possibilities for how forms and functions in living entities could look like. At the end point, the allocation of fitness to living beings is a highly debatable affair for a given biosphere. To do the same for any possible biosphere would require to specify not only all biospheres but also the effects the biospheres have on the fitness of all its members, which again exceeds any possible information processing. Compare to this also the discussion by Edward Weinberger in the afterthoughts of the book.

There is another argument for evolution not solving an optimization problem in any conventional sense, which comes from the no free lunch theorem of optimization [10, 13, 73, 79]. The theorem states that evaluated over all optimization problems, there is no algorithm that performs better than others with respect to given evaluation criteria. In other words, averaged over all possible fitness landscapes, no mechanism finds the landscape's peaks more effectively than all the others. If this is so then it appears to be puzzling that in natural evolution we should only find evolutionary dynamics triggered by random mutation and natural selection (plus genetic recombination and the other mechanisms that specify the transcription and translation of phenotypic traits). If we postulate that the biological systems observable are evidence of success, then we imply that the search mechanism of natural evolution works particularly well for the fitness landscapes that pose the problem of natural evolution. On the other hand, and with the no free lunch theorem in mind, this means that the fitness landscapes of natural evolution can only be a tiny fraction of the prodigious variety of all possible fitness landscapes that the laws of nature could pose. According to an argument by Kauffman [30], this can only mean that the fitness landscapes that underly natural evolution must be tailored by evolution itself. Evolution molds the fitness landscapes in which the mechanisms that cause evolutionary dynamics perform rather well. This hardly qualifies as optimization in a strict mathematical sense.

In this context, another recent contribution to the discussion on how biological organisms deal with (possibly changing) environmental conditions is of interest: the concepts of evolvability and facilitated variation. Evovability [4, 35, 68] means the potential to generate novel features and abilities by genetic changes that maintain survival and reproduction success in future environments. Facilitated variation [32, 40] tries to explain the basic mechanisms for purposeful phenotypic variations as the result of random genetic mutations. This is centered around the fact that small genetic mutations might lead to substantial phenotypic changes, which is a way to explain why small scale mutations can have a considerable impact on phenotype even over a small window of generations. The main statement of facilitated variation is that organisms have intrinsic genetic and non–genetic structures and processes allowing to generate substantial and purposeful phenotypic changes by small scale genetic shifts. On the level of phenotypic traits we find a substantial degree of evolutionary reuse of functional components [57]. In other words, the landscapes underlying natural evolution facilitate evolvability by restricting and directing the effects a genotypic transition can have in the phenotypic space. As discussed before, there are hopes that evolutionary dynamics can be described by establishing rules that are valid locally. Local may mean spatially and/or temporally local. This means that evolutionary processes can be understood as to take place on a succession of (spatially and temporally local) fitness landscapes. Facilitated variation, on the other hand, implies that the succession more likely follows paths that generate landscapes with features searchable by the mechanisms of natural evolution.

As provoking as this may sound, the recent challenges to a fitness landscape approach in the simple form as defined by the modern synthesis do not limit our ability to seek for explanation of evolutionary processes. Evolution is not random and nothing else [7, 8]. Quite the contrary, the results reviewed in this section show that it may be possible to understanding evolution as a physical, information–propagating and algorithmic process [5, 21, 69]. In the next (and final) section, we will summarize the discussion of challenges to and potential for fitness landscapes methods by listing some open problems.

19.3 Summary and Outlook: Eight Open Problems

In the previous section we have briefly discussed some main approaches that research into fitness landscapes could follow. In doing so we have focused on issues and ideas that were not discussed explicitly elsewhere in other chapters of this book. We now summarize the discussion, seize ideas and suggestions given in other chapters of the book, and conclude this chapter with laying out a couple of open problems. Clearly there are more open problems connected to fitness landscapes nameable as the ones listed here. For instance, open problems connected with characterizing fitness landscapes by landscape measures are left aside for there is a comprehensive recent survey and outlook by Malan and Engelbrecht [33]. Moreover, we would like to single out the following questions because we belief that clarifying these questions would particularly promote the understanding of some fundamental issues in evolutionary dynamics.

One. How do the different levels of meaning and significance of fitness interfere with defining evolutionary dynamics?

