Chapter 10
Fitness Landscapes That Depend on Time

Hendrik Richter

Abstract. Landscapes whose fitness values change with time occur in several contexts. A first is that the evolutionary process takes place in a dynamic environment. Dynamics may be connected to optimization problems with changing objective functions, or generally that conditions apart from the genetic makeup of the population, but massively influencing the evolutionary outcome, are not constant. Mathematically, such dynamic fitness landscapes can be described either by static landscapes that are externally driven to change with time, or by spatially extended dynamical systems which internally and simultaneously define topology and dynamics of the landscape. Another setting for time–dependent fitness are coevolutionary processes where the fitness of a given individual depends on the fitness and the genotype of other individuals in a temporal or spatial fashion. This is known to create coupled, interactive, tunable or deformable landscapes. Such coevolutionary processes induce time–dependence that is population–based and produce landscapes that are codynamic. In this chapter we intend to give an unified overview about issues in and problems of time–dependent fitness landscapes and particularly highlight several types of mathematical descriptions and their properties, similarities and differences.

10.1 Introduction

For understanding evolutionary dynamics, it is useful to have a notion of how the individuals' movements are related to, are partly controlled by, partly directed to and partly influenced by possible paths of increasing or decreasing fitness. One way to obtain such a notion is by fitness landscapes. These landscapes are an influential and important concept in evolutionary biology and evolutionary computation, and recent

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

H. Richter and A.P. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, Emergence, Complexity and Computation 6,
DOI: 10.1007/978-3-642-41888-4_10, © Springer-Verlag Berlin Heidelberg 2014

advances in their understanding are the topic of this book. A conventional view on fitness landscapes is that the fitness of a given point of the landscape is not changing its value for the time frame of consideration. A time frame of consideration may be a run of an evolutionary search algorithm, or an interval of evolutionary development to be observed and studied, or an analysis of the fitness landscape, for instance by a random walk on the landscape. Thus, such kind of fitness landscapes are essentially a static concept. However, if looking at the biological reality on the one hand, or the true nature of optimization problems on the other, we notice that the static view is a significant restriction in understanding interesting phenomena. In biology, there are usually several species and several individuals. They can compete or cooperate with the aim to maximize the share they are getting from limited resources. The outcome of competition or cooperation (or mixing both in an overall strategy) definitely influences survival and reproduction processes, and hence fitness in a timely fashion. Therefore, fitness of a given phenotype is affected by the actions (and therefore by the fitness) of other phenotypes for any interesting time interval. Moreover, environmental conditions are changing frequently and hence bringing timely changes to the fitness of a given phenotypic makeup. The same is true if we consider to solve optimization problems using methods of evolutionary computation. Here the objective function of a multitude of real–world problems is frequently affected by timely changes, for instance in dynamic vehicle [4, 73] and network [12, 13] routing, or job scheduling [35].

In recent years we have seen various attempts to describe, model and understand the phenomena connected to fitness that changes over time [9, 33, 52, 74]. In evolutionary computation this was particularly driven by attempts to design algorithms that perform and behave well in such dynamic environments. In evolutionary biology a main interest is in understanding the role changing conditions play in the overall evolutionary process and in particular what role environmental changes play in survival and reproduction success [28, 39, 69]. This chapter deals with a landscape view on such changing fitness. In particular, we will review in which contexts dynamic fitness landscapes occur, what mathematical descriptions are suitable to be implemented in computable models for conducting numerical experiments, and how the descriptions reflect properties of the underlying dynamic structure. In the following we will focus on landscape approaches in evolutionary computation. This means that the landscape's configuration space is a search space usually originating from an optimization problem and that the landscape may be populated by individuals of an evolutionary search algorithm. However, almost all the discussion applies likewise to a more biological context, where the search space is replaced by a genotypic space that is inhabited by phenotypic realizations.

Dynamic landscape phenomena may occur in different contexts. A first and most obvious is that fitness is directly dependent on time and space. This leads to solving dynamic optimization problems and dynamic fitness landscapes. We will call that explicit time– or space–dependent and will distinguish between *internal dynamics* that is proprietary to the mathematical description of the fitness landscape and *external dynamics* that is generated by a separate driving system and subsequently imprinted on the landscape. Another context is that fitness of one individual depends on the fitness or the phenotype of others. This is the case for different types of coevolutionary settings and creates coupled fitness landscapes [34] (also sometimes called interacting fitness landscapes). If it happens within the frame of the time considered, we also have a situation where fitness is changing with time. This time–dependence is implicit and will be called *population–based dynamics*. However, although there is a substantial amount of work on coevolutionary phenomena, see for instance [49] and references cited there, how this creates dynamics and how the dynamics affect the evolutionary search is rarely addressed. One attempt to make the population–based dynamics in coevolution explicit are deformable fitness landscapes [17, 19]. In the following we aim at giving a unified view about fitness landscapes whose fitness values change with time, and in particular to relate dynamic fitness landscapes to coupled and deformable ones.

In the next section, we will briefly review the current understanding of dynamics in evolutionary processes. This is followed by defining and discussing dynamic fitness landscapes and dynamic optimization problems. Section 10.3 deals with coevolutionary processes. We recall the main ideas and issues of coevolutionary models and algorithms and show how coupled and deformable fitness landscapes derive from them. In particular, we draw connections between these landscapes and the dynamic fitness landscapes considered before. In the concluding section we provide a broader view on fitness landscapes that change with time and outline potential directions of future research.

10.2 Dynamic Fitness Landscapes

10.2.1 Dynamics? What Dynamics?

It seems to be generally acknowledged that every timely change of any given quantity is connected with dynamics. This is in agreement with the word's origin from the ancient Greek *dynamis* ($\delta\acute{\upsilon}\nu\alpha\mu\iota\varsigma$), which can be translated as 'ability', 'capacity', 'potentia' or 'power' for doing or moving or changing something. However, there is some confusion as to whether dynamics is the cause, the effect or the process of changing. In addition, the nature of the changes may differ substantially from case to case with the result that the changes may or may not affect what we consider in a particular study. A defining factor to answer this question is how the time frame of what is considered relates to the time frame of the changes. The relations between these time frames can be interpreted as to define a relative speed of the changes. If the speed of the changes is below a certain threshold, it can be neglected as it does not influence results and outcome. Moreover, sometimes even changes of relevant speed may not be taken into account as they are (or can be considered as) spatially, functionally or structurally separated from the object of study. This clearly applies to studies in natural and artificial evolution. Here, it is frequently of interest what severity the change has in terms of fitness. However, if evolutionary speed is defined as the magnitude of the derivative of the evolutionary relevant quantity with respect to time, severe changes are tantamount to high speed dynamics.

Two types of dynamics can be distinguished. A first is evolutionary dynamics. Evolutionary dynamics describes how individuals of an evolutionary search algorithm move in the search space as generations go by [2, 23]. As each point in a search space has a fitness value, and this fitness value can be seen as proprietary to the individual, the movement can be pictured as to happen on the fitness landscape's surface (also see Chapter 1 of this book). This view remains valid in a more evolutionary biology point of view. Evolutionary dynamics here means the movements of phenotypic realization through the genotypic space. Most important here is that evolutionary dynamics is expressed and counted on a generational base. This is in line with the genetic makeup of an individual being constant during its lifetime. In this view adaption within the lifetime of the individual such as learning, phenotypic adaption, plasticity and polymorphism is not directly taken into account. Of course, such traits may tune the fitness of the individual, but the ability to do so is clearly fixed in the phenotype. So, in the end only genetically inheritable traits form the base of fitness, which is considered to be a unique (and therefore constant) value for the time interval of a generation. This allows the interpretation that fitness of an individual is a single value of 'lifetime fitness', the calculation of which is subject to all activities within the lifetime. Considering this, it should be clearly pointed out that evolutionary dynamics as such takes place on a static fitness landscape.

