Chapter 13
Evolutionary Optimization and Dynamic Fitness Landscapes
From Reaction–Diffusion Systems to Chaotic CML

Hendrik Richter

Abstract. Evolutionary algorithms are a promising option for solving dynamic optimization problems. These problems have fitness landscapes whose topological features change dynamically with the run–time of the evolutionary algorithm. In this chapter, we study these landscapes by analyzing and quantifying their properties using topological and dynamical landscape measures such as modality, ruggedness, information content, dynamic severity and two types of dynamic complexity measures, Lyapunov exponents and bred vector dimension. Here, our main focus is on dynamic fitness landscapes that exhibit spatio–temporal chaotic behavior. We further discuss evolutionary algorithms and modifications needed to make them fit to perform in dynamic landscapes and present numerical experiments showing the algorithms' performances. These results allow us to link the landscape measures to the behavior of the evolutionary algorithms.

13.1 Introduction

An evolutionary algorithm is a stochastically driven but systematic search method for solving optimization problems. All of its three main operators, selection, recombination and mutation, depend on random elements. In other words, an evolutionary algorithm, just as its biological inspiration and namegiver natural evolution, is a phenomenon of chance, albeit the effect of chance is directed, mainly as a result of the selection process. However, due to the heavy influence of chance in the working of the algorithm, it is a challenge to establish some sound theory of evolutionary computation. A corner–stone in such a theory is the conceptional framework of fitness landscapes. The concept of fitness landscapes was introduced in the context of

Hendrik Richter
HTWK Leipzig, Fakultät Elektrotechnik und Informationstechnik,
Institut Mess–, Steuerungs– und Regelungstechnik, D–04251 Leipzig, Germany
e-mail: richter@fbeit.htwk-leipzig.de

I. Zelinka et al. (Eds.): Evolutionary Algorithms and Chaotic Systems, SCI 267, pp. 409–446.
springerlink.com

theoretical biology by Wright in the early 1930s [65] and later became an important tool in theoretical studies in evolutionary optimization [19, 27, 32, 55, 57]. A fitness landscape combines a search space with a notation of fitness for every point in it, which for instance can be obtained by a genotype–to–fitness mapping or more generally by encoding the set of all possible solutions of an optimization problem and assorting a fitness value each. So, the fitness landscape appears as a potential function on which the individuals of the population may move. This permits to pose the question of how properties of the fitness landscape reflect, explain and allow to predict the behavior of the evolutionary algorithm, and vice versa [27]. It can also be studied how the population dynamics of the search algorithm (that is, the flow of the individuals of the evolutionary algorithm in the landscape) interrelate with topological and dynamical features of the fitness landscape.

A traditional field of application and theoretical study for evolutionary algorithms is to consider static optimization problems. These problems have a fitness landscape that does not change its topological features while the evolutionary algorithm is running. In recent years, we saw an increasing interest in solving dynamic optimization problems [9, 24, 37, 44, 69]. Here, the fitness landscape has topological features that change dynamically with the run–time of the evolutionary algorithm. Hence, such dynamic fitness landscapes can be viewed as spatially extended dynamical systems. So, our main topics here are how dynamic fitness landscapes can be formulated mathematically, how they relate to spatio–temporal dynamical systems considered in nonlinear dynamics, what their properties are and how these properties correspond to the behavior of evolutionary algorithms used to do optimization in these landscapes.

In this chapter, we consider dynamic fitness landscapes and study their topological and dynamical properties. We show in Sec. 13.2 how these landscapes can be constructed from reaction–diffusion systems modelled by partial differential equations (PDE) and also from coupled map lattices (CML). With this we intend to establish relationships between these different kinds of description. In this context, our main emphasis is on dynamic fitness landscapes that exhibit spatio–temporal chaotic behavior. In Sec. 13.3 the study of topological and dynamical properties of fitness landscapes is formalized and we present different types of landscape measures. We consider the topological landscape measures modality, ruggedness and information content and the dynamical landscape measure severity and two types of dynamic complexity measures, Lyapunov exponents and bred vector dimensions. The evolutionary algorithm and modifications needed to make them fit to perform in dynamic landscapes are discussed in Sec. 13.4. Four types of implementation are considered, hyper–mutation, self–adaption, and two types of memory schemes, direct and abstract memory. We present numerical experiments with the evolutionary algorithm implementations and the fitness landscapes in Sec. 13.5. We use these experiments to evaluate the performance of the algorithms and to link these results to the landscape measures studied before. The chapter ends with concluding remarks and a pointer at further problems.

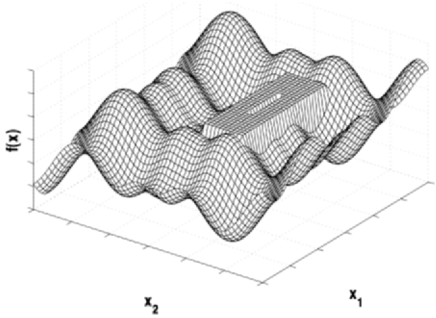

Fig. 13.1 Static fitness landscape in $\mathbb{R}^2$ as mountainous region with peaks, valleys, ridges and plateaus.

13.2 Constructing Dynamic Fitness Landscapes from Reaction–Diffusion Systems and CML

13.2.1 Static and Dynamic Fitness Landscapes

In this chapter we will define dynamic fitness landscapes and provide a framework for posing dynamic optimization problems. According to [27, 57] a static fitness landscape Λ_S is given by

$$\Lambda_S = (S, n, f), \tag{13.1}$$

where S is the search space that can be constructed from a genotype–to–fitness mapping[1] or more generally from encoding the set of all possible solutions of an optimization problem. The neighborhood structure $n(x)$ is a function that assigns to every $x \in S$ a set of neighbors.[2] The fitness function $f(x) : S \to \mathbb{R}$ gives the fitness value for every point in the search space. In Fig. 13.1 a typical fitness landscape is shown over a two–dimensional search space. In this special case with $S = \mathbb{R}^2$ the metaphorical meaning of a fitness landscape as a mountainous region with peaks, valleys, ridges and plateaus becomes particularly apparent. As each point of the search space is characterized by a unique fitness value, solving an optimization problem translates into finding the highest peak (or lowest valley). Hence, a static optimization problem is

[1] In theoretical biology a finer distinction is drawn between genotype and phenotype, e.g. [57], which leads to a genotype–to–phenotype–to–fitness mapping. Genotype here stands for the genetic make–up of a generic individual, i.e. its total genetic information, the sum of all (genetically) possible individuals of a species. The phenotype characterizes a particular individual, i.e. a specific instance of the generic, genotypical individual. In biology this distinction is necessary because genetic fluctuations by mutation can only happen on the level of genotypes, while fitness can only be assigned to phenotypical individuals. In evolutionary computation, genotype can be thought of as standing for the search space, phenotype for the individuals of an evolutionary algorithm, and fitness remains the same.

[2] If the search space is a metric space (for instance a Hilbert (or Banach) space which is frequently taken to define spatially extended systems properly), this neighborhood structure is inherent and there is no need to define it additionally.

$$f_S = \max_{x \in S} f(x), \tag{13.2}$$

which is finding the maximal fitness value f_S and its location $x_S = arg\, f(x_S)$.[3]

If an evolutionary algorithm is employed to solve the problem, the fitness landscape concept becomes once more useful as a population intended to find the optimum can be viewed as though living on the landscape's surface. Moreover, generational change in the population means movement on the surface with the aim to ascent a peak. The picture furthermore illustrates a dynamic optimization problem in that the landscape is to change dynamically beneath the individuals of the population. To describe such a kind of problem, we need the concept of a dynamic fitness landscape[4] which we consider next.

A dynamic fitness landscape Λ_D can be defined by

$$\Lambda_D = (S, n, \Gamma, F, \phi), \tag{13.3}$$

where S is the search space and again represents all possible solutions $x \in S$ of the optimization problem and $n(x)$ is an equivalent neighborhood structure; footnote 2 applies likewise. Γ is a time set (transition semi–group) that defines a measuring and ordering scale for the changes; F is the set of fitness functions and every $f \in F$ with $f : S \times \Gamma \to \mathbb{R}$ depends on time and provides a fitness value to every point in the search space and any element of the time set Γ. The transition map $\phi : F \times S \times \Gamma \to F$ describes how the fitness function changes over time. Further, the map must satisfy the temporal identity and composition conditions, that is $\phi(f, x, 0) = f(x, 0)$ and $\phi(f, x, t_1 + t_2) = \phi(\phi(f, x, t_1), x, t_2)$, $\forall f \in F$, $\forall x \in S$, $\forall t_1, t_2 \in \Gamma$ and the spatial boundary conditions $\phi(f, x_{bound}, t) = f(x_{bound}, t)$, $\forall f \in F$, $\forall t \in \Gamma$ and x_{bound} being the boundary set of search space S. The transition map can depend on continuous and/or discrete values conditional to whether time and/or space possess that property. So, we can put the continuous and discrete number sets, $\mathbb{R}$ and $\mathbb{Z}$, to the time sets and search spaces. For a discrete search space there is $S \subseteq \mathbb{Z}^n$ and for a continuous one $S \subseteq \mathbb{R}^n$, where n is its dimensionality. We use $x \in S \subseteq \mathbb{R}^n$ and $i \in S \subseteq \mathbb{Z}^n$ to specify a point in continuous or discrete search space. The time variables become $\Gamma = \mathbb{Z}$ for discrete and $\Gamma = \mathbb{R}$ for continuous, where we

[3] Optimization problems can be either maximization or minimization problems. As shown with (13.2), we only consider maximization problems here. Between maximization or minimization problems there is the relationship $max\, f(x) = -min\, f(x)$, so this is without loss of generality.

[4] Instead of the term *dynamic fitness landscape* we also find *dynamic environment* or even *non–stationary environment* in the literature. *Environment* and *fitness landscape* are rather synonymous, but we prefer fitness landscape as there is a substantial mathematical theory on fitness landscapes available, which appears to be useful in the context of dynamic optimization. Statistically speaking, the term *non–stationary* implies more than dynamics, namely that the dynamics is generated by a stochastic process and the expected value of the process changes over time. Hence, it should only be used if this is indeed the focus of the dynamics considered.

use $k \in \mathbb{Z}$ and $t \in \mathbb{R}$ to label specific points in time. With these preliminaries, we can formulate the dynamic optimization problem

$$f_S(t) = \max_{x \in S} f(x,t), \qquad \forall t \geq 0, \tag{13.4}$$

which yields the temporarily highest fitness $f_S(t)$ and its solution trajectory[5]

$$x_S(t) = arg\, f_S(t), \qquad \forall t \geq 0. \tag{13.5}$$

For calculating the time evolution of all fitness values in the search landscape, it can be convenient to have an iterative generation law describing how a fitness value at $f(x,t)$ evolves into $f(x,t+\delta t)$ with δt a small time increment. In the dynamic fitness landscape (13.3), this time evolution of a point x not only depends on time and the fitness values of the point itself, but also on the fitness values of surrounding points, that is $f(x+\delta x,t)$ with $\delta x = (\delta x_1, \delta x_2, \ldots, \delta x_n)$. So, a general evolution law becomes

$$f(x,t+\delta t) = \Psi\left(f(x,t), f(x_1+\delta x_1,t), f(x_2+\delta x_2,t), \ldots, f(x_n+\delta x_n,t)\right), \tag{13.6}$$

with Ψ being the generator mapping.

It is noteworthy that such a definition is closely related to the standard definition for dynamical systems, see e.g. [1, 30], Ch. 1. In addition to the elements there, the notation of a time–depended fitness function replaces the state space variables in order to tackle the proposed dynamic optimization problem.

13.2.2 Hierarchy of Fitness Landscapes

For the class of spatially extended systems, a hierarchy of spatio–temporal dynamics has been suggested [15, 26] which stems from the decision of discretization of space and time. We adapt this hierarchy for discussing different kinds of static and dynamic fitness landscapes, see Tab. 13.1.[6] The given classes indicate an increasing degree of complexity which relates to the amount of information required to specify a unique fitness value and hence to one type of scale for the expected difficulty in solving the posed optimization problem. The classes 1 and 2 are static combinatorial and continuous optimization problems which are the topic of a widely ramified and extensive literature in the context of evolutionary computation, e.g. [3, 16, 35]. The

[5] For the dynamic optimization problem in discrete time, we replace formally k for t in (13.4) and (13.5).

[6] In addition to the discretization of space and time, for spatio–temporal dynamics a discretization of the local state variable has been suggested [15, 26], particularly to capture dynamics where states can only have a finite number of different values as for instance described by cellular automata. In our field of application, such a discretization would mean to have discrete fitness values. Such discrete fitness values sometimes occur, for instance in using surrogate models for the fitness function evaluation, but generally, fitness landscapes have rarely this property and so we do not consider such a distinction here.