As pointed out in previous section's discussions, the functionality and success of phenotypic traits is judged by the adaption to environmental conditions and the ability to cope with the selection pressure that results from it. The phenotypic traits aggregate to individual quality [76] in surviving and reproducing, which in turn correlates positively with fitness. However, assigning fitness to an actual biological (or even microbiological) entity is a highly debatable matter [76, 78] (also see Section 1.2 of Chapter 1). As fitness is often considered to be a scalar abstraction of multiple phenotypic traits, it can be seen as a metric on a highly complex data set. In other words, what we call fitness is most likely a proxy and the result of an abstraction, aggregation and interpretation process. Depending on the parameters of this process, this goes along with fitness that may have spatial and temporal granularity [61]. All this leads to the practice of defining fitness as an axiomatic property of the landscape. The question is not if this practice is suitable, because for complex genotypic and phenotypic spaces there seems to be no sensible alternative to it. The more interesting question is how the different levels of meaning and significance of fitness that come with the abstraction, aggregation and interpretation process interfere with defining evolutionary dynamics. In this context, also the aforementioned information flow in biological systems is of interest. For instance as the question of what biological information is needed to specify fitness.

Two. How can recombinational transitions between genotypes be described in landscape dynamics?

An important feature of fitness landscapes is that it allows to describe evolutionary dynamics by transitions between genotypes. A succession of evolutionary transitions gives raise to evolutionary trajectories. The transitions come into being by evolutionary variation that may be asexual or sexual (or induced by mobile genetic elements, see question Three). Asexual variation results in mutational trajectories; sexual variation creates recombinational trajectories. Mutational trajectories through genotypic space can be described by Gillespie's strong selection/weak mutation (SSWM) model [18, 71]. In fact, most of the results on fitness landscapes are for mutational landscapes (see for instance the discussions in Chapter 6 and 7 of this book). For recombinational transitions, results are more sparse. Although it has been shown that for homologous recombination (exchange between two similar or identical molecules of DNA), a treatment similar to mutational landscapes is possible [19, 59, 60]. For non–homologous recombination, however, the resulting genotypic space seems to be no longer metric [58, 62]. Possibly, there is a need for an alternative description of the genotypic space [16, 26]. As sexual variation of genetic material is an important process in evolution, recombinatorial trajectories and hence recombinatorial fitness landscapes are an open research topic.

Three. How to include evolutionary transitions causes by mobile genetic elements, in particular by horizontal gene transfer, in fitness landscapes?

Recent findings about the way genetic information is processed in the lower organizational levels of complex biological systems (as discussed in Section 19.1) have highlighted the importance of another type of evolutionary transition process: those induced by mobile genetic elements, in particular by horizontal gene transfer. There are some examples for describing the evolutionary dynamics that comes with horizontal gene transfer [11, 36, 39, 47]. To some degree these works also address implications on fitness landscapes. On a more general level, the question of including evolutionary transitions causes by mobile genetic elements, in particular by horizontal gene transfer, in fitness landscapes is unsolved.

Four. What predictions about evolutionary dynamics can be derived from a given fitness landscape?

For a landscape with a given (non–changing) structure its topological features (the hills, valley, plains and ridges in the landscape metaphor) result in potentials for driving forces for evolutionary dynamics (the lakes and flows in the landscape). By assuming differences in fitness over genotypic space, the movement goes towards an aim. This defines possible evolutionary pathways that are directed and produces evolutionary outcomes that are more likely than others. This suggests that the topology and structure of a fitness landscape is a predictor of a likely evolutionary path. On a conceptional level, the landscape defines likely paths [72]. On a practical level, predicting an evolutionary path from a given landscape is still notoriously difficult; compare, for instance, the discussion in Chapters 4, 6 and 18. To enhance our abilities to predict likely evolutionary paths for a given landscape by improved or new algorithms and techniques remains an open topic in fitness landscape research. This question is of immense relevance in both evolutionary biology (for predicting paths evolutionary processes are likely to take) and evolutionary computation (to predict the optimization performance of an evolutionary search algorithm). Possibly these abilities can even be extended to cases where the landscape's structure is not known beforehand. Also, it is promising to further support our abilities to predict evolutionary dynamics in a fitness landscape by visualization tools and methods (see for instance [34, 74] and the discussion in Chapters 17 and 18).

Five. What value and significance have empirical fitness landscapes for understanding realistic evolutionary paths?

Given the understanding that a landscape's topology specifies potential evolutionary paths, another interesting question is whether the topology and structure of the fitness landscape at hand really matches the biological reality it supposes to study. A recent contribution to this question is empirical fitness landscapes. Such empirical landscapes are obtained by carrying out step–wise mutational evolution of molecular functions. This implies that molecular evolution allows to measure the fitness effect that experimentally induced mutations have [28, 43, 65] (also see the discussions in Chapters 1 and 18). The value and significance of empirical landscapes is still far from being clear. Some experiments have shown that empirical landscapes share important topological features known from computational landscapes, while some other experiments found not much agreement and applicability. Whether this is due to the restriction of low dimensionality that current experimental techniques and methods allows or points at some more fundamental problems is open. A further issue is comparability of empirical landscapes that result from different experiments. For computational landscapes there is some success in evaluating landscapes by using landscape measures. Possibly, these measures can be modified to become meaningful for empirical landscapes. To advance our knowledge about designing, conducting and evaluating experiments to create and analyze empirical landscapes is another important topic.