Next to evolutionary dynamics, another type of timely change is of interest here, which we call environmental dynamics. Under environmental dynamics we subsume changes outside the considered individual that have influence on its fitness. That might be abiotic changes such as shifts in climate or other physical attributes of the living space, or biotic changes such as alterations in the food and/or prey abundance, or predator and/or parasites existence, or interaction between species or within them. Our interest is in evaluating the effect these environmental changes have on the fitness of individuals. In other words, we study how environmental dynamics relates to evolutionary dynamics. Therefore, it is sensible to relate the time scale of the environmental changes to the generational time of evolutionary dynamics. As fitness of an individual is a single (constant) value for each generation, there are two consequences. A first is that changes that occur over the lifetime of an individual should be considered as to aggregate in their effects in order to result in a lifetime fitness; a second is that the effects of environmental changes should be counted at discrete points of time. From these thoughts it also follows that the environmental changes are the source for the dynamics in dynamic fitness landscapes.

To summarize we define a fitness landscape to be dynamic if (and only if) its fitness takes different values for the same search space point over the time frame of consideration. In our understanding there are many different values over time (meaning clearly more than one) and the changes are scattered over the time frame. A most obvious example for that is a fitness landscape that changes frequently over the run–time of the evolutionary search algorithm. In a more biological context this relates to substantial environmental dynamics while studying a long series of generational evolution of a single species. This is in contrast to situations where the fitness landscape might be subject to some parameters, which can be tuned or adapted for each time frame of consideration. This is known as tuneable fitness landscape and almost always means that fitness values of the landscape are affected, that is tuned, but only once and before the time frame of consideration. This is particularly relevant in considering NK and NKC landscapes and will be discussed below. The time frame of consideration may be an evolutionary run, or studying an interval of an evolutionary development, or a random walk if we design a landscape analysis. Mathematically, this understanding implies that the fitness is a variable depending on time, as opposed to fitness as a parameter to be set before a (numerical) experiment.

In evolutionary computation a fitness landscape is often linked to an optimization problem. We next extend this view to a dynamical setting. We first look at a static optimization problem. It consists of an objective function (frequently equated with a fitness function in evolutionary computation) $f(x)$ defined over some search space S with $x \in S$. We assume that the search space is metric, or that there are some other arrangements about which search space points are next to each other. Optimization means to find the lowest (or highest) value of $f(x)$ and its coordinates among all $x \in S$:

$$f_S = \max_{x \in S} \quad f(x), \tag{10.1}$$

with the location $x_S = arg\, f_S$.

The static optimization problem in Equation (10.1) can be thought of as becoming dynamic by solving it not just once, but somehow modified for a second time. For a modified fitness function $f^*(x)$, we may write the modified problem as

$$f_S^* = \max_{x \in S} \quad f^*(x). \tag{10.2}$$

To rewrite the two static problems in Equations (10.1) and (10.2) as one dynamic problem, we introduce the time variable $k \in \mathbb{N}_0$ (which is nothing other than a counting variable) and define the dynamic fitness function $f(x,k)$, where

$$f(x,0) = f(x), \quad f(x,1) = f^*(x).$$

We may carry on with doing so for the next modification of $f(x)$ to obtain $f(x,2)$, and so on. Hence, a dynamic optimization problem is

$$f_S(k) = \max_{x \in S} \quad f(x,k), \quad \forall k \geq 0 \tag{10.3}$$

with the solution trajectory $x_S(k) = arg\, f_S(k)$. To define a dynamic optimization problem in the given way suggests that there might be more than one way to define a dynamic problem out of a series of static problems, or that we may obtain different dynamic problems out of modifying one and the same static problem. The discussion above may also imply that the modifications alter the problem only lightly. However, even if the modified fitness function $f^*(x)$ is fundamentally different from the function $f(x)$ it is still possible to define both problems such that the one dynamically originates from the other. However, such a point of view is rather futile in terms of usefulness for evolutionary computation. As both problems here have hardly any relations to each other, the solution of one of them gives no information employable for evolutionary solving the other more efficiently. The best way to solve them both by evolutionary means is to let the algorithm run independently for both problems. In other words, to consider a sequence of problems to be dynamic makes most likely more sense if the problems show some alikeness and relation to each other. A main assumption here is that similar problems are best solved with similar algorithms. Moreover, there should be a way to utilize these relations for equipping the evolutionary search algorithm with favorable settings. These settings might concern parameters (for instance general parameters such as population size or diversity management parameters such as hyper–mutation rate or random immigrant proportion) or genetic operators (what kind of selection, recombination or mutation) or collectible information for equipping triggered diversity management schemes such as memory or anticipation/prediction.

10.2.2 Definition of Dynamic Fitness Landscapes

In the last section we have shown how a series of static problems can create a dynamic problem. In the same way we may regard a series of static fitness landscapes as a dynamic fitness landscape. Next, we will formalize this approach. A static fitness landscape Λ_S can be defined by (see e.g. [24, 68])

$$\Lambda_S = (S, n, f), \tag{10.4}$$

where S is the search space with elements $x \in S$, $n(x)$ is a neighborhood function which orders for every $x \in S$ a set of direct and possibly also more distant neighbors (and hence makes the space at least measurable, if not metric), and $f(x) : S \to \mathbb{R}$ is the fitness function giving every search space element a fitness value. The search space is either the product of a genotype–to–phenotype–to–fitness mapping or constructed from encoding and distancing the set of all possible solutions of an optimization problem. Either way it is basically the representation that the evolutionary algorithm uses (for instance binary, integer, real, tree etc.) and the design of the genetic operators that defines the search space, and also its neighborhood structure. This is in line with the general understanding that the concept of fitness landscapes is particularly useful for studying how the evolutionary search algorithm interrelates with the fitness function [24, 67, 71]. Moreover, as shown in [23], the neighborhood structure of a fitness landscape may vary with variation of the genetic operators. Hence, an analysis of the fitness landscape can be helpful for designing genetic operators as it gives insight into which design is more likely than others to belong to the landscape easiest searchable on average [71]. If the representation is fixed, for instance as real numbers, then the search space frequently has a metric and the neighborhood structure is inherent. The exact design and the parameters of the genetic operators, for instance the mutation strength, define which points can be reached on average from a given starting point in the landscape within one generation.

The geometrical interpretation that is central to the intuitive understanding of fitness landscapes is particularly visible if $S \subseteq \mathbb{R}^2$ (see Figure 10.1). The interpretation means that a search space in connection with a neighborhood structure forms a location (and hence makes the space measurable) and that fitness can be viewed as a height over the location. In other words, fitness is a property belonging to the search space as the space's orthogonal projection. Therefore, search space points with high fitness appear as peaks, while low fitness regions are valleys, and points with the same fitness form plateaus. Solving the corresponding optimization problem hence means, according to Equation (10.1), to find the highest peak $f_S = \max_{x \in S} f(x)$ and its location $x_S = arg\, f_S$. The individuals of an evolutionary algorithm used to solve the given optimization problem can be thought of as to populate the fitness landscape. The design of the genetic operators (basically selection, recombination, and mutation) is meant to organize such that they (at least in average) perform a climbing of the hills in the fitness landscape and ideally find the highest one, even in the presence of other (but smaller) hills. With the movement, the individuals contribute to the dynamics of the population and hence to the evolutionary dynamics. Note that in analyzing these processes a distinction can be made between the dynamics generated by genetic variation (mutation and recombination) and the dynamics generated by corrective guidance (selection), which is of considerable interest for fine–tuning the genetic operators.