Table 13.1 Hierarchy of fitness landscapes; S static, D discrete, C continuous

Class	Space	Time	Model
1	D	S	Discrete fitness function
2	C	S	Continuous fitness function
3	D	D	Coupled map lattices (CML)
4	C	D	Continuous fitness function with external discrete dynamics
5	D	C	Lattice of coupled ordinary differential equation (ODE)
6	C	C	Partial differential equation (PDE)

classes 3 and 6 will be closer looked at below. The models in class 3 and 4 include dynamic optimization problems that received much attention in form of continuous or discrete fitness functions whose selected topological features change with a discrete time regime, usually generated by some external source of dynamics.

A well–known example of a class 4 problem is the so–called moving peak benchmark [8, 38] which uses as fitness function $f(x) : S \to \mathbb{R}$ an n–dimensional “field of cones on a zero plane”, where the cones have randomly chosen heights and slopes and are distributed across the landscape. So, we write

$$f(x) = \max\left\{ 0\,,\, \max_{1\leq i\leq N} [h_i - s_i\|x - c_i\|] \right\}, \tag{13.7}$$

where N is the number of cones in the landscape, c_i are the coordinates of the i–th cone, and h_i, s_i specify its height and slope, see Fig. 13.2 for a typical landscape in $\mathbb{R}^2$. The given specification of dynamics requires to move N cones in terms of coordinates, heights and slopes. By defining dynamic sequences for coordinates $c(k)$, heights $h(k)$ and slopes $s(k)$, a dynamic fitness landscape

$$f(x,k) = \max\left\{ 0\,,\, \max_{1\leq i\leq N} [h_i(k) - s_i(k)\|x - c_i(k)\|] \right\} \tag{13.8}$$

can be obtained. In studies of the dynamic fitness landscape (13.8) three main types of dynamics regarding the coordinates $c_i(k)$, heights $h(k)$ and slopes $s(k)$ of the cones have been considered: (i.) regular dynamics usually generated by analytic coordinate transformations, for instance cyclic dynamics where each $c_i(k)$, $h(k)$, $s(k)$ repeats itself after a certain period of time or translatory dynamics where the quantities ascribe a pre–defined track or tour, (ii.) chaotic dynamics generated by a chaotic discrete–time system, for instance the generalized Hénon map, see [44, 45] for details of the generation process, and (iii.) random dynamics with each $c_i(k)$, $h(k)$, $s(k)$ for each k being an independent realization of, for example, a normally or uniformly distributed random variable.

A similar and also popular dynamic fitness landscape is the XOR-generator by Yang [67, 69], which is a class 3 problem. This generator can be constructed from

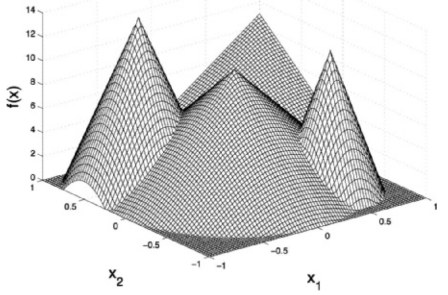

Fig. 13.2 Typical fitness landscape (13.7) for $n = 2$ and $N = 4$.

any binary–encoded stationary function $f(x)$ as follows. For each environment k, an XORing mask $M(k)$ is incrementally generated by

$$M(k) = M(k-1) \oplus T(k), \tag{13.9}$$

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 environment k. $T(k)$ is generated with $\rho \times l$ ($\rho \in (0.0, 1.0]$) random loci set to 1 while the remaining loci are set to 0. For the initial environment $k = 1$, $M(1)$ is set to a zero vector, i.e., $M(1) = 0$.

To summarize, the majority of the literature on evolutionary computation in dynamic fitness landscapes focusses on class 3 and 4 landscapes. In contrast, the class 5 dynamic fitness landscape does not play a major role in studies. It corresponds to a combinatorial optimization problem, where the fitness function changes with continuous time. Even if a practical optimization problem would have such features, we would most likely model discrete time behavior in the dynamic landscape for reasons discussed right afterwards.

A class 3 problem, a CML–based dynamic fitness landscape and its relationship to both a class 4 and class 6 problem, the latter is PDE–based, is the main topic of this chapter. In modelling physical systems, we usually consider continuous changes in both space and time. So, a general dynamic fitness landscape may describe the evolution of fitness values in a search space where the landscape may undergo changes continuously in both space and time. Such a dynamic evolution has to be modelled by a PDE. On the other hand, to facilitate efficient computing, an appropriate discretization is needed, the more so as numerical effort in solving the dynamic optimization problem by an evolutionary algorithm scales with the number of fitness function evaluations. Such a discretization of space and time can be obtained by the CML formalism, in particular for reaction–diffusion systems and surface growth. It is important to note that by doing so essential features of the dynamics are preserved, e.g. [25, 31, 42, 59]. Moreover, as we focus on dynamic fitness functions in which an evolutionary algorithm is used for solving an optimization problem, and as in evolutionary algorithms time is counted by generations and is hence discrete, it appears to be sensible to have dynamic fitness landscapes that change at discrete points in time, too. As mentioned before we put for the CML–based landscape $S \subseteq \mathbb{Z}^n$ and

$\Gamma = \mathbb{Z}$ and for the PDE–based $S \subseteq \mathbb{R}^n$ and $\Gamma = \mathbb{R}$, where n is dimensionality of the search space. Note that this implies $i = (i_1, i_2, \ldots, i_n)$ and $x = (x_1, x_2, \ldots, x_n)$ for the discrete and continuous spatial variables and k and t for the discrete and continuous temporal variables. So, from the generator mapping (13.6) we obtain for discrete time and space the CML–like mapping

$$f(i,k+1) = \Psi\left(f(i,k),\ \sum_{j_1=1}^{J_1} f(j_1,i_2,\ldots,i_n,k),\ \sum_{j_2=1}^{J_2} f(i_1,j_2,\ldots,i_n,k),\ldots\right) \quad (13.10)$$

and for continuous time and space the PDE[7]

$$\frac{\partial f(x,t)}{\partial t} = \Psi\left(f(x,t), \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}, \frac{\partial f^2}{\partial x_1 x_2}, \ldots, \frac{\partial f^2}{\partial x_1 x_n}, \ldots, \frac{\partial f^2}{\partial x_1^2}, \ldots\right). \quad (13.11)$$

With these mathematical descriptions, dynamic fitness landscapes of class 3 and 6 are specified in a very general way. In order to analyze both types and their relationship, we next consider the 2D cases, that is $n = 2$.

13.2.3 Relationships between Coupled Map Lattices and Reaction–Diffusion Systems

For a two–dimensional 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[8] 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

$$\frac{\partial f}{\partial t} = a_1\left(\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2}\right) - a_2 g_1\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) + g_2(f), \quad (13.12)$$

where a_1, a_2 are coefficients and g_1, g_2 are mappings. With (13.12), we describe a class 6 dynamic fitness landscape. It can be interpreted as a reaction–diffusion system with an additional nonlinear term and is a special case of the general description (13.11). This type of PDE has close resemblance to the Kardar–Parisi–Zhang (KPZ) equation [29], 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 [31, 33, 36] while particularly the relation to Coupled Map Lattices (CML) has been a central question. Clearly, both are models of extended dynamical systems. Also and as mentioned before, a numerical solution of a PDE always requires to have some kind of

[7] In doing so, we assume that the space S and the mapping ϕ enjoy properties that guarantee existence and uniqueness of such a spatio–temporal evolution.

[8] To emphasize that the dynamics of the fitness landscape is that of an extended dynamical system, also the (synonymous) term *spatio–temporal fitness landscape* is used.

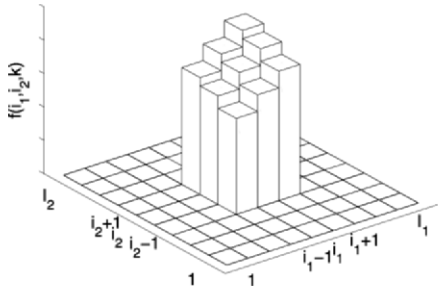

Fig. 13.3 The coupled map lattice (CML) (13.14) as building block for class 3 and 4 fitness landscapes.

discretization of space and time. So, an alternative strategy to a study by any of the methods for numerically solving the PDE, appears to consist of a study of the corresponding CML and their mutual dynamical properties.

Recently, a fitness landscape based on a CML has been studied [46, 47], which is of class 3. We will relate this fitness landscape to the PDE–based landscape (13.12). 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

$$f(i_1,i_2,k), \quad i_1 = 1,2,\ldots,I_1, \quad i_2 = 1,2,\ldots,I_2, \tag{13.13}$$

where (i_1,i_2) denote the spatial indices in vertical and horizontal directions, respectively, see Fig. 13.3. We interpret this height $f(i_1,i_2,k)$ 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 [12, 25]

$$\begin{aligned} f(i_1,i_2,k+1) = (1-\varepsilon)g(f(i_1,i_2,k)) + \frac{\varepsilon}{4}\Big[\, & g(f(i_1-1,i_2,k)) + g(f(i_1+1,i_2,k)) \\ & + g(f(i_1,i_2-1,k)) + g(f(i_1,i_2+1,k))\Big], \end{aligned} \tag{13.14}$$

where $g(f(i_1,i_2,k))$ is a local mapping function and ε is the diffusion coupling strength. As local mapping function we use the logistic map

$$g(f(i_1,i_2,k)) = \alpha f(i_1,i_2,k)(1-f(i_1,i_2,k)). \tag{13.15}$$

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 (13.15) exhibits chaotic behavior, for instance in the parameter interval $\alpha \in [3.57,4]$. This 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

$$\begin{aligned} f(I_1+1,i_2,k) &= f(1,i_2,k), \\ f(i_1,I_2+1,k) &= f(i_1,1,k). \end{aligned} \tag{13.16}$$

Initialization of the CML is done by initial heights $f(i_1,i_2,0)$ 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 CML (13.14) can be seen as a special case in two dimensions of the general CML–like spatio–temporal mapping (13.10).

The CML are known to exhibit a rich spatio–temporal behavior, including different types of spatio–temporal periodicity and chaos, quasi–periodicity and pattern formation. So, the CML are 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.

We now link the discrete space and time fitness landscape $f(i_1,i_2,k)$ to the landscape with continuous space and time $f(x_1,x_2,t)$ according to eq. (13.12). We take the continuum limit of the CML and employ the following discretizations: a forward difference of the time derivative

$$\frac{\partial f(x_1,x_2,t)}{\partial t} \leftrightarrow \frac{f(i_1,i_2,k+1)-f(i_1,i_2,k)}{\delta t} \tag{13.17}$$

and the central differences of the space derivatives

$$\frac{\partial f(x_1,x_2,t)}{\partial x_1} \leftrightarrow \frac{1}{2}\frac{f(i_1+1,i_2,k)-f(i_1-1,i_2,k)}{\delta x_1}, \tag{13.18}$$

$$\frac{\partial f(x_1,x_2,t)}{\partial x_2} \leftrightarrow \frac{1}{2}\frac{f(i_1,i_2+1,k)-f(i_1,i_2-1,k)}{\delta x_2} \tag{13.19}$$

and the second derivatives

$$\frac{\partial^2 f(x_1,x_2,t)}{\partial x_1^2} \leftrightarrow \frac{f(i_1+1,i_2,k)-2f(i_1,i_2,k)+f(i_1-1,i_2,k)}{(\delta x_1)^2}, \tag{13.20}$$

$$\frac{\partial^2 f(x_1,x_2,t)}{\partial x_2^2} \leftrightarrow \frac{f(i_1,i_2+1,k)-2f(i_1,i_2,k)+f(i_1,i_2-1,k)}{(\delta x_2)^2}, \tag{13.21}$$

with the time step δt and the spatial steps $\delta x_1, \delta x_2$ being equal to one in the used system of units. So, we obtain the PDE

$$\frac{\partial f}{\partial t} = \frac{\alpha\varepsilon}{4}\left(\frac{\partial^2 f}{\partial x_1^2}+\frac{\partial^2 f}{\partial x_2^2}\right) - \frac{\alpha\varepsilon}{2}\left(\left(\frac{\partial f}{\partial x_1}\right)^2+\left(\frac{\partial f}{\partial x_2}\right)^2\right)+(\alpha-1)f-\alpha f^2, \tag{13.22}$$

where $k\delta t \rightarrow t$, $i_1\delta x_1 \rightarrow x_1$, $i_2\delta x_2 \rightarrow x_2$ and $f(i_1,i_2,k) \rightarrow f(x_1,x_2,t)$. With eq. (13.22), we have a parabolic PDE of the reaction–diffusion type (13.12).