Six. What driving forces cast relevant fitness landscapes?

In the last section we have discussed that there are strong arguments for natural evolution creating and recasting the fitness landscapes in which evolutionary dynamics takes place. This means that not only the landscape's topology is changing continuously, but also that the basic structure of the landscape is not constant. A possible way to describe such features might be spatially and temporarily local landscapes. We have further discussed that on a fundamental level the driving forces that cause evolutionary dynamics can be thought of as to be the consequence of a symmetry breaking process. Possibly the idea of symmetry breaking discussed in Section 19.2 can be stretched as far as to describe the transitions between spatially and temporarily neighboring local landscapes. Locally valid landscapes can also be understood as subspaces of a global landscape, as a kind of collapsed genotypic space. The concepts of evolvability and facilitated variation imply that the succession of (spatially and temporally local) landscapes is not as random as it could be by just considering the multitude of possibilities the high–dimensionality of the genotypic space creates. All these ideas could contribute to deepen our understanding of the driving forces that cast relevant fitness landscapes and hence cause evolutionary dynamics.

Seven. How can the understanding of coevolution be promoted by a landscape view?

If we are not only considering an isolated species, but attempt to describe the evolutionary dynamics of interacting populations, we end up with coevolutionary scenarios. Coevolution also occurs as the result of the merger of evolutionary and ecological timescales in eco–evolutionary dynamics. We refer to the discussion in the Chapters 10, 11, 12 and 13. It is our understanding that fitness landscapes are a tool for describing the link between the driving forces of evolution and the actual evolutionary dynamics. This view should extend to coevolution between populations. Coevolutionary fitness landscapes are deformable and time–dependent (compare Section 12.3 of Chapter 12). It has been suggested that the deformation of the fitness landscape that combine two populations can be understood as codynamics between the fitness landscapes of each population (see Section 10.3 of Chapter 10). It remains open and is an open problem if and how the understanding of coevolution can be promoted by such or another landscape view.

Eight. What mathematical properties have realistic genotype–to–fitness maps?

Fitness landscapes are about the relationships between the genetically coded possibility (genotype), the actually realized biological system (phenotype) and the reproduction and survival success (fitness). Recent results (as reviewed in Section 19.1) about the biochemical mechanisms of information transfer and inheritance in biological systems suggest that these relationships are much more complicated than initially assumed. Research of fitness landscapes should meet these challenges and clarify which consequences are connected with these findings. We think that a central issue is the mathematical properties of the genotype–to–fitness maps. For deriving these maps realistically, new approaches are needed. To look for all–compassing laws and prestatable plans for evolution will not do. This was, for instance, impressively stated in Stuart Kaufman's foreword. Studying living matter cannot be based on a small number of fundamental laws expressible by rather simple equations. Thus, in want for anything else spatially and temporarily local rules in connection with a computational approach may be more suitable. Possibly, the idea of symmetry breaking can be utilized and approaches such the interaction–centered, signal–and–boundary modeling as suggested by Holland [25] might also be useful. In addition, it would be desirable to have additional input to the discussion from computational landscapes. It has been clear for a long time that exhibiting life–like, realistic evolutionary behavior requires computational models that have structures and fitness that evolve themselves [46]. Such computational models could be viewed as targetless evolutionary algorithms. Dynamic and deformable fitness landscapes most likely are starting points for understanding evolving fitness. How to mathematically model landscape structures that evolve is for now far from being clear.

Currently we can observe another wave of scientific revolution in biology and a "revival of the landscape paradigm" [54]. This implies that fitness landscapes as a method can and should incorporate the findings this revolution yields and remain an important concept and computational tool for retrospective explanation and guiding experimental setups. We hope that the eight questions laid out in this section will promote discussions about interesting topics in fitness landscape research. This is related closely with the wish to stimulate further understanding of evolutionary dynamics and in the end maybe add another little step to answering the question of how matter organizes (or self–organizes) into entities that are evolvable and evolving.