In order to define a dynamic fitness landscape, there is the need to set how the elements in Equation (10.4) change over time. With respect to the view of a dynamic fitness landscape that is generated by a series of static landscapes, defining dynamics means to explain how one landscape produces the temporary subsequent one. All of the landscape's three defining ingredients (i.e. search space S, fitness function $f(x)$, and neighborhood structure $n(x)$) can, at least in principle, be dynamically changing. Hence, we additionally need a time set and mappings that indicate how S, $f(x)$ and/or $n(x)$ evolve with time [58, 59]. Dynamic optimization problems considered in the literature so far address all these possibilities of change to some extend. Whereas a real alteration of the fundamental components of a search space such as dimensionality or representation (binary, integer, discrete, real, etc.) is really rare, a change in the feasibility of individuals is another and less substantial kind of dynamic search space and is discussed within the problem setting of dynamic constraints [43, 60, 61]. Next is a changing neighborhood structure which can partially be found in the context of works on dynamic routing [4, 73]. However, most of the work so far has been devoted to time–dependent fitness functions [37, 42, 56, 62, 63, 65, 72, 78]. For these reasons, and to keep the next definition simple, a dynamically changing search space and neighborhood structure is omitted in the following, only the fitness function is time–dependent. A definition encompassing all three landscape components to be dynamic can be obtained by adding transition maps for the other quantities.

We define a dynamic fitness landscape as

$$\Lambda_D = (S, n, K, F, \Phi_f). \tag{10.5}$$

Equivalent to the static landscape of Equation (10.4) the search space S represents all possible solutions to the optimization problem and the neighborhood function

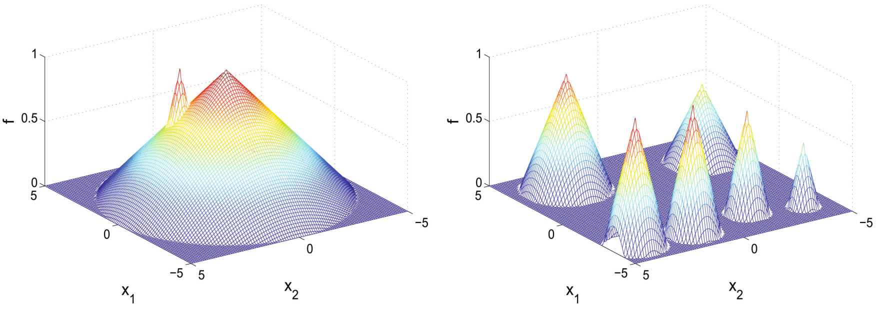

Fig. 10.1 A static fitness landscape in 2D: static peaks that may become moving peaks

$n(x)$ gives a set of neighbors to every search space point. The time set $K \subseteq \mathbb{Z}$ provides a scale for measuring and ordering dynamic changes; F is the set of all fitness functions in time $k \in K$ and so every $f \in F$ with $f : S \times K \to \mathbb{R}$ also depends on time and gives fitness values to every search space point for any $k \in K$. The transition map $\Phi_f : F \times S \times K \to F$ defines how the fitness function changes over time. The map must satisfy the temporal identity and composition conditions, that is $\Phi_f(f,x,0) = f(x,0)$ and $\Phi_f(f,x,k_1+k_2) = \Phi_f(\Phi_f(f,x,k_1),x,k_2)$, $\forall f \in F$, $\forall x \in S$, $\forall k_1,k_2 \in K$ and the spatial boundary conditions $\Phi_f(f,x_{bound},k) = f(x_{bound},k)$, $\forall f \in F$, $\forall k \in K$ where x_{bound} is the boundary set of search space S. With these definitions we assume that the changes in the fitness landscape happen (or come into effect) at discrete points in time and are the result of comparing the landscape at points in time k to the following points $k+1$. This is in line with fitness landscapes being a tool for analyzing the behavior of evolutionary algorithms. A generation of an evolutionary algorithm can be defined as the time interval between subsequent fitness function evaluations of the whole population. In other words, a generation indicates the time between serial and self–contained steps in the solution finding process. As fitness evaluation in an evolutionary algorithm usually takes place just once in a generation, a difference in fitness can only be noticed by the algorithm at discrete points in time. Hence, if we model the changes by a fitness landscape, the most natural and straightforward time regime is discrete time.

An intuitive geometrical interpretation of a static fitness landscape still holds to some extend for the dynamic case. The main difference is that the hills and valleys move within the search space and/or change their topological form. This includes that hills grow and shrink, valleys deepen or flatten, or the landscape completely or partially turns inside out. According to Equation (10.3), the corresponding dynamic optimization problem now reads $f_S(k) = \max_{x \in S} f(x,k), \forall k \geq 0$ which yields the temporarily highest fitness $f_S(k)$ and its solution trajectory $x_S(k) = arg\, f_S(k), \forall k \geq 0$.

As before the individuals of the evolutionary algorithm are meant to climb the hills, and moreover to follow if they are moving and find hills that dynamically appear. Even from such a simplifying picture it becomes obvious that the standard genetic operators (selection, recombination, mutation) might not be sufficient to perform the task. Indeed, there exists a multitude of modifications to deal with the changes induced by a dynamic fitness landscape, for instance different types of memory [7, 60, 65], random–based diversity enhancement techniques [42, 72] or anticipation and prediction schemes [5, 62, 66]. Equation (10.3) describes the dynamic optimization problem and its solutions. However, for evaluating the performance and the behavior of an evolutionary algorithm used to deliver such solutions, other quantities can be more interesting, meaningful and significant. These quantities usually generalize the solution trajectory in Equation (10.3) over the run time and/or runs of the evolutionary algorithm, may include data from the evolving population's fitness and make them statistically evaluable. See [1, 75] but also Chapter 14 and references cited there for an overview about dynamic performance evaluation.

Above, we have argued for dynamic fitness landscapes useful in evolutionary computation to have a time regime that only allows changes at discrete points in time. We will end this section with relaxing this kind of modelling and reviewing the effects this has for mathematical descriptions of dynamic fitness landscapes. At first it can be noted that the definition in Equation (10.5) is still valid and meaningful if we consider a real value time set $K = \mathbb{R}$. For not confusing the elements of the discrete valued time set with the real one, we write $t \in \mathbb{R}$. A main consequence of such a real valued time set is that we can mathematically describe dynamic fitness landscapes that change continuously in time. Once we have defined a search space that is metric (or at least measurable) a fitness function can be defined in very general terms. Every $f : S \to \mathbb{R}$ that maps points from the said search space to a scalar variable can be interpreted as a fitness function. Hence, defining the timely change of the function $f(x,t)$ defines a dynamic fitness function. If there are a countable number of possible solutions x_i (and it hence is a combinatorial optimization problem) the dynamics of each of the solutions can be put as depending on the fitness of the solution x_i at time t as well as on the fitness of the μ other solutions

$$\frac{df(x_i,t)}{dt} = \psi_i\left(f(x_1,t),\ldots,f(x_i,t),\ldots,f(x_\mu,t)\right). \tag{10.6}$$

With Equation (10.6) we obtain as dynamic fitness landscape a lattice of (nonlinear) ordinary differential equations (ODEs). There are a considerable number of dynamic combinatorial optimization problems for which the timely evolution of the fitness $f(x_i,t)$ does not depend on all the μ solutions but only on solutions neighboring x_i. Hence, Equation (10.6) modifies to

$$\frac{df(x_i,t)}{dt} = \psi_i^*\left(f(x_i,t),f(n(x_i),t))\right). \tag{10.7}$$

For a non–countable number of possible solutions the dynamics of the fitness $f(x,t)$ may be expressible by a nonlinear partial differential equation (PDE). If the properties of the search space allow to define spatial derivatives, we obtain (for phenotypes described by n–dimensional vectors) the PDE

$$\frac{\partial f}{\partial t} = \psi\left(f(x,t),\frac{\partial f}{\partial x_1},\cdots\frac{\partial f}{\partial x_n},\ldots,\frac{\partial^2 f}{\partial x_1^2},\ldots,\frac{\partial^2 f}{\partial x_1 \partial x_n}\cdots\right). \tag{10.8}$$

With Equation (10.8), we have a description of the changing fitness as a spatially extended dynamical system. The timely evolution of fitness values specified by the PDE does not depend on all the other fitness values or an explicitly assignable neighborhood, but on the spatial derivatives $\frac{\partial f}{\partial x_i}$ of first and higher order. Geometrically interpreted, this means the timely evolution of the fitness of any search space point is subject to differences in fitness of points laying infinitesimally around that point. To express it differently, the fitness evolution is a function of the curvature the fitness creates in the landscape, or how the fitness deforms the landscape spatially.