As shown in [47] from the CML (13.14), a fitness landscape of class 4 with continuous space and the search space variable x can been defined by setting scaling factors $s_1, s_2 \in \mathbb{R}_+$ and by imposing a rounding condition, so that

$$\begin{pmatrix} \lceil s_1 x_1 \rceil \\ \lceil s_2 x_2 \rceil \end{pmatrix} = \begin{pmatrix} i \\ j \end{pmatrix}. \tag{13.23}$$

So, we obtain the dynamic fitness function for the two–dimensional CML (13.14) as

$$f(x,k) = \left\{ \begin{array}{cc} f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k) \text{ for } & \begin{array}{c} 1 \leq \lceil s_1 x_1 \rceil \leq I_1 \\ 1 \leq \lceil s_2 x_2 \rceil \leq I_2 \end{array} \\ 0 & \text{otherwise} \end{array} \right\}, k \geq 0. \tag{13.24}$$

This dynamic fitness landscape, see Fig. 13.4, will be the test bed for the numerical experiments reported in Sec. 13.5. For (13.24), we can pose a dynamic optimization problem

$$f_S(k) = \max_{x \in \mathbb{R}^2} f(x,k) = \left\{ \max_{\substack{1 \leq \lceil s_1 x_1 \rceil \leq I_1 \\ 1 \leq \lceil s_2 x_2 \rceil \leq I_2}} f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k) \right\}, k \geq 0, \tag{13.25}$$

which yields a sequence $f_S(k)$ of the highest fitness. Solving the dynamic optimization problem defines a solution trajectory

$$x_S(k) = \arg\max_{x \in \mathbb{R}^2} f(x,k) = \arg \left\{ \max_{\substack{1 \leq \lceil s_1 x_1 \rceil \leq I_1 \\ 1 \leq \lceil s_2 x_2 \rceil \leq I_2}} f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k) \right\}, k \geq 0, \tag{13.26}$$

which we intend to find by using an evolutionary algorithm.

To summarize, we have shown the mathematical relationship between a CML–based dynamic fitness landscape and a class of parabolic PDEs. In doing so, we have created a link between a computational efficient and hence numerically experimentable model and a description of real physical phenomena such as surface growth and reaction–diffusion dynamics. So, the dynamic fitness landscape (13.24) can be considered to be a more realistic description of real–world dynamic optimization problems than the benchmarks such as (13.8) or (13.9).

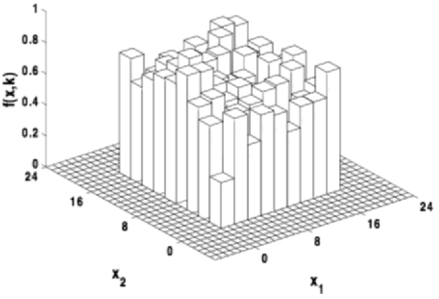

Fig. 13.4 CML–based fitness landscape (13.24) for $I_1 = I_2 = 9$ and $s_1 = s_2 = 0.5$.

13.3 Properties of Dynamic Fitness Landscapes

A main concern in evolutionary computation is developing, testing and applying algorithms that solve optimization problems. From a theoretical perspective, it is therefore desirable to classify the tackled optimization problems in order to compare the approaches and also to suggest possibilities for improving the algorithms. A first and rather structural classification is provided by the hierarchy of fitness landscapes given in Sec. 13.2.2. An important aspect of the concept of fitness landscapes is that it provides a theoretical framework for describing, evaluating, comparing and quantifying the difficulty of a given optimization problem. Based on this notation, it is intended to obtain an estimation for the behavior an evolutionary algorithm is expected to have. Within the conceptional framework this question is addressed by concepts and quantifiers for measuring fitness landscapes [21, 27, 57, 62]. These landscape measures can also be seen as an attempt to define some types of metric for fitness landscapes. Some of these measures have been suggested for static fitness landscapes and hence account for *topological properties* of the fitness landscape. Besides, for dynamic fitness landscape also *dynamical properties* have to be considered.

13.3.1 Topological Properties and Topological Problem Difficulty

Unfortunately, even for a static fitness landscape the question of how difficult a certain optimization problem is for an evolutionary algorithm is not easy to answer. If we view a fitness landscape as in the Figs. 13.1 or 13.2, we see a collection of hills and valleys that can be accompanied by ridges, plateaus, etc. The optimization task is constituted by finding the highest hill (or lowest valley). The evolutionary algorithm puts individuals into these landscapes, while in the generational circle they should trawl the search space and finally find the optimum. In this process, the only feedback from the landscape comes from the fitness values of all member of the population. Moreover, a movement towards an optimum can only be expected if either an individual with high fitness pulls other individuals to itself in the recombination step or random fluctuations working on the individuals during the mutation step put them nearer to the optimum. However, as both movements are censored by the selection step, if either one or the other movement leads to a decreasing fitness, it becomes futile. From these thoughts it is clear why a single sphere of ever increasing fitness with a single highest value, see Fig. 13.5, is a particularly easy problem to solve. There is no distraction for the evolutionary process. The individuals just move up the single hill. However, in more complex landscapes with more than one hill also their number, size, form and distribution constitute difficulty in the search process of stochastically driven search procedures as evolutionary algorithms. In terms of an optimization task these features correspond to the number of optima, how they are distributed and what the space in–between the optima looks like.

It is easily understood and intuitively reasonable that the difficulty of finding the global optimum among several local optima depends on their number; the larger the

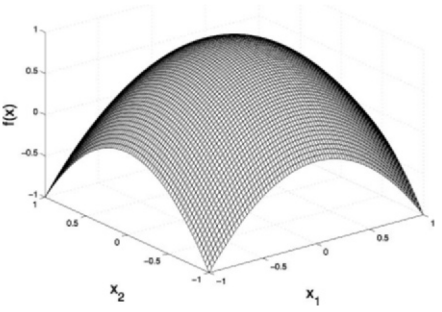

Fig. 13.5 Sphere–like fitness landscape of continuously increasing fitness towards the single optimum.

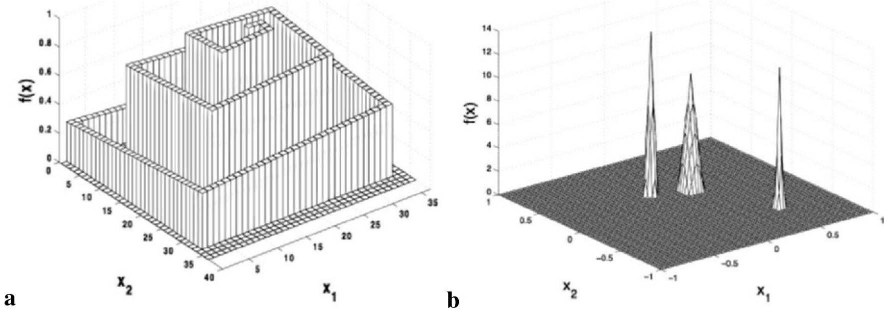

Fig. 13.6 Different types of fitness landscapes: a) long–path problem, b) neutrality with spike–like peaks.

number of local optima, the more difficult the problem is [27, 57]. Second, distribution of the optima relates to problem difficulty. The problem becomes different if the optima are either grouped in one subset of the search space or if they are scattered widely. In the former case, the population only has to find the area of the optima and can then jump from lower optima to higher optima using the stochastic drive in the algorithm. If the optima are distributed widely, the population might split and some areas might not be searched at all. Furthermore, as pointed out above an evolutionary algorithm is using differences in the fitness of individuals populating the fitness landscape in its search process. That means not only the wider surrounding of the optima counts, but also the nearer neighborhood is of interest. For instance, the problem belongs to a different category, if the optima consist of peaks that gradually slope down into all directions, see Fig. 13.5, or the peak can only be approached by a narrow single path of monotonically increasing fitness (so–called long path problems [23]), see Fig. 13.6a, or there are slim and distant peaks on an otherwise plain surface of equal or nearly equal fitness (so–called neutrality [55]), see Fig. 13.6b. These geometrically motivated features are addressed by the notion of accessability (or basin of attraction) of optima and they are a third major factor of landscape topology that contributes to problem hardness.

However, from these main ingredients of problem difficulty, it cannot be easily deduced how in a certain landscape the number of local optima, their distribution and their accessability balance each other in terms of problem difficulty [55, 62]. Clearly, modality, which expresses the number of local optima, is a primary factor and might in addition have the advantage that it can be assigned rather straightforwardly by enumeration. However, apart from the sheer number of optima, it is the interplay of all three aspects that defines hardness of optimizing in a fitness landscape for an evolutionary algorithm. This is the reason why landscape measures have been suggested that are aimed to catch problem hardness more generally rather than just by accounting for a single aspect of the landscape topology. In addition, these measures allow to weight the three factors.

For evaluating topological problem difficulty the topological landscape measures

- modality = number and distribution of local maxima,
- ruggedness = analysis of the static correlation structure,
- information content = an entropic landscape measure,
- epistasis = a Walsh analysis

have been suggested and for static landscapes, these measures have been studied intensively [19, 57, 62]. In [22, 46, 47], these measures were applied to dynamic fitness landscapes. It has been shown that dynamic fitness landscapes inherit topological aspects of problem difficulty from their static counterparts. On the other hand, the features of the dynamics in the landscape contribute in their own way. The definitions of the topological measures and the results obtained for the CML–based dynamic fitness landscape (13.24) are briefly recalled in Sec. 13.3.3 to have a reference and comparison to the dynamical measures considered next. Only the Walsh epistasis measure is omitted as it had been shown in [47] that it poorly reflects problem hardness for the CML–based dynamic landscape considered here.

13.3.2 Dynamical Properties and Dynamical Problem Difficulty

In a dynamic fitness landscape not only topological features constitute problem difficulty, there is also a contribution of features of the involved dynamics [9, 24, 37]. Similar to the situation with topological properties, there is no simple classification. If we look at a dynamic fitness landscape (imagine a landscape as in the Figs. 13.1 or 13.2, where now the hills, valleys and plateaus are changing their position and shape and move around the plane that forms the search space) then again some intuitively comprehensible factors that make finding the moving optima easy or hard can be seen. An evolutionary algorithm carries out a parallelized population–based search in which detecting the optimum depends on improvements over a certain number of generations; with the more generations available, the better for problem solving. Therefore, finding the optimum in just one generation is highly improbable and generally speaking controverts the fundamental idea of evolutionary search. Dynamical problems that can be solved robustly by an evolutionary algorithm should involve a change pattern that allows the algorithm at least a certain number of generations.

So, the speed at which the landscape changes must have some influence. Generally it must hold that the faster the speed (more changes per time interval) is, the more complicated is the dynamical problem. In defining the speed of the landscape changes, the time scale of the dynamic fitness landscape needs to be related to the computation time of the evolutionary algorithm, which counts time by generations. In general, a generation of the evolutionary algorithm results from the computation time for the fitness evaluation, which needs to be carried out for all the individuals of the population and usually is the main contribution, and the time needed for executing the evolutionary operators such as selection, recombination and mutation, which is a minor part. This gives an estimate for the time required to calculate one generation. Note that for a given implementation and hardware, this time can be converted into real CPU time. So, for a population size that is constant over the evolutionary run, for every generation the (approximately) same time interval should go by. For a dynamic fitness landscape that also has a continual change pattern, this means that both time scales relate linearly.

We can describe the dynamics of an evolutionary algorithm by the generation transition function ψ, see e.g. [3], p. 64–65, which can be interpreted as a nonlinear probabilistic dynamical system that maps and transforms a population $P(\tau)$ at generation $\tau \in \mathbb{N}_0$ into a population $P(\tau+1)$ at generation $\tau+1$,

$$P(\tau+1) = \psi(P(\tau)), \tau \geq 0 \tag{13.27}$$

by using the evolutionary operators selection, recombination and selection (and possibly some additional operators such as memory, hyper–mutation and so on). Starting from an initial population $P(0)$, eq. (13.27) describes the population dynamics in the search space. With the proposed linear scale between both the time scales of the evolutionary algorithm τ and the time scale of the dynamic fitness landscape k, we obtain a relation by the change frequency $\gamma \in \mathbb{N}$.[9] There is

$$\tau = \gamma k \tag{13.28}$$

with γ being constant.[10] The quantity γ can be interpreted as the computation time that the algorithm needs to solve the problem and hence is an estimate of the required

[9] Instead of the term *change frequency*, we can also find *change period* in the literature. Change period is motivated by interpreting γ as time interval, change frequency because γ indicates after how many generations the landscape changes. In the following, we prefer the latter interpretation.

[10] The relation (13.28) links the time scales for dynamic fitness landscapes of class 3 and 4 with discrete time as for instance given by (13.8) or (13.10). For dynamic fitness landscapes of class 6 as modelled by (13.11) and (13.12) the changes happen continuously. This means changes in the dynamic fitness landscape occur in–between generations or several (in theory an infinite number of) times within one generation. But as fitness evaluation in an evolutionary algorithm usually takes place just once in a generation, these changes would probably not come into effect before the next generation, that is the next synchronization point between $t \in \mathbb{R}$ and $\gamma^{-1}\tau$. Therefore, the discussion above applies to continuous dynamic fitness landscapes in the same way.

time between changes of the fitness landscape.[11] However, numerical experiments in evolutionary computation of dynamic fitness landscapes usually view the change frequency γ slightly differently. They consider γ an adjustable parameter that can be used to evaluate and compare different types and implementations of evolutionary algorithms. This view is justified by the fact that γ can indeed be adjusted by modifications done on the parameters and implementation of the evolutionary algorithm and the hardware on which the algorithm runs. Note that this view also means that the change frequency is no longer a property of the dynamic fitness landscape but is seen to be independent of it. In the numerical experiments, we will adopt this view and consider change frequency as something that can be adjusted. Note further that by doing so, change frequency has a unique role among all other topological and dynamical properties defining problem hardness. While all the other properties belong to the dynamic fitness landscape considered, change frequency is defined by the evolutionary algorithm and hence assumed to be freely settable (at least within certain limits).