References

[1] Anderson, P.W.: More is different: Broken symmetry and the nature of the hierarchical structure of science. Science 177, 393–396 (1972)

[2] Bailey, N.W.: Evolutionary models of extended phenotypes. Trends in Ecology & Evolution 27, 561–569 (2012)

[3] Becks, L., Ellner, S.P., Jones, L.E., Hairston Jr., N.G.: The functional genomics of an eco–evolutionary feedback loop: linking gene expression, trait evolution, and community dynamics. Ecol. Lett. 15, 492–501 (2012)

[4] Borenstein, E., Meilijson, I., Ruppin, E.: The effect of phenotypic plasticity on evolution in multipeaked fitness landscapes. Jour. Evolut. Biology 19, 1555–1570 (2006)

[5] Boto, L.: Horizontal gene transfer in evolution: facts and challenges. Proc. R. Soc. B 277, 819–827 (2010)

[6] Brading, K., Castellani, E. (eds.): Symmetries in Physics: Philosophical Reflections. Cambridge University Press, Cambridge (2003)

[7] Conway Morris, S.: The predictability of evolution: glimpses into a post-Darwinian world. Naturwissenschaften 96, 1313–1337 (2009)

[8] Conway Morris, S.: Evolution: like any other science it is predictable. Phil. Trans. R. Soc. B 365, 133–145 (2010)

[9] Crick, F.H.C.: Central dogma of molecular biology. Nature 227, 561–563 (1970)

[10] Culberson, J.C.: On the futility of blind search: An algorithmic view of "no free lunch". Evolut. Comput. 6, 109–127 (1998)

[11] Deem, M.W.: Statistical mechanics of modularity and horizontal gene transfer. Annu. Rev. Condens. Matter Phys. 4, 287–311 (2013)

[12] Drossel, B.: Biological evolution and statistical physics. Adv. Phys. 50, 209–295 (2001)

[13] Droste, S., Jansen, T., Wegener, I.: Optimization with randomized search heuristics–the (A)NFL theorem, realistic scenarios, and difficult functions. Theor. Comp. Sci. 287, 131–144 (2002)

[14] Dufresne, F., Jeffery, N.: A guided tour of large genome size in animals: what we know and where we are heading. Chromosome Res. 19, 925–938 (2011)

[15] Eddy, S.: Non-coding RNA genes and the modern RNA world. Nat. Rev. Genet. 2, 919–929 (2001)

[16] Flamm, C., Stadler, B.M.R., Stadler, P.F.: Generalized topologies: Hypergraphs, chemical reactions and biological evolution. In: Basak, S.C., Restrepo, G., Villaveces, J.L. (eds.) Advances in Mathematical Chemistry: With Applications to Chemoinformatics, Bioinformatics, Drug Discovery, and Predictive Toxicology, Bentham Publishers, Sharjah (in press, 2013), `http://www.tbi.univie.ac.at/newpapers/pdfs/TBI-p-2012-9.pdf`

[17] Gilbert, C., Cordaux, R.: Horizontal transfer and evolution of prokaryote transposable elements in eukaryotes. Genome Bio. Evol. 5, 822–832 (2013)

[18] Gillespie, J.H.: Molecular evolution over the mutational landscape. Evolution 38, 1116–1129 (1984)

[19] Gitchoff, P., Wagner, G.P.: Recombination induced hypergraphs: a new approach to mutation-recombination isomorphism. Complexity 2, 37–43 (1996)

[20] Goldenfeld, N., Woese, C.: Biology's next revolution. Nature 445, 369 (2007)

[21] Goldenfeld, N., Woese, C.: Life is physics: Evolution as a collective phenomenon far from equilibrium. Annu. Rev. Condens. Matter Phys. 2, 375–399 (2011)

[22] Gregory, T.R.: The Evolution of the Genome. Elsevier Academic Press, Burlington (2005)

[23] Gregory, T.R.: Animal Genome Size Database, `http://www.genomesize.com` (accessed May 22, 2013)

[24] Hairston Jr., N.G., Ellner, S.P., Geber, M.A., Yoshida, T., Fox, J.A.: Rapid evolution and the convergence of ecological and evolutionary time. Ecol. Lett. 8, 1114–1127 (2005)

[25] Holland, J.H.: Signals and Boundaries: Building Blocks for Complex Adaptive Systems. MIT Press, Cambridge (2012)

[26] Huttegger, S.M., Mitteroecker, P.: Invariance and meaningfulness in phenotype spaces. Evol. Biol. 38, 335–351 (2011)

[27] Johnson, B.R., Lam, S.K.: Self-organization, natural selection, and evolution: Cellular hardware and genetic software. BioScience 60, 879–885 (2010)

[28] Lobkovsky, A.E., Wolf, Y.I., Koonin, E.V.: Predictability of evolutionary trajectories in fitness landscapes. PLoS Comput. Biol. 7(12), e1002302 (2011), doi:10.1371/journal.pcbi.1002302

[29] Kauffman, S.A.: The Origin of Order: Self–Organization and Selection in Evolution. Oxford University Press, New York (1993)

[30] Kauffman, S.A.: Investigations. Oxford University Press, New York (2002)