It could be argued that Equations (10.6), (10.7) and (10.8) are the most basic way of describing the timely evolution of a fitness landscape as any change in nature must be a change in physical entities whose dynamics is in continuous time. However, it is most unlikely that the equations can be directly used as computational models because they do only in exceptional cases have an analytic formula–like closed solution. Any computational approach therefore requires to discretize time for Equations (10.6) and (10.7) and time as well as space for Equation (10.8). At the end of the process there should be an iterative generation law describing how a fitness value $f(x,t)$ evolves into $f(x,t+\delta t)$, with δt being a small time increment. For Equation (10.7) (likewise for Equation (10.6)) we thus end up with

$$f(x_i,t+\delta t) = \Psi_i\left(f(x_i,t), f(n(x_i),t)\right) \tag{10.9}$$

and for Equation (10.8) with

$$f(x,t+\delta t) = \Psi\left(f(x,t), f(x_1+\delta x_1,t), f(x_2+\delta x_2,t), \ldots, f(x_n+\delta x_n,t)\right) \tag{10.10}$$

where Ψ_i and Ψ are some generator mappings. With these equations we obtain a temporal and spatial discretization employable in numerical algorithms for calculating the evolution of fitness values recursively forward in time. To formalize this computational approach we go back to the discrete time variable k and introduce a temporal renormalization. With this, and because $x_i + \delta x_i$ imposes a spatial discretization and can be interpreted as a neighborhood, we may rewrite and generalize Equations (10.9) and (10.10) as

$$f(x_i,k+1) = \phi_i\left(f(x_i,k), f(n(x_i),k)\right) \tag{10.11}$$

with ϕ_i being a generator mapping.

Note that this neighborhood $n(x_i)$ is not necessarily the same neighborhood used in laying out and distancing the search space according to the definition in Equation (10.5). There might be one neighborhood that defines which search space points are next to a given point, and another neighborhood that expresses which fitness values are influencing the timely evolution of that point. With formulating Equation (10.11) we argue that this recursive evolution law is not only indispensable in calculating a dynamic fitness landscape but also an integral part in defining it. Solving Equation (10.11) yields the transition map Φ_f in the definition in Equation (10.5). So, no definition of a dynamic fitness landscape is complete without giving such an equation. The next section reviews some examples of dynamic fitness landscapes and such equational descriptions are explicitly given.

10.2.3 Examples of Dynamic Fitness Landscapes

As shown in the definition given in Equation (10.4), algebraic equations can be used to describe static fitness landscapes. As further shown in Equation (10.5), for dynamic fitness landscapes we need to add a mathematical prescription for evolving the fitness values forward in time. In principle, there are two ways for doing so. A first is to take the algebraic description of a static fitness function, select some terms from these equations, and define how these terms depend on an additional (and usually external defined) time regime. Geometrically speaking, this means that we define dynamics laws for how selected topological features in the landscape evolve with time. This only implicitly results in explaining the landscape's dynamics in the whole. On the other hand, this also implies that we have, in fact, dynamics only for certain elements in the landscape. By changing these elements dynamically, we implicitly also describe how the neighborhood of the peaks behave with time. We will call this kind of dynamics generation *external*.

The easiest and most obvious way for defining such time–dependent landscapes is to use a static landscape and a set of rules for changing certain features in the landscape with time. A well–known example is the so–called moving peak benchmark [7, 41], which uses as fitness function a static n–dimensional field of peaks (or cones) on a zero plane $f(x) = \max\left\{ 0\,, \max\limits_{1\leq i\leq N} [h_i - s_i\|x - c_i\|] \right\}$ (see also Figure 10.1). Here, $x \in S$ is an element in search space $S \subset \mathbb{R}^n$, c_i is the coordinate of the i–th cone, there are N cones in total, and h_i and s_i are the height and slope, respectively, of the i–th cone. By using the discrete time variable k, we may set dynamic sequences for coordinates $c(k)$, heights $h(k)$ and slopes $s(k)$ to obtain a dynamic fitness landscape with moving peaks

$$f(x,k) = \max\left\{ 0\,, \max_{1\leq i\leq N} [h_i(k) - s_i(k)\|x - c_i(k)\|] \right\}. \tag{10.12}$$

For all static landscapes $f(x)$ we may similarly identify topological features in the the landscape's mathematical description and change elements of the mathematical description dynamically. There are other problems such as dynamic sphere, dynamic Ackley, dynamic Rosenbrook etc., but also dynamic combinatorial optimization problems such a dynamic knapsack, dynamic royal road or dynamic bit–matching [70] that fit into this category, see e.g [15, 44] for an overview of such kind of dynamic problems.

If a dynamic fitness landscape relies on external dynamics, the question of how to generate dynamic sequences arises. A first step is to select terms in the algebraic fitness landscape description that are to change with time. For the moving peak benchmark in Equation (10.12), these are $c_i(k)$, $h_i(k)$ and $s_i(k)$. The dynamic changes are induced by moving sequences

$$z = (z(0), z(1), \ldots, z(k), z(k+1), \ldots) \tag{10.13}$$

that is

$$c_i(k) = z_{ci}(k), \quad h_i(k) = z_{hi}(k), \quad s_i(k) = z_{si}(k). \tag{10.14}$$

In principle, the dynamic changes can be of three types:

- regular dynamics
- chaotic dynamics
- random dynamics.

The moving sequences $z(k)$ can be generated according to these types. Regular changes are usually obtained by analytic coordinate transformations, for instance cyclic dynamics where each $z(k)$ repeats itself after a certain period of time and shows recurrence or translatory dynamics where the quantities ascribe a pre–defined track or tour. The period of the recurrence and the cycle width can be adjusted and normalized so that moving sequences become comparable. For instance, cyclic dynamics can be generated by

$$z_i(k) = g_i(k) = \sin(\omega_i k + \delta_i) \tag{10.15}$$

with ω_i and δ_i appropriate frequencies and phases. Note that an analytic equation such as Equation (10.15) enjoys the property to be not recursive. The value of $z(k)$ can be calculated without knowing $z(k-1)$. This also means a fitness landscape externally driven by Equation (10.15) has strictly speaking no recursive evolution law as given in Equation (10.11), but is knowable beforehand for any given point in time k. It is hence completely predictable.

Chaotic changes can be generated by a discrete–time dynamical system,

$$z(k+1) = g(z(k)), \tag{10.16}$$

which is recursive. Such systems are known to show chaotic behavior for certain parameter values and initial states $z(0)$, for instance the generalized Hénon map. Refer to [55, 56] for details of the generation process. For using these moving sequences in numerical experiments, there might be the need for preprocessing as (depending on the dynamical systems used) the amplitudes $z(k+1)$ might be not unitary. If so, a re–normalization should take place. We get random changes if we select that each $c_i(k)$, $h_i(k)$, $s_i(k)$ for each k is an independent realization of, for example, a normally or uniformly distributed random variable. Again, the statistical properties of the random variable should guarantee (maybe after renormalization) comparability. In some sense, also fitness landscapes externally driven by a random process have no recursive evolution law as given in Equation (10.11). The main difference is here that the value of $z(k-1)$ does in no way specify the value of $z(k)$. The evolution of a random fitness landscape is only statistically describable. A general feature of the three types of dynamics is that regular dynamics is completely predictable, chaotic dynamics is short–term predictable, and random dynamics unpredictable. This property transforms to fitness landscapes externally driven by these dynamics.