Aside from the (relative) speed of the landscape changes, a second major dynamical influence on problem hardness addresses the spatial distance that the optimum moves if the landscape changes, that is the (relative) strength of the landscape changes. As optimum finding for an evolutionary algorithm implies to trawl the search space for a certain time, time restrictions as those coming from a changing landscape mean that the average distance between subsequent optima is a good measure for problem hardness. This dynamical property is called dynamic severity for which there are several notations [9, 45, 63]. They all have in common that they measure the (relative) magnitude of the changes by comparing the landscape at subsequent points in time, for instance between k to $k+1$ or t to $t+\delta t$. In terms of the dynamic fitness landscapes of class 3 and 4, dynamic severity means to evaluate the (average) distance from the highest peak's coordinates $x_S(k) = arg f_S(k)$ before and after a change, as given by (13.5).[12] With eq. (13.44), this is applied to the CML–based dynamic fitness landscape considered here.

In dynamic optimization, we sometimes find a discrimination between gradual and abrupt changes. What distinguishes gradual from abrupt changes is basically understood as different degrees of dynamic severity, but somehow change frequency also contributes and is intertwined with it. Our view is this. For landscapes with discrete time the situation is rather straightforward. As in discrete time the changes happen one after the other to distinct points in time, an abrupt change is one with a large severity, a gradual one has a low severity, no matter what the change frequency is. For discrete time dynamic fitness landscape of class 3 and 4, the landscape has

[11] Usually, γ is considered to be constant for all generations τ, but it might also be a function of k and even be different (for instance a positive integer realization of a random process) for every k.

[12] Using a similar argumentation as for change frequency, see footnote 10, dynamic severity can be defined for class 6 dynamic fitness landscapes as for instance (13.11) in the likewise fashion. Therefore, the highest peak's coordinates as given from the solution trajectory of the dynamic optimization problem (13.4) must be compared for a time lapse between t and $t+\delta t$.

no speed by itself, the only base for comparison is the generational dynamics of the evolutionary algorithm linearly linked via the change frequency (13.28). For changes in the fitness landscape in continuous time, a more elaborated discussion is necessary; for those dynamic fitness landscapes we need a notation of speed on its own. The (direction–less) speed $v_S(t)$ of the optimum $x_S(t)$ at time t can be defined by

$$v_S(t) = \lim_{\delta t \to 0} \quad \frac{\|x(t+\delta t) - x(t)\|}{\delta t}. \tag{13.29}$$

For δt being small and constant, the average speed of the optimum $\langle v_S(t) \rangle$ can be calculated by

$$\langle v_S(t) \rangle = \frac{1}{K} \sum_{k=0}^{K-1} \frac{\|x(t+(k+1)\delta t) - x(t+k\delta t)\|}{\delta t} \tag{13.30}$$

with K sufficiently large. The nominator term is dynamic severity (cf. eq. (13.44) so that for continuous time dynamic fitness landscape an abrupt change is indicated by a $\langle v_S(t) \rangle \delta t$ above a certain limit, while a gradual change is characterized by a small value of this quantity.

Dynamic severity is an intrinsic property of the fitness landscape. This also applies to a third dynamical property of fitness landscapes that tries to capture the complexity of the dynamics. In this context, complexity refers to limits in the long–term predictability of the spatio–temporal evolution, even if explicit stochastic elements in the describing equations are absent. So, studying dynamic complexity is highly linked to and conceptionally as well as methodically overlapping with the study of deterministic chaos.[13] A first and again rather structural classification of dynamic complexity is to categorize dynamics as either *regular*, *chaotic* or *random*. Here, regular dynamics is completely predictable and usually generated by analytic coordinate transformations. It might, for instance, be cyclic, where we have a periodic recurrence of all topological features of the landscape after a certain time interval or translatory, where the topological features follow a pre–defined track or tour. Chaotic dynamics is generated by deterministic chaotic systems (that might be locally interacting with the spatially distributed landscape) and is predictable only for a short term. Random dynamics stems from a stochastic process, that is from realizations of a random variable, and is unpredictable even for short terms.

Apart from this verbal assignment and in order to have a quantification, we can resort to quantifiers of dynamics used and established in the field of nonlinear dynamics, such as Lyapunov exponents and vectors, different types of entropies and

[13] An alternative (and complimentary to the degree of predictability) approach to define complexity is by using concepts from algorithmic information theory, e.g. [5, 14]. Accordingly, algorithmic complexity of a spatio–temporal evolution is defined by the length of the smallest algorithm capable of specifying the evolution. As chaotic evolutions are nonperiodic and oscillatory, their algorithmic complexity is large. However, algorithmic complexity only superficially allows to separate chaos and random, as all chaotic behavior is algorithmically complex, but not all evolutions that are algorithmically complex, are chaotic, too.

information flows, correlations and related quantities as bred vector dimensions, or fractal dimensions. All these quantities are widely used as an analyzing tool in nonlinear dynamical systems theory. However, they were developed to deal with low–dimensional nonlinear (possibly chaotic) dynamical systems described by ordinary differential equations (ODEs). Only for such systems, these quantities are unambiguously meaningful and the relationships between the quantities are largely understood.[14] In recent years, several attempts have been made to extend the theory of low–dimensional dynamical systems to (infinite–dimensional) spatio–temporal systems and by doing so to establish quantities similar to the conventional Lyapunov exponents, dimensions and entropies. Still, this work is in its infancy and the power of the theory is confined to certain limits. Also, while some quantities as Lyapunov exponents or bred vector dimensions have shown to be meaningful in quantifying patterns of space–time dynamics, others are rather ambiguous and allusive. Moreover, the relationships between the quantities are still far from being clear.

In the following, we consider the dynamic landscape measures

- change frequency = speed of fitness landscape changes relative to EA,
- dynamic severity = distance between subsequent optima,
- dynamic complexity = predicability of spatio–temporal evolution:
  - Lyapunov exponents = divergence rate between nearby evolutions,
  - bred vector dimensions = analysis of the dynamic correlation structure.

In the next section, we will study dynamical landscape measures for the CML–based fitness landscape. As discussed above, change frequency is here considered as a parameter to be set. On the other hand, dynamic severity and the quantities to measure dynamic complexity, that are Lyapunov exponents and bred vector dimensions, will be looked at as depending on the dynamic landscape.

13.3.3 Topological and Dynamical Landscape Measures for the CML–Based Landscape

13.3.3.1 Topological Measures

Topological landscape measures have been intensively studied for the CML–based dynamic fitness landscape (13.24) in [46, 47] and we therefore only briefly recall the definitions and some of the results.

Modality. For the fitness landscape (13.24), the topological landscape measure modality accounts for the average number of local maxima and can be assigned by enumeration. We consider as neighborhood structure the surrounding heights. That means the neighborhood structure $N_{\pm 1}(i_1,i_2)$ of the (i_1,i_2)–cell is

$$N_{\pm 1}(i_1,i_2) = (i_1+\delta_1, i_2+\delta_2), \tag{13.31}$$

[14] For instance, there are the relationships between Lyapunov exponents and fractal dimensions via the Kaplan–Yorke dimension [13, 18, 49] or between Lyapunov exponents and entropies via the Kolmogovov–Sinai entropy and the Pesin entropy formula, see e.g. [4].

where (δ_1, δ_2) are taken as disjunction of the permutations over the set $\mathbb{S} = \{-1, 0, 1\}$ that is

$$\begin{aligned}(\delta_1, \delta_2) = (-1,-1) \wedge (-1,0) \wedge (-1,1) \wedge (0,-1) \\ \wedge (0,1) \wedge (1,-1) \wedge (1,0) \wedge (1,1).\end{aligned} \qquad (13.32)$$

Additionally, the cell specified by $(\delta_1, \delta_2) = (0,0)$ is excluded from the neighborhood structure $N_{\pm 1}(i_1, i_2)$. Here, $(i_1, i_2)^T = (\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil)^T$. Hence, the fitness function possesses a local maximum at point (i_1, i_2) and time k if

$$f(i_1, i_2, k) \geq f(N(i_1, i_2), k). \qquad (13.33)$$

We denote $\#_{LM}(k)$ the number of local maxima at time k. As this quantity may change over time, we consider its time average $\langle \#_{LM}(k) \rangle$, which is

$$\langle \#_{LM}(k) \rangle = \lim_{K \to \infty} \quad \frac{1}{K} \sum_{k=0}^{K-1} \#_{LM}(k). \qquad (13.34)$$

For computing an approximate value of the time average number of local maxima, the $\langle \#_{LM}(k) \rangle$ is replaced by $\#_{LM} = \frac{1}{K} \sum\limits_{k=0}^{K-1} \#_{LM}(k)$ with K sufficiently large.

Ruggedness. The topological landscape measure ruggedness can be analyzed by the static correlation structure. This method works by performing a random walk on the landscape and calculating its random walk correlation function. For the dynamic fitness landscape (13.24), this begins with generating a time series

$$f(\tau_s, k) = f(i_1(\tau_s), i_2(\tau_s), k), \quad \tau_s = 1, 2, \ldots, T \qquad (13.35)$$

of the heights $f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k)$ with $(i_1, i_2)^T = (\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil)^T$. For doing the random walk, we create $2 \times T$ independent realizations (t_{i_1}, t_{i_2}) of an integer random variable uniformly distributed on the set $\mathbb{S} = \{-1, 0, 1\}$. Starting from an initial cell $(i_1(1), i_2(1))^T$, the next cell indices $(i_1(\tau_s + 1), i_2(\tau_s + 1))^T$ on the walk are obtained by adding the two independent realizations of the random variable to the current cell indices:

$$(i_1(\tau_s + 1), i_2(\tau_s + 1))^T = (i_1(\tau_s) + t_{i_1}, i_2(\tau_s) + t_{i_2})^T. \qquad (13.36)$$

In addition, the boundary condition (13.16) is observed. From the random walk in the two spatial dimensions that is specified by $(i_1(\tau_s), i_2(\tau_s))^T$, we obtain the needed time series on the dynamic fitness landscape by recording the heights $f(\tau_s, k) = f(i_1(\tau_s), i_2(\tau_s), k)$ at time k. For this time series, the spatial correlation can be calculated. The spatial correlation is widely used in determining ruggedness of static landscape [19, 56, 64]. It is an estimate $r(t_L, k)$ of the autocorrelation function of the time series with time lag t_L, also called random walk correlation function:

$$r(t_L,k) = \frac{\sum\limits_{\tau=1}^{T-t_L} \left(f(\tau_s,k)-\bar{f}(k)\right)\left(f(\tau_s+t_L,k)-\bar{f}(k)\right)}{\sum\limits_{\tau_s=1}^{T} \left(f(\tau_s,k)-\bar{f}(k)\right)^2}, \tag{13.37}$$

where $\bar{f}(k) = \frac{1}{T}\sum\limits_{\tau_s=1}^{T} f(\tau_s,k)$ and $T \gg t_L > 0$. The spatial random walk correlation function measures the correlation between different regions of the fitness landscape for a fixed k. As $r(t_L,k)$ changes over time, we consider its time average $\langle r(t_L,k)\rangle$, for which we calculate numerically an approximated value $r(t_L)$ similarly as for the average number of maxima. From this quantity, the correlation of the lag t_L

$$\lambda_R(t_L) = -\frac{1}{\ln\left(|r(t_L)|\right)} \tag{13.38}$$

can be obtained. Among the correlations of the lag t_L, it has been shown that ruggedness is best expressed by the correlation length [56]

$$\lambda_R = -\frac{1}{\ln\left(|r(1)|\right)}, \tag{13.39}$$

which is the correlation of the lag $t_L = 1$. The lower the value of λ_R, the more rugged is the landscape. This kind of evaluating fitness landscapes relies upon the assumption that the landscape is statistically isotropic [19, 20]. This means that the value of $r(t_L,k)$ obtained from the random walk does not depend on the specific random walk used and particularly not on the chosen initial cell. Our numerical results have shown that this holds for the landscapes considered here.