[31] Keeling, P.J., Palmer, J.F.: Horizontal gene transfer in eukaryotic evolution. Nat. Rev. Genet. 9, 605–618 (2008)

[32] Kirschner, M.W., Gerhart, J.C.: The Plausibility of Life: Resolving Darwin's Dilemma. Yale University Press, New Haven (2005)

[33] Malan, K.M., Engelbrecht, A.P.: A survey of techniques for characterising fitness landscapes and some possible ways forward. Information Science 241, 148–163 (2013)

[34] McCandlish, D.M.: Visualizing fitness landscapes. Evolution 65, 1544–1558 (2011)

[35] Meyers, L.A., Bull, J.J.: Fighting change with change: adaptive variation in an uncertain world. Trends in Ecology & Evolution 17, 551–557 (2002)

[36] Mozhayskiy, V., Tagkopoulos, I.: Horizontal gene transfer dynamics and distribution of fitness effects during microbial in silico evolution. BMC Bioinformatics 13, S13-1–S13-17 (2012)

[37] Noble, D.: Genes and causation. Phil. Trans. R. Soc. A 366, 3001–3015 (2008)

[38] Noble, D.: A biological relativity view of the relationships between genomes and phenotypes. Prog. Biophys. Mol. Biol. 111, 59–65 (2013)

[39] Park, J.M., Deem, M.W.: Phase diagrams of quasispecies theory with recombination and horizontal gene transfer. Theor. Popul. Biol. 70, 479–485 (2006)

[40] Parter, M., Kashtan, N., Alon, U.: Facilitated variation: How evolution learns from past environments to generalize to new environments. PLoS Comput. Biol. 4(11), e1000206 (2008), doi:10.1371/journal.pcbi.1000206

[41] Pelletier, F., Garant, D., Hendry, A.P.: Eco-evolutionary dynamics. Phil. Trans. R. Soc. B 364, 1483–1489 (2009)

[42] Pellicer, J., Fay, M.F., Leitch, I.J.: The largest eukaryotic genome of them all? Bot. Jour. Linnean Soc. 164, 10–15 (2010)

[43] Poelwijk, F.J., Kiviet, D.J., Weinreich, D.M., Tans, S.J.: Empirical fitness landscapes reveal accessible evolutionary paths. Nature 445, 383–386 (2007)

[44] Poli, R., Vanneschi, L., Langdon, W.B., McPhee, N.F.: Theoretical results in genetic programming: the next ten years? Genet. Program Evolvable Mach. 11, 285–320 (2010)

[45] Prügel–Bennett, A.: Symmetry breaking in population-based optimization. IEEE Trans. Evolut. Comp. 8, 63–79 (2004)

[46] Ray, T.S.: Is it alive or is it GA? In: Belew, R.K., Booker, L.B. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 527–534. Morgan Kauffman, San Mateo (1991)

[47] Raz, Y., Tannenbaum, E.: The influence of horizontal gene transfer on the mean fitness of unicellular populations in static environments. Genetics 185, 327–337 (2010)

[48] Rogers, A., Prügel–Bennett, A., Jennings, N.R.: Phase transitions and symmetry breaking in genetic algorithms with crossover. Theor. Comp. Sci. 358, 121–141 (2006)

[49] Saito, N., Ishihara, S., Kaneko, K.: Evolution of genetic redundancy: The relevance of complexity in genotype-phenotype mapping. Submitted to Europhys. Lett. (2013), `http://arxiv.org/pdf/1302.2234v1.pdf`

[50] Sakata, A., Hukushima, K., Kaneko, K.: Replica symmetry breaking in an adiabatic spin-glass model of adaptive evolution. Europhys. Lett. 99, 68004 (2012)

[51] Sayama, H., Kaufman, L., Bar-Yam, Y.: Symmetry breaking and coarsening in spatially distributed evolutionary processes including sexual reproduction and disruptive selection. Phys. Rev. E 62, 7065–7069 (2000)

[52] Schaack, S., Gilbert, C., Feschotte, C.: Promiscuous DNA: horizontal transfer of transposable elements and why it matters for eukaryotic evolution. Trends in Ecology & Evolution 25, 537–546 (2010)

[53] Schoener, T.W.: The newest synthesis: understanding the interplay of evolutionary and ecological dynamics. Science 331, 426–429 (2011)

[54] Schuster, P.: A revival of the landscape paradigm: Large scale data harvesting provides access to fitness landscapes. Complexity 17, 6–10 (2012)

[55] Shakhnovich, E.I.: Protein folding thermodynamics and dynamics: where physics, chemistry, and biology meet. Chem. Rev. 106, 1559–1588 (2006)