A second example for external drive is the XOR-generator by Yang [77, 78]. This generator can be used for any binary–encoded static function $f(x)$ as follows. For each discrete step in landscape time k, a XOR mask $M(k)$ is incrementally calculated by

$$M(k) = M(k-1) \oplus T(k), \tag{10.17}$$

where "$\oplus$" is a bitwise exclusive-or (XOR) operator (i.e., $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, and $0 \oplus 0 = 0$) and $T(k)$ is an intermediate binary template generated for time step k. $T(k)$ is generated with $\rho \times l$ ($\rho \in (0.0, 1.0]$) randomly selected bits set to 1 while the remaining bits are set to 0. For the initial step $k = 1, M(0)$ is set to a zero vector, i.e., $M(0) = 0$. The static fitness function $f(x)$ finally becomes dynamic by

$$f(x,k) = f(x \oplus M(k)). \tag{10.18}$$

The XOR generator is hence a way to externally drive a binary fitness function by the recursive law in Equation (10.17).

In a second approach to describe dynamic changes, we may formulate a general law for the fitness landscape's time evolution that applies to all fitness values in the search landscape. Hence, the specification of the timely changes are part of the mathematical description of the landscape. Thus, the fitness of every point $f(x, k+1)$ may depend on the fitness one time step before, $f(x,k)$ and the (element–wise) fitness values of all of its neighbors, $f(n(x),k)$ (see Equation (10.11)). With such a description we have formulated a dynamic fitness landscape as a spatially extended dynamical system. To have such description means that the topology and the dynamics of the fitness landscape are generated simultaneously and by the same equation. In other words, the dynamics here is *internal* to the fitness landscapes. An example for such an evolution law is fitness landscapes constructed from ordinary differential equations (ODE), partial differential equations (PDE) and coupled map lattices (CML). A special property of this type of dynamic fitness landscape is that not only is time discrete but also the search space has a countable number of elements. Such a search space characteristic corresponds strictly speaking to combinatorial optimization problems (see [58, 59] for further discussion).

For a two–dimensional real valued search space S, a dynamic fitness landscape can be viewed as the time evolution of the surface over a 2D plane at point x and time t. Such a general dynamic 2D fitness landscape that describes the dynamics of the fitness value $f(x_1, x_2, t)$ with continuous spatial variables (x_1, x_2) and continuous time t can be modelled by the parabolic PDE

$$\frac{\partial f}{\partial t} = a_1 \left(\frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} \right) - a_2 g_1 \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right) + g_2(f), \tag{10.19}$$

where a_1, a_2 are coefficients and g_1, g_2 are mappings. The dynamic fitness landscape in Equation (10.19) can be interpreted as a reaction–diffusion system with an additional nonlinear term and is a special case of the general description given in Equation (10.8). This type of PDE has close resemblance to the Kardar–Parisi–Zhang (KPZ) equation [27], which has been proposed to model surface growth. The main difference is that the KPZ equation includes an explicit stochastic (Gaussian noise) term. Recently, the KPZ equation has been intensively studied [30, 36, 40], with the relation to coupled map lattices (CML) as a central question. Clearly, both are models of extended dynamical systems. Also, and as mentioned before, a numerical solution to a PDE always requires some kind of discretization of space and time. An alternative strategy to a study by any of the methods for numerically solving the PDE therefore appears to consist of a study of the corresponding CML and their mutual dynamical properties.

We will next relate the PDE–based landscape, characterized in Equation (10.19), to a landscape based on a CML [57, 58]. It has been shown that the CML landscape can be obtained by a spatial and temporal discretization of the PDE landscape [59]. For the CML, we lay out a lattice grid with $I_1 \times I_2$ equally sized cells, which builds a $2D$–structure. For every discrete time step k, $k = 0, 1, 2, \ldots$, each cell is characterized by its height

$$f(x_1, x_2, k), \quad x_1 = 1, 2, \ldots, I_1, \quad x_2 = 1, 2, \ldots, I_2, \tag{10.20}$$

where (x_1, x_2) denote the spatial indices in vertical and horizontal directions, respectively (refer to Figure 10.2). We interpret this height, $f(x_1, x_2, k)$, as fitness according to the geometrical metaphor of a fitness landscape. It is subject to changes over time, which are described by the two–dimensional CML with nearest–neighbor coupled interaction [10, 25]:

$$\begin{aligned} f(x_1,x_2,k+1) = (1-\varepsilon) g(f(x_1,x_2,k)) + \frac{\varepsilon}{4}\Big[\, & g\left(f(x_1-1,x_2,k)\right) + g\left(f(x_1+1,x_2,k)\right) \\ & + g\left(f(x_1,x_2-1,k)\right) + g\left(f(x_1,x_2+1,k)\right) \Big], \end{aligned} \tag{10.21}$$

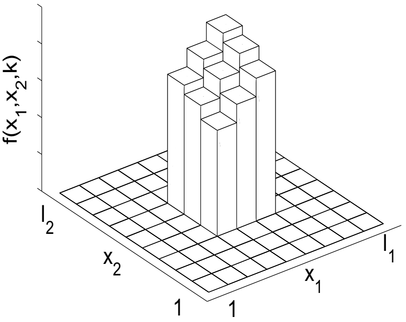

Fig. 10.2 The coupled map lattice (CML) of Equation (10.21) as an example for internal dynamics in a fitness landscape

where $g(f(x_1,x_2,k))$ is a local mapping function and ε is the diffusion coupling strength. As local mapping function we use the logistic map,

$$g(f(x_1,x_2,k)) = \alpha f(x_1,x_2,k)(1-f(x_1,x_2,k)). \qquad (10.22)$$

It is a nonlinear map with the parameter $0 < \alpha < 4$, which is defined for the unit interval $f \in [0,1]$. For some parameter α, the map given by Equation (10.22) exhibits chaotic behavior, for instance in the parameter interval $\alpha \in [3.57,4]$. The local chaotic behavior is distributed to other areas of the lattice by coupling. So, it is the source of spatio–temporal chaos in the extended dynamical system. Finally, we need to set the period boundary conditions as

$$\begin{aligned} f(I_1+1,x_2,k) &= f(1,x_2,k), \\ f(x_1,I_2+1,k) &= f(x_1,1,k). \end{aligned} \qquad (10.23)$$

Initialization of the CML is done by initial heights $f(x_1,x_2,0)$, being realizations of a random variable uniformly distributed on $[0,1]$. The spatio–temporal behavior of the CML depends on the lattice size $I_1 \times I_2$ and two parameters, the coupling strength ε and the nonlinear parameter α. The timely evolution of the CML given in Equation (10.21) also provides the recursive evolution law of the dynamic fitness landscape of Equation (10.11).

The CML is known to exhibit a rich spatio–temporal behavior, including different types of spatio–temporal periodicity and chaos, quasi–periodicity and pattern formation. The CML is therefore an instructive example for the principle of generating high–dimensional complex spatio–temporal dynamics by using local chaos created by a low–dimensional mechanism that is transmitted to a spatial extension by coupling. The obtained fitness landscape shows a similar complex behavior. Refer to [57, 58, 59] for detailed studies.

10.3 Coevolution, Codynamics and Their Fitness Landscapes

10.3.1 Coevolutionary Dynamics

This section reviews coevolutionary algorithms and their fitness landscapes. This is done with the aim to provide background for relating coevolutionary fitness landscapes, namely coupled and deformable landscapes, in the framework of dynamic fitness landscapes discussed in the previous section. There is an ongoing debate about the question whether or not fitness landscapes that originate from coevolutionary processes could (or should) be treated as dynamic and what possible benefits (if any) such a treatment could have. In the following we will not argue strongly for either view, but we will demonstrate that such a view is possible.

Coevolutionary algorithms differ structurally from evolutionary algorithms for several reasons. First, there is usually more than one population. Thus, there is no longer only one genotypic space populated by individuals of one species, but several spaces in which the evolutionary dynamics of other species take place. In this view, coevolution operates on the level of phenotypes. Anyway, coevolution also has an impact on the fitness landscape as a whole, that is, on the level of genotypes. For the dynamic interdependency on this level of the landscape we will introduce the term *codynamics*. Coevolution between species results in codynamics between the respective fitness landscapes. We will restrict ourselves here to coevolution between two species (and hence the dynamic interaction, that is, codynamics, of two fitness landscapes). Systems with more than two species can, at least in principle, be treated along the same mathematical framework.