Information content. The topological landscape measure information content can be accounted for by entropic measures [58, 61, 62]. Starting point for this method to evaluate landscapes is again, as for the correlation structure considered above, a time series (13.35), $f(\tau_s,k)$, which is generated by a random walk on the dynamic landscape for a fixed time k. From this time series, we code the difference in fitness between two consecutive walking steps by the symbols $s_{\tau_s} \in \mathbb{S}$, $\tau_s = 1,2,\ldots,T-1$, taken from the set $\mathbb{S} = \{-1,0,1\}$. These symbols are calculated by

$$s_\tau(e,k) = \begin{cases} -1, & \text{if} \quad f(\tau_s+1,k)-f(\tau_s,k) \ < e \\ 0, & \text{if} \quad |f(\tau_s+1,k)-f(\tau_s,k)| \leq e \\ 1, & \text{if} \quad f(\tau_s+1,k)-h(\tau_s,k) \ > e \end{cases} \tag{13.40}$$

for a fixed $e \in [0,L]$, where L is the maximum difference between two fitness values. The obtained symbols are concatenated to a string

$$S(e,k) = s_1 s_2 \ldots s_{T-1}. \tag{13.41}$$

The parameter e defines the sensitivity by which the string $S(e,k)$ accounts for differences in the fitness values. For $e = 0$, the string $S(e,k)$ contains the symbol zero only if the random walk has reached a strictly flat area. It hence discriminates very sensitively between increasing and decreasing fitness values. On the other hand, for $e = L$, the string only contains the symbol zero, which makes evaluating the structure of the landscape pointless. In this way, a fixed value of e with $0 < e < L$ defines a level of detail of information about the landscape's structure. The string (13.41) represents this information depending on e and codes it by subblocks over the set $\mathbb{S}$. In other words, varying e allows to zoom in on or to zoom out of the information structure of the landscape.

For defining entropic measures of the landscape, we look at the distribution of subblocks of length two, $s_{\tau_s}s_{\tau_s+1}$, $\tau_s = 1,2,\ldots T-2$, within the string (13.41). These subblocks stand for local patterns in the landscape. We denote the probability of the occurrence of the pattern $\delta_1\delta_2$ with $\delta_1,\delta_2 \in \mathbb{S}$ and $\delta_1 \neq \delta_2$ by $p_{\delta_1\delta_2}$. For numerical calculation, we approximate this probability by the relative frequency of the patterns within the string $S(e,k)$. As the set $\mathbb{S}$ consists of three elements, we find six different kinds of subblock $s_{\tau_s}s_{\tau_s+1} = \delta_1\delta_2$ with $\delta_1 \neq \delta_2$ within the string. From their probabilities at a fixed time k and a given sensitivity level e we calculate the entropic measure [62]

$$H_{IC}(e,k) = -\sum_{\substack{\delta_1,\delta_2\in\mathbb{S}\\ \delta_1\neq\delta_2}} p_{\delta_1\delta_2}(e,k)\log_6\left(p_{\delta_1\delta_2}(e,k)\right), \qquad (13.42)$$

which is called information content of the fitness landscape. Note that by taking the logarithm in Eq. (13.42) with the base 6, the information content is scaled to the interval $[0,1]$. As for the other landscape measures, for evaluating dynamic fitness landscapes, we consider the time average $\langle H_{IC}(e,k)\rangle$ for which we numerically calculate an approximated value $H_{IC}(e)$. In the numerical calculation, we set the value $e = 0$, that is we consider the information content of highest sensitivity. As was shown in [47], epistasis measured by Walsh analysis is not a particularly meaningful quantity for the CML–based fitness landscape with chaotic behavior. Therefore, we do not consider it here.

Fig. 13.7 shows the topological landscape measures modality $\#_{LM}$, ruggedness λ_R and information content H_{IC} for varying ε, $\alpha = 3.999$, $s_1 = s_2 = 1$ and constant lattice sizes. We see that in large areas of the parameter space of ε there are similar characteristics, which in the case of λ_R scales in an inverse manner. Further, there are rarely intervals in ε where the measures remain constant. Also, only for the modality measure $\#_{LM}$ we find real differences for varying lattice sizes. The other two measures are very similar no matter if the quadratic lattice size is $I_1 = I_2 = 12$ or $I_1 = I_2 = 20$. Moreover, for $0.1 \leq \varepsilon \leq 0.3$, all topological measures behave differently from the other parameter values; we find a rather erratic characteristic. These results can be understood by considering the spatio–temporal behavior of the CML that defines the fitness landscape. In this parameter range, the CML are known to possess spatio–temporal periodic patterns, while elsewhere the system exhibits spatio–temporal chaos [66].

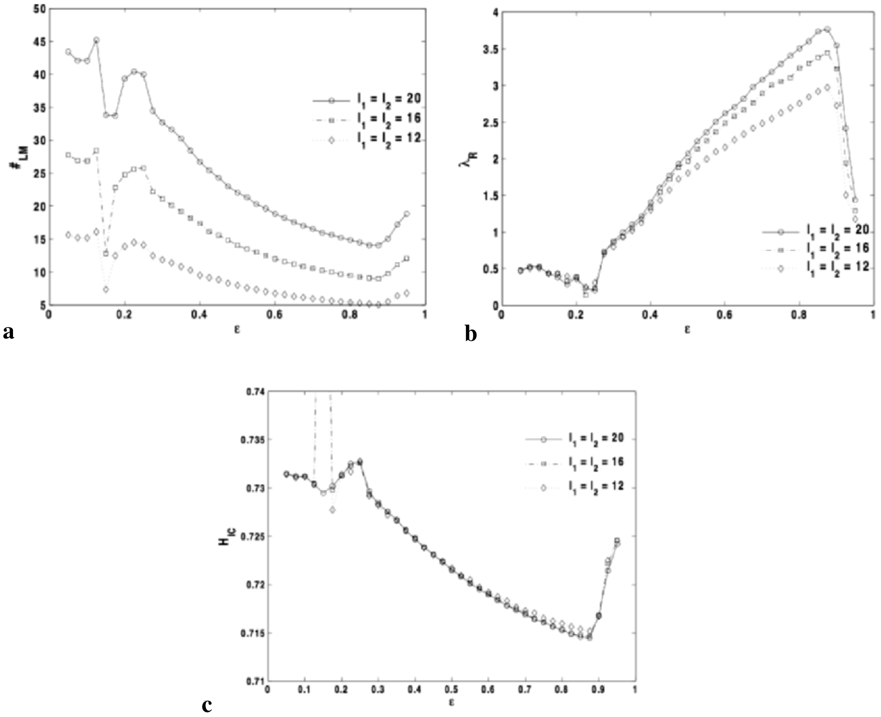

Fig. 13.7 The topological landscape measures for varying ε, $\alpha = 3.999$ and constant lattice sizes: a) modality $\#_{LM}$, b) ruggedness λ_R and c) information content H_{IC}.

13.3.3.2 Dynamical Measures

We next consider dynamic landscape measures and start with dynamic severity as change frequency is treated as a property of the evolutionary algorithms, but not of the dynamic fitness landscape.

Dynamic severity. As discussed above, dynamic severity accounts for the average distance from the highest peak's coordinates $x_S(k) = arg f_S(k)$ before and after a change.

Hence, dynamic severity σ can be calculated for the CML–based fitness landscape (13.24) by

$$\sigma(k+1) = \|x_S(k+1) - x_S(k)\| \tag{13.43}$$

with

$$x_S(k) = \arg\left\{ \max_{\substack{1 \leq \lceil s_1 x_1 \rceil \leq I_1 \\ 1 \leq \lceil s_2 x_2 \rceil \leq I_2}} f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k) \right\}$$

being the solution of the dynamic optimization problem (13.26). As this quantity may vary with time k, we consider the time average severity

$$\langle \sigma(k) \rangle = \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} \sigma(k) \tag{13.44}$$

and calculate an approximative value σ similarly as done for the other measures.

Dynamic complexity measure: Lyapunov exponents. A first method to measure dynamic complexity is by using the concept of Lyapunov exponents. The Lyapunov exponents give the divergence or convergence rates between a time evolution of the system and its nearby evolution that results from being displaced from the original one by an infinitesimal perturbation [28, 40]. For calculating the (largest) Lyapunov exponent for the CML–based fitness landscape (13.24), we therefore consider the time evolution itself, $f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k)$ and its neighboring evolution $\Delta f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k)$, which is obtained by linearizing the system along the evolution $f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k)$. The linearized system determined along $f(\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil, k)$ describes exactly the result of an infinitesimal displacement and allows to observe if both time evolutions diverge (positive Lyapunov exponent) or converge (negative Lyapunov exponent). A positive Lyapunov exponent generally indicates chaos, and the magnitude of the positive Lyapunov exponent can be seen as a dynamic complexity measure.

We write the linearization of the CML–based fitness landscape (13.24) as follows:

$$\begin{aligned}\Delta \tilde{f}(i_1, i_2, k) &= \frac{d}{df} g(f(i_1, i_2, k))\\ \Delta f(i_1, i_2, k+1) &= (1-\varepsilon)\tilde{f}(i_1, i_2, k) + \frac{\varepsilon}{4}\Big[\, \tilde{f}(i_1 - 1, i_2, k) + \tilde{f}(i_1 + 1, i_2, k)\\ &\qquad + \tilde{f}(i_1, i_2 - 1, k) + \tilde{f}(i_1, i_2 + 1, k) \Big],\end{aligned} \tag{13.45}$$

where $(i_1, i_2)^T = (\lceil s_1 x_1 \rceil, \lceil s_2 x_2 \rceil)^T$. This linearization is the tangential system to the 2D CML and has dimension $I_1 \times I_2$. For its calculation, the boundary conditions (13.16) have to be taken into account. From the linearization (13.45), we can define the (largest) Lyapunov exponent λ_L as

$$\lambda_L = \lim_{k \to \infty} \frac{1}{k} \ln \frac{\|\Delta f(i_1, i_2, k)\|}{\|\Delta f(i_1, i_2, 0)\|}, \quad \forall \quad 1 \leq i_1 \leq I_1, 1 \leq i_2 \leq I_2. \tag{13.46}$$

Calculation of the Lyapunov exponent can be done using standard methods, for instance using QR–factorization, which is known to be computationally efficient, reliable and robust, see e.g. [11].

Dynamic complexity measure: bred vector dimension. As a second measure for dynamic complexity, we consider bred vector dimensions. Bred vector dimensions are a concept for evaluating the dynamic correlation structure of spatially extended systems [17, 41, 43]. It has been used to identify local regions where this correlation is high and which can therefore be used for prediction and short term forecast. Here, we will use bred vector dimensions as a general measure for the dynamic correlation structure of fitness landscapes and so as another dynamic landscape measure.

For calculating the bred vector dimension, we again consider a neighborhood structure as in (13.31), but with a difference: $N_{\pm\ell}(i_1,i_2)$ is now the disjunction of the permutations over the set $\mathbb{S} = \{-\ell,-\ell+1,\ldots,-1,0,1,\ldots,\ell-1,\ell\}$ and $(\delta_1,\delta_2) = (0,0)$ remains included. So, we specify $(2\ell+1)^2$ cells around and including the cell (i_1,i_2) by this neighborhood structure. For doing so around every (i_1,i_2), again the boundary conditions (13.16) need to be satisfied. As there are maximally $I_1 \times I_2$ cells with (possibly) different fitness values in the CML–based fitness landscape and because taking into account the same spatial subsection twice would bias the results, we need to limit $\ell \leq \frac{1}{2}(\sqrt{I_1 I_2} - 1)$. We now calculate the dynamic fitness landscape for a point in time k, which should be large enough so that the transients starting from $f(i_1,i_2,0)$ are removed. At this point in time, we disturb κ times the fitness landscape $f(i_1,i_2,k)$ within the neighborhood structure $N_{\pm\ell}(i_1,i_2)$ by a small Gaussian noise with mean zero and standard deviation std, that is

$$f_j(N_{\pm\ell}(i_1,i_2),k) := f_j(N_{\pm\ell}(i_1,i_2),k) + \mathcal{N}(0,std), \quad j = 1,2,\ldots,\kappa. \tag{13.47}$$

Then, we define a time lag k_L and calculate the time evolution of all κ disturbed landscapes for further k_L iterations. The obtained $f_j(N_{\pm\ell}(i_1,i_2),k+k_L)$ are formed into a vector $\hat{f}_j$ of length $(2\ell+1)^2$, which is called the bred vector. It is normalized to unity and hence contains the normalized fitness values for the entire neighborhood structure of the disturbed landscape. From the κ bred vectors $\hat{f}_j$ as columns, we build a matrix B, which is of dimension $(2\ell+1)^2 \times \kappa$. The matrix $B^T B$ can be regarded as its corresponding covariance matrix. It expresses the local linear independence of the κ local bred vectors. A measure for this independence can be obtained from the singular values σ_i of the matrix B, which are the roots of the eigenvalues of the covariance matrix $B^T B$, $\sigma_i = \sqrt{eig_i(B^T B)}$. These singular values are a measure for the amount of variance in the set of bred vectors. In other words, they account for the degree of difference imposed by the disturbances applied to the fitness landscape k_L time steps before. From these singular values we finally calculate the quantity

$$\psi_B = \frac{\left(\sum_{i=1}^{\kappa} \sigma_i\right)^2}{\sum_{i=1}^{\kappa} \sigma_i^2}, \tag{13.48}$$

which is called the bred vector dimension [41]. As each bred vector, which forms one of the columns of the matrix B, is normalized to unit length, ψ_B can have values $1 \leq \psi_B \leq \kappa$. The value $\psi_B = 1$ indicates that all bred vectors are equal, meaning that the correlation between them is maximal. Any $1 < \psi_B \leq \kappa$ expresses differences in the bred vectors with the magnitude of ψ_B being a measure of the amount of difference. An integer ψ_B can even be interpreted as to relate to the dimensionality of the subspace spanned by the bred vectors. So, by fixing a time lag k_L, we obtain a quantity for the degree of temporal divergence and correlation that disturbances in the fitness landscape cause and hence a dynamic correlation structure related to