[56] Shapiro, J.A.: Revisiting the central dogma in the 21st century. N.Y. Acad. Sci. 1178, 6–28 (2009)

[57] Shapiro, J.A.: Mobile DNA and evolution in the 21st century. Mobile DNA 1, 1–14 (2010)

[58] Shpak, M., Wagner, G.P.: Asymmetry of configuration space induced by unequal crossover: implications for a mathematical theory of evolutionary innovation. Artificial Life 6, 25–43 (2000)

[59] Stadler, P.F., Wagner, G.P.: The algebraic theory of recombination spaces. Evol. Computation 5, 241–275 (1998)

[60] Stadler, P.F., Seitz, R., Wagner, G.P.: Population dependent Fourier decomposition of fitness landscapes over recombination spaces: evolvability of complex characters. Bull. Math. Biol. 62, 399–428 (2000)

[61] Stadler, P.F., Stephens, C.R.: Landscapes and effective fitness. Comm. Theor. Biol. 8, 389–431 (2003)

[62] Stadler, B.M.R., Stadler, P.F., Wagner, G.P., Fontana, W.: The topology of the possible: Formal spaces underlying patterns of evolutionary change. J. Theor. Biology 213, 241–274 (2001)

[63] Syvanen, M.: Cross-species gene transfer: implications for a new theory of evolution. J. Theor. Biology 112, 333–343 (1985)

[64] Syvanen, M.: Evolutionary implications of horizontal gene transfer. Ann. Rev. Gen. 46, 341–358 (2012)

[65] Szendro, I.G., Schenk, M.F., Franke, J., Krug, J., de Visser, J.A.G.M.: Quantitative analyses of empirical fitness landscapes. Jour. Stat. Mech., P01005 (2013), dx.doi.org/10.1088/1742-5468/2013/01/P01005

[66] Tanzer, A., Riester, M., Hertel, J., Bermudez-Santana, C.I., Gorodkin, J., Hofacker, I.L., Stadler, P.F.: Evolutionary genomics of microRNAs and their relatives. In: Caetano Anolles, G. (ed.) Evolutionary Genomics and Systems Biology, pp. 295–327. Wiley–Blackwell, Hoboken (2010)

[67] Thompson, J.N.: Rapid evolution as an ecological process. Trends in Ecology & Evolution 13, 329–332 (1989)

[68] Wagner, A.: Robustness and Evolvability in Living Systems. Princeton University Press, Princeton (2007)

[69] Walker, S.I., Davies, P.C.W.: The algorithmic origins of life. J. Royal Society Interface 10, 20120869 (2013), http://arxiv.org/abs/1207.4803

[70] Wallau, G.L., Ortiz, M.F., Loreto, E.L.S.: Horizontal transposon transfer in eukarya: detection, bias, and perspectives. Genome Bio. Evol. 4, 801–811 (2012)

[71] Weinreich, D.M., Watson, R.A., Chao, L.: Sign epistasis and constraint on evolutionary trajectories. Evolution 59, 1165–1174 (2005)

[72] Weinreich, D.M., Sindi, S., Watson, R.A.: Finding the boundary between evolutionary basins of attraction, and implications for Wright's fitness landscape analogy. Jour. Stat. Mech., P01001 (2013), doi:10.1088/1742-5468

[73] Whitley, D., Watson, J.P.: Complexity theory and the no free lunch theorem. In: Burke, E.K., Kendall, G. (eds.) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, pp. 317–339. Springer, New York (2005)

[74] Wiles, J., Tonkes, B.: Hyperspace geography: Visualizing fitness landscapes beyond 4D. Artificial Life 12, 211–216 (2006)

[75] Wills, P.R.: Informed generation: Physical origin and biological evolution of genetic codescript interpreters. J. Theor. Biology 257, 345–358 (2009)

[76] Wilson, A.J., Nussey, D.H.: What is individual quality? An evolutionary perspective. Trends in Ecology & Evolution 25, 207–214 (2010)

[77] Win, M.N., Liang, J.C., Smolke, C.D.: Frameworks for programming biological function through RNA parts and devices. Chem. Biol. 16, 298–310 (2009)

[78] Wolf, J.B., Wade, M.J.: On the assignment of fitness to parents and offspring: whose fitness is it and when does it matter? Jour. Evolut. Biol. 14, 347–356 (2001)

[79] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evolut. Comp. 1, 67–82 (1997)

[80] Zeldovich, K.B., Shakhnovich, E.I.: Understanding protein evolution: from protein physics to Darwinian selection. Annu. Rev. Phys. Chem. 59, 105–127 (2008)

[81] Zhaxybayeva, O., Ford Doolittle, W.: Lateral gene transfer. Current Biology 21, R242–R246 (2011)

Chapter 20
Fitness Landscapes – 20 Years Later

Edward D. Weinberger

I never told Stu Kauffman this, but, when I set out to climb fitness peaks with him 20 years ago as his post-doc, I was a bit suspicious that we were over-simplifying things – at least as far as biological evolution was concerned. Biological evolution, after all, is a dynamical process, and there are many aspects of the dynamics that are not obvious from knowledge of the landscape alone. While my subsequent career has been in quantitative finance, I kept thinking about these caveats, even as I was working with those infamous sub-prime mortgage backed securities, the ones that caused the financial crisis.