A second reason is that the fitness of the individuals of both species are interdependent and as a consequence coevolution relies upon an alternative concept of how the fitness of an individual is generally defined. This is the crucial element of coevolution. There are some implementations [18] that only use a single population to create coevolutionary effects, but these implementations as well as the ones with multiple populations all share the following property: they use a population–dependent fitness. This fitness is subject to the progress the search process itself is making. It therefore is also called a subjective fitness, as opposed to objective fitness in conventional evolutionary computation. As the search process is dynamic, this necessarily results in dynamic fitness. This is in contrast to fitness evaluation considered so far. For an evolutionary algorithm (and for static problems) the fitness value is a property of a search space point (or a given genotype) and is constant for the entire run–time. So, if in the course of the evolutionary search, the same point were to be visited again by an individual, it would yield the same fitness value as in the visit before. In other words, a reevaluation of a specific genotype always gives the same fitness value. For an evolutionary algorithm and a dynamic problem the search space points may change their fitness values but this happens because the landscape is internally or externally changing with time. This takes place independently from the evolutionary search process and is hence unconnected to the fitness values of the population's individuals. There is no feedback from the evolutionary search to the landscape.

In coevolutionary algorithms fitness is assigned differently. The fitness of an individual is the result of interaction with other individuals. The individuals that serve to interact with the one for which a fitness value is to be assigned are called evaluators. The fitness of an individual at a given point in run–time therefore depends on which individuals are actually selected to interact with, and also on the current fitness of these evaluators. As a consequence, a given point in search space (a genotype) can have a completely different fitness value if it were to be reevaluated. Therefore, a specific genotype frequently has variable, time–dependent fitness where the time–dependence is induced explicitly by the generational search. This is because in the reevaluation other individuals may act as evaluators and even if the same individuals are taken a second time, they might have different fitness values. Only if the evaluators remain the same, and their fitness values do not change in the evolutionary run, we would fall back to the situation where a genotype has a fixed fitness value. This, of course, would make a pointless coevolutionary algorithm. In addition, there are even coevolutionary algorithm implementations that deviate from the simple design principles set out above. Elevators are usually a subset of a population as an interaction with all the members of all possible populations, called full mixing or complete mixing, might be time–consuming and therefore computationally expensive. There are for instance implementations that vary the number of individuals for interaction or the number of interactions with time, either depending on the run–time of the algorithm [46] or depending on some internal states of the algorithm [47]. Other implementations use an archive of past solutions to select evaluators from [45, 48, 64]. All these mechanisms for calculating fitness dramatically complicate and obscure the relationship between fitness and genotype in coevolutionary processes. Clearly these relationships are dynamic but not in the way fitness landscapes are dynamic as discussed in the previous section.

It should be pointed out that much of the work on coevolutionary algorithms has been centered on the question of how to select evaluators and how to set the rules of interaction. A main difference in algorithmic design is drawn between algorithms that have just one population (single population coevolutionary algorithm) from which the evaluators are taken, and algorithms that have many populations (multi population coevolutionary algorithm) with possibly complex rules as to which population may supply evaluators for assigning the fitness of individuals of another population. This may include the case that some populations are only used to provide evaluators. Another difference frequently discussed is the exact nature of the interaction. The main distinctive feature that is used is if the interaction is cooperative or competitive [53, 64]. Cooperative means that the individual and the evaluators interact and collaborate to solve a problem that is harder or impossible to solve by each of them alone. The better they perform together the higher the reward and hence the fitness. This sometimes even means that the same fitness is given to all cooperating individuals. In competitive interaction the individual is rewarded for out–performing the evaluators, which sometimes means that the fitness of one individual is increased at the expense of the others. The terms *cooperative* and *competitive* are frequently used in computer science approaches to coevolution. In the biological and ecological literature, see for instance [3, 8], these terms are sometimes replaced by mutualistic coevolution and antagonistic coevolution.

Further issues occur in the case of the fitness of an individual being the subject of more than one interaction and hence consist of several partial finesses that have to be aggregated. Another topic is relevant if there are multiple populations and communication between them needs to be organized. The most common here is to have a centralized clock that defines the generational time for each population and also specifies at which points in time exchange of information and interaction takes place. We will assume this in the discussion to follow. All these questions are important for coevolutionary search and deeply influence the behavior and performance of such algorithms. On the other hand, they only touch on the main question of this chapter, namely how the fitness landscape describing coevolution relate to dynamics and creates codynamics.

10.3.2 Codynamic Fitness Landscapes

Basically, in coevolution there are as many fitness landscapes as there are populations. Since the focus here is on two coevolving populations, we have, in fact, two fitness landscapes. They might or might not have the same search space and neighborhood structure according to the definition set out in Equation (10.4). Let us assume they have (the case of unequal search spaces would not alter the following discussion substantially), but for distinguishing them we write S_x for the one, and S_y for the other. We now consider that the individuals of population $P_1(k)$ can take possible values $x \in S_x$ and the population $P_2(k)$ may have the values $y \in S_y$. So far we are still in line with the definitions of static (see Equation (10.4)) and dynamic (Equation (10.5)) fitness landscapes as given in Section 10.2.2. However, as a consequence of the coevolutionary allocation of fitness values discussed above, we now face the question of defining fitness values $f(x)$ and $f(y)$, which cannot be done by considering the search spaces separated from each other. This only becomes possible by considering the interaction between both populations. As mentioned before, the fitness here is no longer a constant property of a point in search space; fitness is generated on–the–fly by the process of coevolution. Hence, we might be able to formulate a recursive evolution law just as in Equation (10.11).

To illustrate this fact, it is instructive to combine both landscapes into one shared landscape $S = \{S_x, S_y\}$. This may result in a unique (static) landscape for simple coevolutionary scenarios and is particularly convenient if the coevolving populations are one–dimensional. Therefore, combining the two one–dimensional landscapes leads to a shared two–dimensional landscape. This approach has been followed in previous research on understanding coevolutionary phenomena by fitness landscapes [50, 51]. We use this example to describe the dynamics in coevolving fitness landscapes. Therefore, we consider the coevolution on simple functions, for instance ridge functions as suggested in [50, 51]. The simplest function has one ridge:

$$f(x,y) = n + 2\min(x,y) - \max(x,y) \tag{10.24}$$

with $x, y \in \mathbb{R}$ and n is a parameter that sets the size and the hight of the landscape (see Figure 10.3a). The landscape has a single maximum at $f(n,n) = 2n$ and a ridge diagonally from $f(0,0) = n$ to $f(n,n)$ that separates two planar surfaces. There are two minima at $f(0,n) = f(n,0) = 0$. Equation (10.24) is the fitness function for both populations P_1 and P_2 and can be interpreted as the static shared fitness landscape $S = \{S_x, S_y\}$ of the coevolution process.

The coevolutionary process we consider works as follows. Both populations evolve subsequently and alternatively along the conventional evolutionary algorithm's generational process (fitness evaluation followed by selection, recombination and possibly mutation); that is, population P_1 starts, and after it has finished its first generation, P_2 takes over, then P_1 starts again in the second generation and so on. The main difference to traditional evolutionary computation is how the fitness of either population is calculated, because the fitness evaluation in P_1 is subject to evaluators from P_2, and vice versa. As the populations take turns in evolving, this creates a coupling via the (time–dependent) fitness values (or some quantities derived from them) of the respective population. As an effect, both populations coevolve, and the landscapes show codynamics. Let us now look at how this dynamic behavior is reflected in their fitness landscapes.