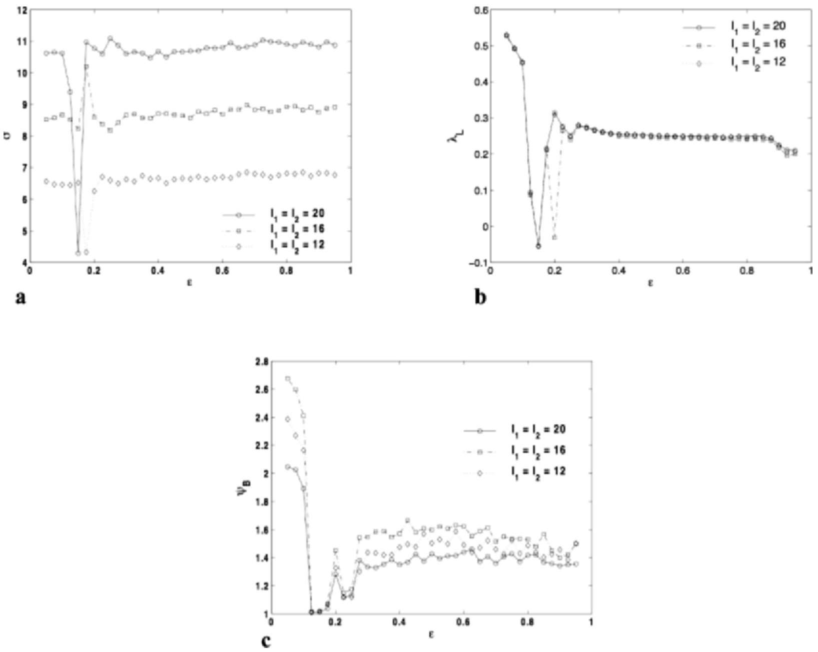

Fig. 13.8 The dynamical landscape measures for varying ε, $\alpha = 3.999$ and constant lattice sizes: a) dynamic severity σ, b) Lyapunov exponent λ_L and c) bred vector dimension ψ_B.

dynamic complexity. Note also that the bred vector dimension is a measure that is conceptionally similar to the Lyapunov exponents and Lyapunov vectors considered above. The main difference is that Lyapunov vectors account for the effect of infinitesimal perturbations to the time evolution, while bred vectors evaluate the result of finite perturbations. In the numerical calculation, the values $k_L = 25$, $\kappa = 5$ and $std = 0.0001$ have been taken. As reference cell in the center of the neighborhood structure, the cell with the maximum fitness value at time k was used.

Fig. 13.8 shows the dynamical landscape measures dynamic severity σ and the dynamic complexity measures Lyapunov exponent λ_L and bred vector dimension ψ_B for varying ε, $\alpha = 3.999$, $s_1 = s_2 = 1$ and constant lattice sizes. The most striking feature is that all three dynamical landscape measures show large parameter intervals in ε, particularly for $\varepsilon > 0.3$ where the values vary only slightly. This is a characteristic not found in the topological measures, see Fig. 13.7. Also, only dynamic severity shows a strong dependency on the lattice size. Again, as for the topological measures, the parameter interval with spatio–temporal periodic patterns is clearly visible. For the Lyapunov exponents, we obtain negative values, indicating an absence of chaos.

The dynamical measure severity not only depends on the lattice size but also on the scaling factors s_1 and s_2. As shown in Fig. 13.8a, the quantity σ obtained for the quadratic lattice and $(i_1, i_2)^T = (\lceil x_1 \rceil, \lceil x_2 \rceil)^T$ (that is $s_1 = s_2 = 1$) can be regarded as constant for a given lattice size and a large majority of values of ε. So, dynamic severity of the CML–based fitness landscape with $s_1 = s_2 = 1$ is a dynamic property depending on the choice of α, ε and $I_1 \times I_2$. This property can be linearly scaled and adjusted by s_1 and s_2. In general, for quadratic lattices with $I_1 = I_2$, dynamic severity scales as $\sigma \sim \sqrt{(s_1^2 + s_2^2)} \cdot I_1$, while for rectangular lattices, we have $\sigma \sim \sqrt{s_1^2 I_1^2 + s_2^2 I_2^2}$.

13.4 Evolutionary Optimization

As we have discussed in the previous section, problem hardness in a dynamic optimization problem not only depends on topological features of the associated fitness landscape, but also on its dynamical properties. In this context, it is interesting to note that one of the driving forces behind the development of evolutionary algorithms was exactly to have search methods for topological difficult problems. So, it has been shown that evolutionary algorithms are remarkably successful in solving static optimization problems with a high degree of problem difficulty. In recent years it has further been shown that these problem solving abilities can also be used to tackle dynamic optimization problems. However, certain modifications in the algorithmic structure of the evolutionary algorithm are necessary to make it work in dynamic fitness landscapes. The working principle of evolutionary algorithms is to maintain and evolve a population of candidate solutions through selection, recombination and mutation. The next generation's population is generated by first selecting relatively fitter individuals from the current population and then applying changes to the selected individuals. This is sequentially done by either more deterministic means (recombination) and or more stochastic (mutation). With these steps the new off–spring of the next generation are created. In the normal working mode the individuals in the population will eventually converge to the optimal solution due to the selection pressure. This convergence property, when happening at a proper pace, is intended and expected from the evolutionary algorithm in order to locate the optimal solution of the static problems.

For solving dynamic optimization problems, however, this convergence property causes big problems for the evolutionary algorithm because it deprives the population of genetic diversity. Dynamic optimization means not longer to find one optimal solution, but to track the movement of the optimal solution with time. Consequently, when the fitness landscape changes, it is hard for the population to escape from the old optimal solution in order to search for the new one, if its diversity is low. The algorithm's convergence is simultaneously corroding its genetic diversity that after a change is exactly needed to explore the search space and to find the optimum again. This situation means that the algorithm must be equipped with some additional schemes which can control, maintain and occasionally enhance the population's diversity.[15] For achieving this goal, several approaches have been suggested.

[15] This often requires to detect the point in time where a change in the landscape occurs, see [48] for a discussion of the involved problems.

One way to classify them is by looking at what element of the evolutionary algorithm is modified and if therefore rather stochastic or deterministic means are used. With this classification, there are four groups of schemes for diversity management to make evolutionary algorithms fit to perform in dynamic fitness landscapes:

- on the level of the algorithm's individuals by mainly stochastic means, as for instance hyper–mutation [39] or random immigrants [60],
- on the level of the algorithm's population by mainly deterministic means, as for instance with different types of memory [8, 50, 53, 68] and multi–population approaches [10],
- on the level of the algorithm's parameters, as for instance by stochastic self–adaption of the mutation [2, 7],
- on the level of the algorithm's operators with additional and completely different operators, as for instance for anticipating and predicting the dynamics [6, 51, 54].

In a specific implementation, there can be hybrid types of the above mentioned schemes. One of those is an abstract memory scheme [50, 51], which combines a memory with a prediction of dynamics. Apart from the abstract memory scheme, we will consider a hyper–mutation scheme, a standard direct memory and a self–adaptive mutation scheme in the numerical experiments with the CML–based dynamic fitness landscape.

The considered schemes work as follows, see Tab. 13.2 for details of the parameter settings. With the hyper–mutation scheme, we have a standard evolutionary algorithm with selection, recombination and mutation. In the implementation here, we use a population size μ, a tournament selection of tournament size 2, a fitness–related intermediate sexual recombination (which is operated μ times and for each recombination two individuals are chosen randomly to produce offspring that is the fitness–weighted arithmetic mean of both parents), and a standard mutation with mutation probability p_m and base mutation rate bm (that means a mutated individual x' differs from an un–mutated individual x by $(x'-x) \sim bm \cdot \mathcal{N}(0,1)$). For the initial population, individuals are generated whose elements are realizations of a random variable normally distributed on $[0, \omega^2]$. The hyper–mutation now increases the mutation strength if a change in the fitness landscape has occurred for a limited number of generations (usually one or two). Therefore, the base mutation rate is multiplied by the hyper–mutation rate hm, so that the hyper–mutated individuals are $(x'-x) \sim bm \cdot hm \cdot \mathcal{N}(0,1)$. In this way, for a certain generational lag, the need for an abrupt increase in genetic diversity is satisfied.

In self–adaption, we use the standard operators of an evolutionary algorithm and have used the same implementation as for hyper–mutation, that is population size μ, tournament selection and fitness–related intermediate sexual recombination. In contrast, the mutation rate itself is a parameter that undergoes a permanent optimization and adaption process. Therefore, the mutation rate is considered as an additional subject to optimize. For every individual in the population, its mutation rate becomes an extra component. In other words, the mutation rate becomes an additional dimension in the fitness landscape. So, the mutation rates mr are subject to the selection process and are continuously changed by an own mutation (and hence adaption) process, see e.g. [2, 3, 34]. They are adapted every generation with $mr(\tau+1) = mr(\tau) \cdot \exp(\tau_A \mathcal{N}(0,1))$, where τ_A is an adaption rate and for generation $\tau = 0$, $mr(0) = mr_0$. Based on this self–adapting rate, the individuals are mutated according to the magnitude that the quantity has in a particular generation, that is $(x' - x) \sim mr(\tau+1) \cdot \mathcal{N}(0,1)$. With such a design of the mutation process, the mutation rate is no longer a parameter to be set before the evolutionary run and controlled by the experimenter. So, the mutation rates might converge to some (optimal) values or might be oscillating between certain values. As the rates are subject to the internal optimization and adaption process, they often have well–fitting values. On the other hand, this may also mean that the rates take values that are poorly suited. To counter this effect, in the mutation step, which follows the adaption of the mutation rates, the number λ of individuals generated by mutation, called offspring candidates, exceeds the population size of the parents. In a second selection process only the best offspring candidates are picked, eliminating ill–fitting mutation rates in the process.

The principle of memory schemes is to store useful information from the old fitness landscape and reuse it later in a new one. Therefore, a memory with the same representation as the population is set up that splits an extra storage space to explicitly store information from a current generation. This information is employed later, for instance by merging the stored individuals with the population at that time. This is known as direct memory [8, 53, 68]. The memory has the size μ_{mem} and is feeded by individuals selected for their high fitness. If a change in the landscape occurs, the stored individuals are inserted in the population, replacing individuals with low fitness. Since memory space is usually limited, we need to update the information stored in the memory. A general strategy is to select one memory space to be replaced by the best individual of the population.

A second example for a memory that additionally realizes some elements of predicting the dynamics of the landscape is the abstract memory scheme [50, 51]. The basic idea of abstract memory is that the solutions are not stored directly, that is as individuals representing points in search space, but as their abstraction. We understand as an abstraction of a good solution its approximate location in the search space, which is therefore partitioned with a grid of size ε_G. In an abstract storage process, μ_{stor} individuals with high fitness are taken and sorted according to the partition in the search space which they represent. Each individual sorted increases a counter belonging to that partition and indicates how often a good solution has occurred in exactly this subsection of the search space. In the abstract retrieval process, we fix a number of individuals to be inserted in the population by μ_{retr} and create these individuals randomly such that their statistical distribution regarding the partition matches that stored in the memory. Hence, abstract memory combines ideas from memory such as saving individuals for future re–insertion with attempts to predict the time evolution of the dynamic fitness landscape: storing the abstraction of good solutions, that is to use their approximate location in the search space, allows to deduce a probabilistic model for the spatial distribution of future solutions of the problem.

Table 13.2 Parameter of the tested evolutionary algorithms

Considered Scheme	Design parameter	Symbol	Value
All schemes	Population size	μ	50
	Initial population width	ω^2	5
Hyper–mutation	Base–mutation rate	bm	0.1
	Hyper–mutation rate	hm	30
	Mutation probability	p_m	0.25
Self–adaption	Offspring size	λ	50
	Initial mutation rates	mr_0	1.5
	Adaption rate	τ_A	1.25
Direct memory	Memory size	μ_{mem}	10
Abstract memory	Grid size	ε_G	1.0
	Individuals to memory	μ_{stor}	3
	Individuals from memory	μ_{retr}	20

The parameters used in the implementations that are the subjects of the numerical experiments reported in the next section, are summarized in Tab. 13.2.