And I wondered if we are making similar mistakes in evolutionary theory. The textbooks define the fitness of an organism as the expected number of reproductively viable offspring in a single generation. This definition has the same veneer of exactitude, the same pretense of quantitative certainty that I often see in finance. In finance, we have a parallel to expected number of offspring, and that is expected return on an investment. However, investing has another dimension – entirely distinct from expected return – and that is risk. There is a big difference between an investment for which a 5% return is guaranteed, and an investment where, half the time, returns are 40%, but a 30% loss is equally likely! Yet both of these investments have the same expected return of 5%. Similarly, equating fitness with expected numbers of offspring equates steady population growth with a population growth/decline that could lead to either huge populations or extinction. In fact, a population that is equally likely to grow by 40% per year and shrink by 30% per year will, with certainty, eventually become extinct, as is evident from the following:

After N generations, suppose we have population growth in G of those generations, and population shrinkage in $N - G$ of those generations. Suppose further that,

Edward D. Weinberger
Weinberger Post-Quantitative, Inc., 370 Central Park West, #110,
New York, NY 10025, U.S.A.
Department of Finance and Risk Engineering, Polytechnic Institute of New York University,
6 Metrotech Center, Brooklyn, NY 12201, U.S.A.
e-mail: edw@wpq-inc.com

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_20, © Springer-Verlag Berlin Heidelberg 2014

if the population grows by the factor g if it grows and s if it shrinks. Then the population will grow to $g^G s^{N-G}$ individuals per individual in the starting population after N generations (Note that this latter quantity is a random variable because G is.).

What are the circumstances under which $g^G s^{N-G}$ is smaller than one? Whatever they are, they are certainly the same as those under which $G \ln \frac{g}{s} + N \ln s < 0$. But, by the Strong Law of Large Numbers, $G = p_G N + o(N)$ with probability 1 as $N \rightarrow \infty$, where p_G is the probability that there is growth for a given generation. It follows that there will be extinction with probability 1 if

$$p_G \ln \frac{g}{s} + \ln s < 0 \tag{20.1}$$

(In the situation given above, $p_G = \frac{1}{2}$, $g = 1.4$, and $s = 0.7$, so $p_G \ln \frac{g}{s} + \ln s \approx -0.0101$, thus proving my claim above.).

Finance also teaches us that no man – or company, or organism – is an island. Companies have suppliers, customers, and competitors. Organisms have prey, predators, and ecological niches. A projection of corporate earnings that ignores the overall state of the economy and a calculation of "expected number of offspring" of a deer that ignores whether nearby hunters have bows and arrows or deer rifles are equally meaningless. Yet the above definition of fitness ignores the fact that coevolution can re-define the dynamics of evolutionary change to the point where fitness in a static environment is no longer relevant. Also refer to the discussions in the Chapters 10, 11 and 12 of this book.

Perhaps the most precise possible definition of fitness can be found in Eigen's quasi-species theory [2, 3, 4]. Given the experimentally observed sequence specific replication of RNA macromolecules in vitro in the presence of appropriate RNA monomers and replication enzymes, fitness is simply the replication rate of a specific sequence. Yet, even in that simple case, the fittest (in the sense of fastest replicating) sequence was not the only one that survived. Instead, sequences that were likely to arise from copying errors of the rapid replicators were also found, even if these sequences were, themselves, not quite the best. In fact, depending on the details of the replication rates of all of the sequences involved, the sequence found in the highest concentration might not have the absolute fastest replication rate, but might, instead, be the likely beneficiary of copying errors from many other sequences with slightly faster replication rates but with fewer close mutants that also have fast replication rates.