We first need to define how the interaction between individuals of one population with evaluators from the other is organized, how the solution of the interaction is calculated, and how the solution translates to (personal or collective) fitness of the individuals. These questions can be addressed by employing the framework of *interactive domains* and *solution concepts* [49]. We here use a very simple version of this framework, as suggested in [51]. The interactive domain to calculate the fitness of population $P_2(k)$ consists of the individuals in P_2 interacting with population P_1 by picking the current best individual $x_{best}(k)$ of $P_1(k)$. The solution concept is as simple as just taking this value and inserting it in Equation (10.24). For P_1 it is the same but reverse, i.e. take the current best individual $y_{best}(k-1)$ and insert in Equation (10.24) to calculate the fitness of population $P_1(k)$. Because the populations take turns in coevolving with population P_1 starting and P_2 following, this calculation has to deal with P_2 lagging behind for one generation. We resolve this by taking a random $y = y_{rand}$ from population P_2 in the initial generation at $k = 1$; that is $y_{best}(0) := y_{rand}$. For $k = 2$, we take the value $y_{best}(1)$ and so on.

Interestingly, for this simple example it is still undefined if the given interaction is either cooperative or competitive. This only depends on the notion of what constitutes the best individual of either population, $x_{best}(k)$ and $y_{best}(k)$, respectively. As shown in [50, 51] a cooperative interaction is imposed if the task for both populations is the same, that is, both are to find the maximum or minimum of the fitness function (10.24). A competitive interaction takes place if one population is to search for the maximum of Equation (10.24), while the other is to find the minimum of Equation (10.24). Either way, we obtain the same codynamics expressed as a coupled fitness landscape. From the perspective of population P_1 the fitness landscape is dynamic by

$$f(x,k) = n + 2\min\left(x, y_{best}(k-1)\right) - \max\left(x, y_{best}(k-1)\right), \tag{10.25}$$

while from the perspective of population P_2 we get

$$f(y,k) = n + 2\min\left(x_{best}(k), y\right) - \max\left(x_{best}(k), y\right). \tag{10.26}$$

From either perspective alone it appears that fitness is calculated on–the–fly while the coevolutionary algorithm is running. Due to the simplicity of the example the codynamic fitness landscape can be depicted as a function of time. See Figure 10.3b as an example of cooperative interaction. This figure can be directly derived from the shared fitness landscape in Figure 10.3a by looking from the x–axis and considering the value for $y = y_{best}(k)$ as slices of the S_x space. For the first and the second generation the relatively low values of $y_{best}(k)$ lead to a comparably flat landscape to be searchable for population P_1. Hence, the maximum fitness that can be obtained is relatively low. The theoretically possible maximum cannot be reached at all; the dynamic fitness landscape simply does not include it for these generations. After a certain number of generations, the landscape curves up (due to better values of

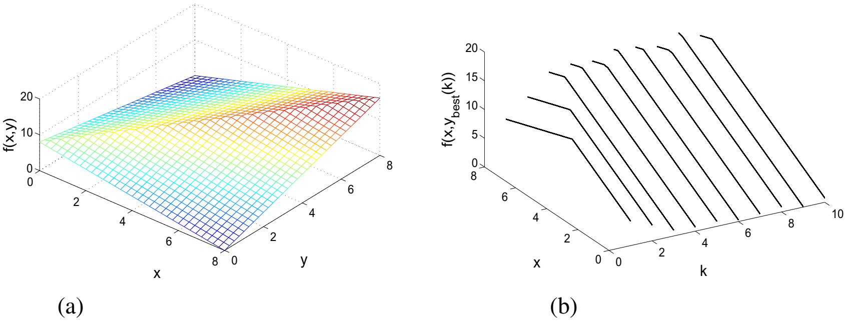

Fig. 10.3 (a) The oneRidge function of Equation (10.24) for $n = 8$. (b) The codynamic fitness landscape $f(x,k)$, see Equation (10.25), depending on k.

$y_{best}(k)$) and the overall maximum becomes finally accessible. It should be noted that this figure is an illustration, and not a verifiable numerical result. This is the landscape for one run, another run might produce a landscape that is similar but different in detail.

These results could have been obtained and presented in this way due to the extreme simplicity of the interaction and the solution concepts of the example. Only so, there is this unique relationship between the codynamic fitness landscapes of Equations (10.25) and (10.26) on the one hand, and the static shared landscape of Equation (10.24) on the other. Even if this simple example would be made more complicated the clear–cut relationship would cease. For instance, the interaction could use not only the current best, but past bests, or it could not only take the best, but a group of high fitness individuals including the best. The solution concept could entail a comparison or other operations of the fitness of picked individuals from P_1 and P_2. All these modifications would make the relationships between the codynamic landscape more complicated and finally question if the static shared landscape like the one given in Equation (10.24) can be uniquely decomposed into codynamic landscapes expressed as Equations (10.25) and (10.26). However, as long as the interaction produces a phenotypic point as a result and the solution concept gives it a fitness value that is unique and constant for the coevolutionary run, codynamic landscapes can be obtained from the overall landscape. Our conjecture is that if a fitness function such as Equation (10.24) is used as solution concept, the codynamic fitness landscapes are subspaces (for instance slices) of the fitness function that obtain their dynamics by being spanned every generation. Let us for instance again consider the example above and assume that the interaction is to pick a certain number of current best individuals from the other population, compare it with another number of best individuals from past generations, discard some individuals due to their low fitness and save the rest, and calculate as the solution of this interaction the weighted average of all individuals that were saved. Surely this is a complicated process, but it again produces a phenotypic point in the end. If the fitness is assigned by the fitness function in Equation (10.24), the situation is not altered significantly. No matter what the interaction yields, the result can finally be interpreted as an individual (or a group of individuals that can be conflated) and that can be inserted in a fitness function such as Equation (10.24). In some way, it could be argued that this kind of coevolution has an objective fitness after all. To generalize, for a shared fitness landscape $f(x,y)$ and the operator $interact(P_1,P_2,archive)$ describing the interactive domain (possibly supported by an archive), we obtain the codynamic fitness landscape

$$f(x,k) = f(x, interact\,(P_1(k),P_2(k),archive(k)) . \qquad (10.27)$$

For the variable y describing the coevolution of P_2, we obtain a similar mathematical description. Note that the codynamic landscape of Equation (10.27) implies a recursive evolution law as given in Equation (10.11) if we observe the evolutionary dynamics of the populations $P_1(k)$ and $P_2(k)$ describable by a generation transition function just as given by Equation (10.32). In line with this view and compatible with the external and internal drive in dynamic fitness landscapes considered in Section 10.2, codynamic fitness landscapes have a population–based dynamics.

A considerable number of coevolutionary settings fall into the category for which codynamic fitness landscapes can be models by Equation (10.27). The defining element is that a shared fitness landscape $f(x,y)$ serves as the solution concept. This applies to almost all coevolutionary algorithms used to solve optimization problems by decomposition, which have been called compositional problems [49]. Examples of compositional problems are [20, 22, 48, 50, 51]. For these examples, there are even first promising attempts to measure the resulting codynamic fitness landscapes and draw useful conclusions from it [29]. However, there is also an important group of coevolutionary problems that do not have a solution concept expressible as an equation–like formula known analytically beforehand and generally being knowable without doing numerical experiments. Consider for instance the case where in the course of evaluating an individual, there are multiple interactions with other individuals from the same but also from the other population that have multiple values attached that are aggregated to the individual's fitness. This is frequently the case for so–called test–based problems [11, 16]. For these problems there seems to be no static shared fitness landscape $f(x,y)$ mapping uniquely the set of all possible values x versus all possible values y and equipping every pair with a constant fitness value. To remain with the geographical metaphor of the fitness landscape, we no longer obtain a closed landscape, that is a surface (which may be rugged or smooth), but disjunct fitness islands. All we get is a phenotypic landscape but not a genotypic one. We conjecture that it may generally hold true that compositional problems have a shared fitness landscape, while test–based problems do not.