13.5 Numerical Experiments

The performance of the algorithms is measured by the Mean Fitness Error (MFE), defined as below:

$$MFE = \frac{1}{R}\sum_{r=1}^{R}\left[\frac{1}{T}\sum_{\tau=1}^{T}\left(f(x_s(k),k) - \max_{x_j(\tau)\in P(\tau)} f(x_j(\tau),k)\right)\right]_{k=\lfloor\gamma^{-1}\tau\rfloor}, \qquad (13.49)$$

where $\max_{x_j(\tau)\in P(\tau)} f\left(x_j(\tau),\lfloor\gamma^{-1}\tau\rfloor\right)$ is the fitness value of the best–in–generation individual $x_j(\tau)\in P(\tau)$ at generation τ, $f\left(x_S(\lfloor\gamma^{-1}\tau\rfloor),\lfloor\gamma^{-1}\tau\rfloor\right)$ is the maximum fitness value at generation τ, T is the number of generations used in the run, and R is the number of consecutive runs. We set $R = 150$ and $T = 1500$ in all experiments. Note that $f\left(x_s(\lfloor\gamma^{-1}\tau\rfloor),\lfloor\gamma^{-1}\tau\rfloor\right)$ and $\max_{x_j(\tau)\in P(\tau)} f\left(x_j(\tau),\lfloor\gamma^{-1}\tau\rfloor\right)$ change every γ generations according to Eq. (13.28). The MFE serves as a performance criterion and as behavior data for the evolutionary optimization in the dynamic fitness landscape (13.24). Fig. 13.9 show the MFE for the four considered evolutionary algorithm implementations hyper–mutation, self–adaption, direct and abstract memory and the landscape parameters $\alpha = 3.999$, $I_1 = I_2 = 16$, $s_1 = s_2 = 1$ and different values of ε. For these values the dynamic fitness landscape shows mostly spatio–temporal chaotic behavior, but also spatio–temporal periodic patterns. The figures give the MFE and its 95% confidence interval for the change frequencies $\gamma = 10$, $\gamma = 20$ and $\gamma = 30$. We observe that for most values of ε, the curves for each γ are distinct with the smallest γ leading to the largest MFE, and vice versa.

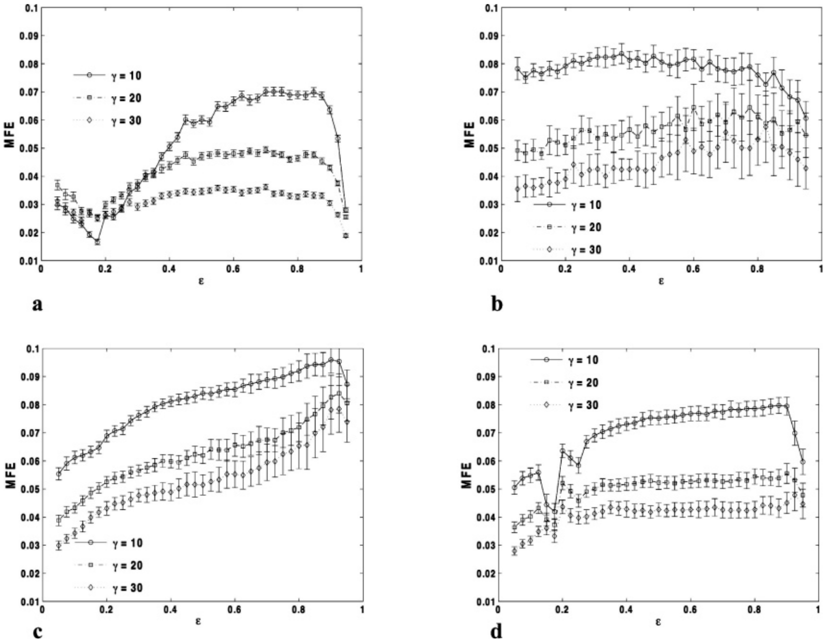

Fig. 13.9 Behavior of different implementations of the evolutionary algorithm expressed as the mean fitness error MFE (13.49) for the CML–based fitness landscape (13.24) with $I_1 = I_2 = 16$ and $\alpha = 3.999$: a) hyper-mutation, b) self–adaption, c) direct memory, d) abstract memory.

Looking the results, we see some differences between the four implementations. Although all results are for 150 runs, the confidence intervals that reflect to what degree the given mean value can be expected to be indeed the mean value if an infinite number of runs would have been carried out, are much higher for self–adaption and direct memory than for hyper–mutation and abstract memory. This means that for both implementations, for self–adaption even to a larger degree than for direct memory, some performance results are much better than the mean, but others are much worse. This result may possibly be explained for self–adaption by the fact that the mutation rates evolve towards optimal values after a change, but sometimes exactly these optimal values become unfavorable after the next change. As there is only the feedback via the individuals' fitness and the selection process (including potentially ill–posed mutation rates), it might take a certain time until optimal mutation rates are obtained again. This point of view seems to be confirmed by the observation that the confidence intervals are particularly large for small γ, that is for a landscape that changes fast. For direct memory, the effect might be that the stored individuals inserted after a change do not help in that particular new environment. The retrieved individuals might be favorable after another change, but not after the given one. Also, it is interesting to note that some implementations, particularly hyper–mutation, but to some extent also abstract memory, react on the presence of spatio–temporal periodic patterns at $0.1 \leq \varepsilon \leq 0.3$ with a drop in the MFE.

However, our interest here is neither to discuss possible performance enhancing alterations in the algorithms' parameters or implementation details or more generally the optimal design of the algorithms, nor to argue about which implementation is superior over another. To do so, the extent of the presented results is much to small, and also we find it more illuminating to study the underlying working principles of the algorithms. In other words, in theoretical studies the behavior of the evolutionary algorithm is sometimes much more interesting than the actual performance record. Therefore, we next approach the question of which implementation fulfils the expectations with respect to the landscape measures consider above. To get a metric of the strength of the relationship between landscape measures and the algorithms' behavior, we apply a parametric and a nonparametric correlation analysis, e.g. [52]. In particular, we study the Pearson product–moment correlation and the Kendall rank–order correlation. So, we run tests for relationships between these quantities, these are linear relation (Pearson) and piecewise linear relation (Kendall). With this, we intend to establish how reliable a linear or a piecewise linear relation between the landscape measures and the algorithms' behavior is. We would like to stress that this correlation analysis cannot imply any simple causation between landscape properties and the algorithm's behavior. As frequently in correlation analysis, to claim causation from observed correlation can be questionable or even misleading. Clearly, the landscape measures reflect different aspects of the landscape's problem difficulty, and this problem difficulty, in turn, must affect the flow of individuals in the landscape and hence the behavior of the evolutionary algorithm. However, each topological and dynamical measure emphasizes a specific aspect in problem hardness and there is no guarantee that the considered measures do not ignore properties that are important for details of the algorithms' behavior in exactly that landscape.

In the Figs. 13.10 and 13.11, the Pearson correlation coefficient ρ_P^2 (also known as "Pearson's r") and the Kendall correlation coefficient ρ_K^2 (also known as "Kendall's τ") are given. We write $\rho_P^2(MFE, \#_{LM})$ for the squared Pearson correlation between MFE and $\#_{LM}$, $\rho_K^2(MFE, \#_{LM})$ for the squared Kendall correlation and so on. The square of the correlation coefficients ρ_P^2 is also known as the coefficient of determination and can be interpreted as follows. The squared correlation coefficient represents the fraction of variance that is expressed by the fit between the MFE and the landscape measures. If the data from the landscape measures and the behavior data from the evolutionary algorithm were used in a statistical model, the quantity ρ_P^2 would be a metric of how well this model is able to predict further data. Hence, we view ρ_P^2 as a measure of reliability, strength and predictive power of the relationship between the landscape measures and the evolutionary algorithms' behavior. For the correlation coefficient ρ_K^2 a likewise interpretation is possible.

The results in the Figs. 13.10 and 13.11 show some clear trends. A first is that the topological landscape measures modality, ruggedness and information content have a stronger correlation with the behavior of the evolutionary algorithm expressed by

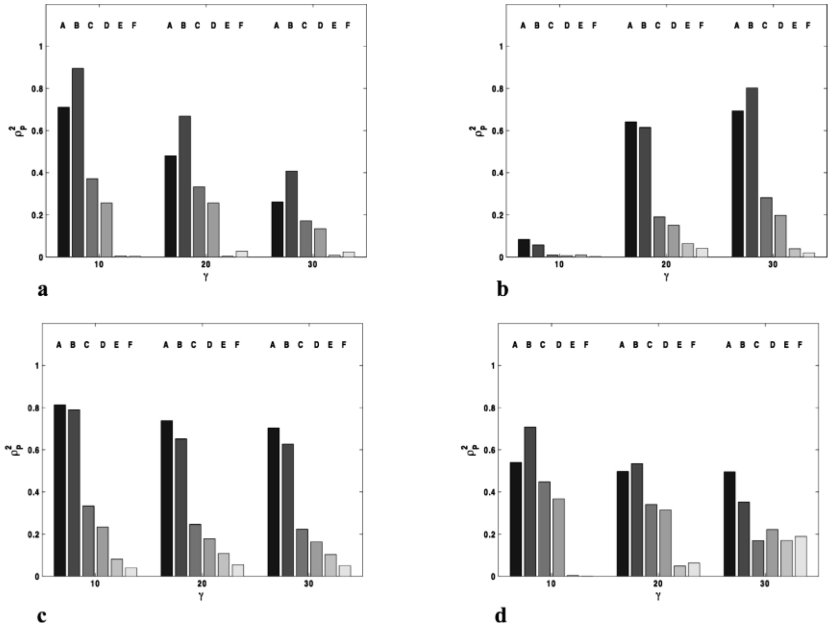

Fig. 13.10 Correlation analysis using Pearson product–moment correlation between topological and dynamical landscape measures and the behavior of the evolutionary algorithm expressed by the mean fitness error MFE for different implementations: a) hyper-mutation, b) self–adaption, c) direct memory, d) abstract memory. A - $\rho_P^2(MFE, \#_{LM})$, B - $\rho_P^2(MFE, \lambda_R)$, C - $\rho_P^2(MFE, H_{IC})$, D - $\rho_P^2(MFE, \sigma)$, E - $\rho_P^2(MFE, \lambda_L)$, F - $\rho_P^2(MFE, \psi_B)$.

the performance measure MFE, as compared to the dynamical landscape measures. A second trend is that the correlation of the topological measures decrease with increasing change frequency γ. A possible explanation is that change frequency seems to be the fundamental factor determining the MFE, see Fig. 13.9, where γ seems to set the level of the curves but not their form, which is done by ε and so by the other landscape measures. If γ gets larger, that is, the evolutionary algorithm has more generations to find the optimum, then the algorithms' performance is no longer so heavily influenced by topological problem hardness. The performance of an evolutionary algorithm having enough time to search for the maxima is weaker affected by the difficulties that are accounted for by the topological landscape measures. For the dynamical landscape measures severity, Lyapunov exponents and bred vector dimensions the correlations are generally much weaker but such a ceasing relationship for increasing γ is also not obtained. On the contrary, in some cases these measures seem to predict the algorithms' behavior even stronger for larger γ than for smaller ones. A comparison between the two types of correlation coefficients considered here yields rather equivocal results, although Pearson correlation ρ_P^2 seems to give slightly stronger indications, particularly for the topological landscape measures.

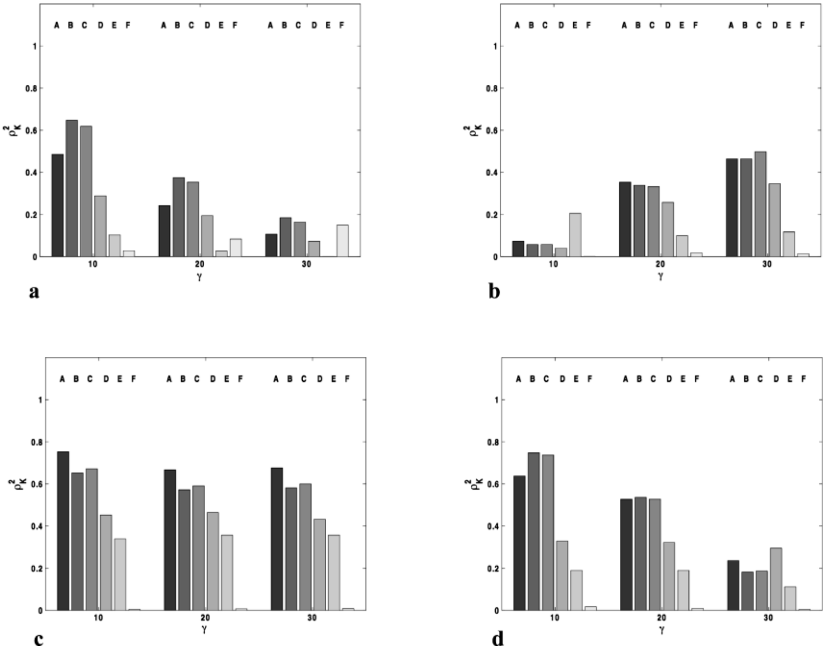

Fig. 13.11 Correlation analysis using Kendall rank–order correlation between topological and dynamical landscape measures and the behavior of the evolutionary algorithm expressed by the mean fitness error MFE for different implementations: a) hyper-mutation, b) self–adaption, c) direct memory, d) abstract memory. A - $\rho_K^2(MFE,\#_{LM})$, B - $\rho_K^2(MFE,\lambda_R)$, C - $\rho_K^2(MFE,H_{IC})$, D - $\rho_K^2(MFE,\sigma)$, E - $\rho_K^2(MFE,\lambda_L)$, F - $\rho_K^2(MFE,\psi_B)$.