In settings closer to what we would generally think of as biological, the concept of fitness is even more problematic. Almost every organism interacts with the rest of the biosphere as a prey species for some other organism or a predator of some other organism or both. Thus, the dynamics of that organism's population growth and decline must also account for the population of its predator(s) and/or prey, necessarily coupling the dynamics of all species involved. Even the simplest form of this coupling, in which the growth/death rate of the first species is proportional to both its current population size and that of the prey/predator, is non-linear, in contrast with the linear growth rate implicit in any reasonable definition of fitness. So, not surprisingly, we find all of the behaviors – stability, limit cycles, and even chaos

– inherent in nonlinear systems in the dynamics of actual biological populations (The observation of the first two kinds of dynamics in fish catches in the Adriatic inspired the familiar Lotka-Volterra model. Less familiar to many readers may be the recent discovery that a time series of Canadian lynx populations – well known to defy many previous modeling attempts – is well fitted by a chaotic model. See, for example, Small and Carmel [6].).

Obviously, to the degree that actual biological populations are governed by chaotic dynamics, fitness, and thus the whole edifice of Darwinian selection is called into question. Yet the viability of the concept of fitness has been demonstrated in a host of theoretical and experimental situations, from optimization algorithms in computer science to the emergence of antibiotic resistant bacteria. So when can we say that fitness the ability of an organism to reproduce regardless of chance fluctuations and populations of other organisms in its world is a meaningful concept, and when must we admit that it is not?

One small step towards an answer to this question is suggested by the fact that expected growth rates seem to do a better job of describing the dynamical properties of the process than expected number of offspring per generation. Nor is this an isolated example. As is well known to students of finance, the above population model, a geometric random walk, converges to geometric Brownian motion in the limit as the number of generations per time unit approaches infinity. Unlike the wide class of processes that converge to it, geometric Brownian motion is characterized completely by specifying only the instantaneous growth rate, μ, which would be obtained if there were no random fluctuations, and the standard deviation, σ, of percentage fluctuations in the actual growth rate. The expected instantaneous growth rate of the process is then given by $\mu - \frac{1}{2}\sigma^2$ (See, for example, [5] for details). This last fact allows us to distinguish situations when chance fluctuations are or are not important, simply by observing whether $\mu \gg \sigma^2$.

Expected growth rates also allow us to determine the relevance of coevolutionary effects, especially given the close, albeit surprising relationship between expected growth rates and information theory (see, for example, a detailed discussion of the relationship between the expected growth rate of an investment portfolio and information theory in Cover and Thomas [1]). As I discuss more fully in [8], evolutionary adaptation is the result of decisions made by nature about what kind of organisms continue to populate the earth, and what kind of organisms become extinct, just as price movements in the stock market are the result of the collective buying and selling decisions of every market participant. Both kinds of decisions are informed by what I call "pragmatic information", the amount of information actually used in making a decision [7, 8, 9]. Also refer to Chapter 3 for an application to biological systems. Viewing evolution through the lens of pragmatic information offers an elegant resolution to the problems noted above. If the rate at which the evolutionary decision is being made depends primarily on factors within the organism, then it is reasonable to talk about fitness. If not, not.

In the last 20 years much progress has be made in understanding fitness landscapes underlying both natural and artificial evolutionary processes. At the same time, further fundamental research questions opened up and may become within reach of landscape methods. The chapters of this book reviewed and discussed both the progress and the potential of these methods. May these ideas be received well and contribute to further understanding of biological and artificial evolution.

References

[1] Cover, T., Thomas, J.: Elements of Information Theory. John Wiley & Sons, New York (1991)

[2] Eigen, M.: Self-organization of matter and the evolution of biological macromolecules. Die Naturwissenschaften 58, 465–532 (1971)

[3] Eigen, M., Schuster, P.: The Hypercycle: A Principle of Natural Self-Organization. Simon and Schuster, New York (1979)

[4] Eigen, M., McCaskill, J., Schuster, P.: Molecular quasi-species. The Journal of Physical Chemistry 92, 6881–6891 (1988)

[5] Hofbauer, J., Sygmund, S.: Dynamical Systems and the Theory of Evolution. Cambridge University Press, Cambridge (1988)

[6] Small, M., Carmeli, C.: Re-examination of evidence for low-dimensional chaos in the Canadian lynx data. In: Ueta, T. (ed.) Proc. 2009 International Symposium on Nonlinear Theory and Its Applications, NOLTA 2009, pp. 651–654 (2009)

[7] Weinberger, E.D.: A theory of pragmatic information and its application to the quasispecies model of biological evolution. BioSystems 66, 105–119 (2013), http://arxiv.org/abs/nlin.AO/0105030 (accessed April 10, 2013)

[8] Weinberger, E.D.: Pragmatic information and Gaian development. Mind and Matter 4, 219–234 (2013), http://arxiv.org/abs/nlin.AO/0606012 (accessed April 10, 2013)

[9] Weinberger, E.D.: Pragmatic information rates, generalizations of the Kelly criterion, and financial market efficiency (2011), http://arxiv.org/abs/0903.2243 (accessed April 10, 2013)

Index