So, the question if linear relationships (Pearson) are to prefer over piecewise linear (Kendall) is not definitely answerable.

A general exception to these rules are the results for self–adaption. Here, an implementation detail might offer some clues. Self–adaption is the only of the four implementations that does not directly and externally–triggered react on a landscape change. The other three implementations detect a change and immediately response with diversity enhancing actions such as hyper–mutation or inserting individuals from the memory. Self–adaption, on the other hand, relies upon the mutation rates to adjust themselves over a certain number of generations as the result of the self–adaption process. This might be a reason why the problem solving abilities of the self–adaption scheme, particularly for small γ, are less determined by the problem hardness accounted for by the landscape measures, while the actual performance results are comparable to the other implementations. For self–adaption this seems to mean that the population's diversity is high enough all the time. So, while the other three schemes experience jump–like increases in their diversity as a result of the actions carried out after a detected change, but also a rapidly deterioration of diversity afterwards, self–adaption includes a continuing and not dwindling diversity management that makes it more independent from the landscape properties.

13.6 Concluding Remarks

In this chapter of the book, we have considered the behavior of evolutionary algorithms in dynamic fitness landscapes that exhibit spatio–temporal chaos. These landscapes can be constructed from reaction–diffusion systems or from coupled map lattices (CML) and both kinds of description are related to each other. We have analyzed and quantified their properties using topological and dynamical landscape measures such as modality, ruggedness, information content, dynamic severity and two types of dynamic complexity measures, Lyapunov exponents and bred vector dimension. Four types of evolutionary algorithm implementations, hyper–mutation, self–adaption, and two kinds of memory schemes, direct and abstract memory, were studied and their performance in the spatio–temporal chaotic fitness landscapes was recorded. We used these performance data to relate the algorithms' behavior to the landscape measures using a correlation analysis. So, it was shown that the topological landscape measures correlate stronger with the algorithms' behavior, particularly for landscapes that change frequently. This correlation tends to cease for landscapes with a slower change pattern. Albeit dynamical landscape measures do show weaker correlations, they tend to remain preserved for varying change frequency.

As initially stated, a main point in a theoretical approach to evolutionary computation is to study how properties of the fitness landscape reflect, explain and allow to predict the behavior of the evolutionary algorithm, and vice versa. This question was posed here specifically for dynamical fitness landscapes and our hope is that the given approach might be useful as a starting point for a more general theory of dynamic fitness landscapes, which still is in its infancy. For further developing such a theory it might be helpful to go on taking inspiration from both the theory of static fitness landscapes and of spatially extended dynamical systems. We believe that only by bringing these fields together (which is a variation of the overall topic of the present book) substantial progress can be achieved.

References

1. Aoki, N., Hiraide, K.: Topological Theory of Dynamical Systems. North-Holland, Amsterdam (1994)
2. Arnold, D., Beyer, H.: Optimum tracking with evolution strategies. Evol. Comput. 14, 291–308 (2006)
3. Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies. In: Evolutionary Programming, Genetic Algorithms. Oxford University Press, Oxford (1996)
4. Barreira, L., Pesin, Y.: Nonuniform Hyperbolicity: Dynamics of Systems with Nonzero Lyapunov Exponents. Cambridge University Press, Cambridge (2007)

5. Batterman, R., White, H.: Chaos and algorithmic complexity. Found. Phys. 26, 307–336 (1996)
6. Bosman, P.: Learning and anticipation in online dynamic optimization. In: Yang, S., Ong, Y., Jin, Y. (eds.) Evolutionary Computation in Dynamic and Uncertain Environments, pp. 129–152. Springer, Heidelberg (2007)
7. Boumaza, A.: Learning environment dynamics from self-adaptation. In: Yang, S., Branke, J. (eds.) GECCO Workshops 2005, pp. 48–54 (2005)
8. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Angeline, P., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 1999, pp. 1875–1882. IEEE Press, Piscataway (1999)
9. Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer Academic Publishers, Dordrecht (2001)
10. Branke, J., Kau, T., Schmidt, C., Schmeck, H.: A multi–population approach to dynamic optimization problems. In: Parmee, I. (ed.) Proc. of the 4th Int. Conf. on Adaptive Computing in Design and Manufacturing, pp. 299–308 (2000)
11. Bremen, H., Udwadia, F., Proskurowski, W.: An efficient method for the computation of Lyapunov numbers in dynamical systems. Physica D 101, 1–16 (1997)
12. Chazottes, J., Fernandez, B.: Dynamics of Coupled Map Lattices and of Related Spatially Extended Systems. Springer, Heidelberg (2005)
13. Chlouverakis, K., Sprott, J.: A comparison of correlation and Lyapunov dimensions. Physica D 200, 156–164 (2005)
14. Chrianti, A., Falconi, M., Mantic, G., Vulpiani, A.: Applying algorithmic complexity to define chaos in the motion of complex systems. Phys. Rev. E 50, 1959–1967 (1994)
15. Crutchfield, J., Kaneko, K.: Phenomenology of spatiotemporal chaos. In: Hao, B. (ed.) Directions in Chaos, vol. 1, pp. 272–353. World Scientific, Singapore (1987)
16. Eiben, A., Smith, J.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
17. Francisco, G., Muruganandam, P.: Local dimension and finite time prediction in spatiotemporal chaotic systems. Phys. Rev. E 67, 066204–1–5 (2003)
18. Frederickson, P., Kaplan, P., Yorke, E., Yorke, J.: The Lyapunov dimension of strange attractors. J. Diff. Equations 49, 185–203 (1983)
19. Hordijk, W.: A measure of landscapes. Evol. Comput. 4, 335–360 (1996)
20. Hordijk, W.: Correlation analysis of the synchronizing–CA landscape. Physica D 107, 255–264 (1997)
21. Hordijk, W., Stadler, P.: Amplitude spectra of fitness landscapes. Adv. Complex Systems 1, 39–66 (1998)
22. Hordijk, W., Kauffman, S.: Correlation analysis of coupled fitness landscapes. Complexity 10, 42–49 (2005)
23. Horn, J., Goldberg, D., Deb, K.: Long path problems. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 149–158. Springer, Heidelberg (1994)
24. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments – A survey. IEEE Trans. Evol. Comput. 9, 303–317 (2005)
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. Kallel, L., Naudts, B., Reeves, C.: 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)
28. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge University Press, Cambridge (1999)
29. Kardar, M., Parisi, G., Zhang, Y.: Dynamic scaling of growing interfaces. Phys. Rev. Lett. 56, 889–892 (1986)
30. Katok, A., Hasselblatt, B.: Introduction to the Modern Theory of Dynamical Systems. Cambridge University Press, Cambridge (1995)
31. Katzav, E., Cugliandolo, L.: From coupled map lattices to the stochastic Kardar–Parisi–Zhang equation. Physica A 371, 96–99 (2006)
32. Kauffman, S.: The Origin of Order: Self–Organization and Selection in Evolution. Oxford University Press, Oxford (1993)
33. Ma, K., Jianga, J., Yanga, C.: Scaling behavior of roughness in the two-dimensional Kardar–Parisi–Zhang growth. Physica A 378, 194–200 (2007)
34. Meyer, S., Nieberg, B.H.: Self–adaptation in evolutionary algorithms. In: Lobo, F., Lima, C., Michalewicz, Z. (eds.) Parameter Setting in Evolutionary Algorithm, pp. 47–75. Springer, Heidelberg (2007)
35. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)
36. Miranda, V., Aarão Reis, F.: Numerical study of the Kardar–Parisi–Zhang equation. Phys. Rev. E 77, 031134–1–6 (2008)
37. Morrison, R.: Designing Evolutionary Algorithms for Dynamic Environments. Springer, Heidelberg (2004)
38. Morrison, R., De Jong, K.: A test problem generator for non–stationary environments. In: Angeline, P., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 1999, pp. 2047–2053. IEEE Press, Piscataway (1999)
39. Morrison, R., De Jong, K.: Triggered hypermutation revisited. In: Zalzala, A., Fonseca, C., Kim, J., Smith, A., Yao, X. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2000, pp. 1025–1032. IEEE Press, Piscataway (2000)
40. Ott, E.: Chaos in Dynamical Systems. Cambridge University Press, Cambridge (1993)
41. Patil, D., Hunt, B., Kalnay, E., Yorke, J., Ott, E.: Local low dimensionality of atmospheric dynamics. Phys. Rev. Lett. 86, 5878–5881 (2001)
42. Pesin, Y., Yurchenko, A.: Some physical models of the reaction–diffusion equation and coupled map lattices. Russ. Math. Surv. 59, 481–513 (2004)
43. Primo, C., Szendro, I., Rodríguez, M., López, J.: Dynamic scaling of bred vectors in spatially extended chaotic systems. Europhys. Lett. 765, 767–773 (2006)
44. Richter, H.: Behavior of evolutionary algorithms in chaotically changing fitness landscapes. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 111–120. Springer, Heidelberg (2004)
45. 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)
46. 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)

47. Richter, H.: Coupled map lattices as spatio–temporal fitness functions: Landscape measures and evolutionary optimization. Physica D 237, 167–186 (2008)
48. Richter, H.: Detecting change in dynamic fitness landscapes. In: Tyrrell, A. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2009, pp. 1613–1620. IEEE Press, Piscataway (2009)
49. Richter, H.: Can a polynomial interpolation improve on the Kaplan-Yorke dimension? Phys. Lett. A 372, 4689–4693 (2008)
50. Richter, H., Yang, S.: Memory based on abstraction for dynamic fitness functions. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, A.Ş., Yang, S., et al. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 597–606. Springer, Heidelberg (2008)
51. Richter, H., Yang, S.: Learning behavior in abstract memory schemes for dynamic optimization problems. Soft Computing 13, 1163–1173 (2009)
52. Sheskin, D.: Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press, Boca Raton (1997)
53. 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)
54. 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 2008. LNCS, vol. 5199, pp. 306–315. Springer, Heidelberg (2008)
55. Smith, T., Husbands, P., Layzell, P., O'Shea, M.: Fitness landscapes and evolvability. Evol. Comput. 10, 1–34 (2002)
56. Stadler, P.: Landscapes and their correlation functions. J. Math. Chem. 20, 1–45 (1996)
57. Stadler, P., Stephens, C.: Landscapes and effective fitness. Comm. Theor. Biol. 8, 389–431 (2003)
58. Teo, J., Abbass, H.: An information–theoretic landscape analysis of neuro–controlled embodied organisms. Neural Comput. Appl. 13, 80–89 (2004)
59. Tereshko, V.: Selection and coexistence by reaction–diffusion dynamics in fitness landscapes. Phys. Lett. A 260, 522–527 (1999)
60. Tinós, R., Yang, S.: A self–organizing random immigrants genetic algorithm for dynamic optimization problems. Genet. Program Evol. Mach. 8, 255–286 (2007)
61. Vassilev, V.: Information analysis of fitness landscapes. In: Husbands, P., Harvey, I. (eds.) Proc. Fourth European Conference on Artificial Life, pp. 116–124. MIT Press, Cambridge (1997)
62. Vassilev, V., Fogarty, T., Miller, J.: Information characteristics and the structure of landscapes. Evol. Comput. 8, 31–60 (2000)
63. Weicker, K.: An analysis of dynamic severity and population size. In: Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo Guervós, J., Schwefel, H. (eds.) Parallel Problem Solving from Nature–PPSN VI, pp. 159–168. Springer, Heidelberg (2000)
64. Weinberger, E.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biol. Cybern. 63, 325–336 (1990)
65. Wright, S.: The roles of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones, D. (ed.) Proc. of the Sixth International Congress on Genetics, pp. 356–366 (1932)

66. Xie, F., Hu, G.: Spatio–temporal periodic and chaotic pattern in a two–dimensional coupled map lattice system. Phys. Rev. E 55, 79–86 (1997)
67. Yang, S.: Non–stationary problem optimization using the primal-dual genetic algorithm. In: Sarker, R., Reynolds, R., Abbass, H., Tan, K., Essam, D., McKay, R., Gedeon, T. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2003, pp. 2246–2253. IEEE Press, Piscataway (2003)
68. Yang, S.: Associative memory scheme for genetic algorithms in dynamic environments. In: Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H., et al. (eds.) EvoWorkshops 2006. LNCS, vol. 3907, pp. 788–799. Springer, Heidelberg (2006)
69. Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft Computing 9, 815–834 (2005)