10.3.3 Examples of Modelling Coevolutionary Processes

A classical and well–known example of a fitness landscape that has the potential to be time–dependent, but also offers the ability to model coevolutionary couplings between different species, are the so–called NK or NKC landscapes introduced by

Kauffman, collaborating with Levin and Johnsen [31, 32, 34], where NK and NKC denote the tunable parameter of the landscape. These landscapes use a genotypic coding with a string of length N over a given alphabet, where in most of the implementations the alphabet of length 2, $\mathbb{A} = \{0,1\}$ is considered. The number N gives the number of components that are required to code for each point of the genotypic space. For the binary alphabet we hence have for each genotype a binary string $x = x_0 x_1 \ldots x_i \ldots x_{N-1}$, where x_i is the i–th bit. The parameter K in the NK fitness landscape describes the degree of epistatic interactions between the N components of each genotype. By tuning K between $0 \leq K \leq N-1$, different degrees of ruggedness (that relate to problem hardness) can be adjusted. The epistatic interaction is modelled by setting K neighbors for each bit x_i via a neighborhood function $n(x_i,K)$ and defining a fitness contribution $f_i(x_i,n(x_i,K))$ for each bit x_i and the K neighbors. In principle, there are two ways to set a neighborhood function $n(x_i,K)$. A first is nearest neighbor interaction, where $\frac{K}{2}$ bits on either side of x_i are considered neighbors. As $\frac{K}{2}$ must be an integer, this imposed a bias for odd K to the right or left hand side, and also requires a periodic boundary condition which says that the last and the first bit in the string are direct neighbors. A second design is random interaction, where the K neighbors are chosen at random (with no repetition or reciprocity) from among the remaining $N-1$ bits. The example considered below demonstrates nearest neighbor interaction with a right hand side bias. The fitness of each genotype x is defined by

$$f(x) = \frac{1}{N} \sum_{i=0}^{N-1} f_i(x_i,n(x_i,K)). \tag{10.28}$$

The contributions $f_i(x_i,n(x_i,K))$ are taken as realizations of a random variable uniformly distributed on the interval $[0,1]$ and depend on the interaction parameter K. As an example, consider the binary alphabet and $N = 4$, which gives $2^4 = 16$ genotypes in the landscape. If the Hamming distance H_D between genotypes next to each other is $H_D = 1$, a location is obtained from genotypic space and neighborhood structure, which results in a metric search space as illustrated in Figure 10.4. Tables 10.1 and 10.2 show the calculation of fitness for $x = 0110$.

In order to model coevolution and coupling between different species (and thereby obtaining a coupled fitness landscape), the NK landscape is modified. We next consider the coupling of two species. However, within the given framework the methodology can be straightforwardly extended to an arbitrary number of species. For two coevolving species, we need to define two genotypic sets, which (as an extension of the setting above) are described by two binary strings of length N, $x = x_0 x_1 \ldots x_i \ldots x_{N-1}$ and $y = y_0 y_1 \ldots y_i \ldots y_{N-1}$. Apart from the K epistatic interactions within each genotype (called internal interactions), there are additionally C epistatic interactions from one genotype to the other (and vice versa), which are called external interactions. This yields the name NKC landscape. As before, we can tune $0 \leq C \leq N-1$. For describing these two types of interactions we first need to set the K internal neighbors for each bit x_i and y_i via (usually identical) internal neighborhood functions $n(x_i,K)$ and $n(y_i,K)$. Additionally, C external neighbors in the other genotype need to be specified by an external

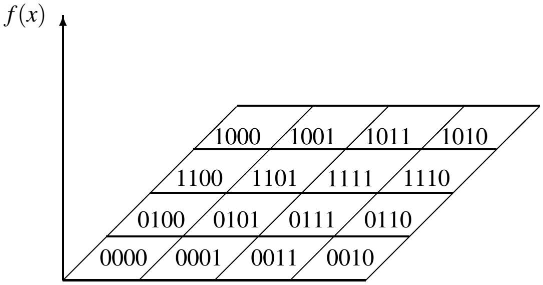

Fig. 10.4 Layout and distancing of the NK landscape defined by Equation (10.28) for $N = 4$ and neighborhood structure with Hamming distance $H_D = 1$

neighborhood function $n(x_i, y_i, C)$. This external neighborhood function is (usually) assumed to be symmetrical, $n(x_i, y_i, C) = n(y_i, x_i, C)$. The effect of internal and external interaction is combined by concatenating both neighborhood functions and results in KC neighborhood functions $n(x_i, K, C) = n(x_i, K)|n(x_i, y_i, C)$ and $n(y_i, K, C) = n(y_i, K)|n(y_i, x_i, C)$. Here $a|b$ means string a concatenated with string b. For these KC neighborhoods fitness contributions $f_i(x_i, n(x_i, K, C))$ and $f_i(y_i, n(y_i, K, C))$ are defined as independent realizations of a uniformly distributed random variable that superpose internal and external fitness contributions. Hence, the fitness of each genotype x and y is given by

$$f(x) = \frac{1}{N} \sum_{i=0}^{N-1} f_i(x_i, n(x_i, K, C)) \tag{10.29}$$

and

$$f(y) = \frac{1}{N} \sum_{i=0}^{N-1} f_i(y_i, n(y_i, K, C)). \tag{10.30}$$

Tables 10.3 and 10.4 give an example of calculating the fitness for $x = 0110$ and $y = 1001$.

So far, neither the NK nor the coupled NKC landscape depends on time. The fitness values do not change during the considered time frame, may that be a random walk on the landscape to calculate some landscape measures or an evolutionary run. With respect to the discussion of coevolutionary processes above, the NKC landscapes defined by the Equations (10.29) and (10.30) do not have timely interactions. Of course, they are coupled via shared fitness contributions f_i but not dynamically. The NK and NKC landscapes can be made dynamic in the same way as the landscapes considered before: by defining that certain landscape features change depending on an external source of dynamics. For the NK landscape this means that

Table 10.1 Example of the NK landscape with $N = 4$ and $0 \leq K \leq 3$ and $x = 0110$. The neighborhood function $n(x_i,K)$ is nearest neighbor interaction with right hand bias, namely $n(x_i,0) = x_i$, $n(x_i,1) = x_i x_{i+1}$, $n(x_i,2) = x_{i-1}x_i x_{i+1}$ and $n(x_i,3) = x_{i-1}x_i x_{i+1}x_{i+2}$ with the periodic boundary condition $x_N := x_0$. The fitness contributions $f_i(K) := f_i(x_i,n(x_i,K))$ are assigned as realizations of a random variable uniformly distributed on the interval $[0,1]$, refer to Table 10.2 as an example of a lookup table containing these values. We obtain the fitness function values $f(0110) = 0.4$ for $K = 0$, $f(0110) = 0.5$ for $K = 1$, $f(0110) = 0.475$ for $K = 2$, and $f(0110) = 0.55$ for $K = 3$.

	$K =$ 0					$K =$ 1				
x	0	1	1	0		0	1	1	0	
i	$n(x_i,0)$				$f_i(0)$	$n(x_i,1)$				$f_i(1)$
0	0				0.5	01				0.2
1		1			0.1		11			0.4
2			1		0.7			10		0.9
3				0	0.3				00	0.5

	$K =$ 2					$K =$ 3				
x	0	1	1	0		0	1	1	0	
i	$n(x_i,2)$				$f_i(2)$	$n(x_i,3)$				$f_i(3)$
0	001				0.1	0011				0.6
1		011			0.5		0110			0.3
2			110		0.6			1100		0.5
3				100	0.7				1001	0.8

Table 10.2 Example of the lookup table for the NK landscape with $N = 4$ and $K = 2$, which is of dimension $N \times 2^{K+1}$

$n(x_i,2)$	$f_0(2)$	$f_1(2)$	$f_2(2)$	$f_3(2)$
000	0.6	0.3	0.5	0.1
001	0.1	0.5	0.9	0.4
011	0.8	0.5	0.1	0.3
010	0.5	0.8	0.9	0.7
100	0.6	0.3	0.2	0.7
101	0.7	0.9	0.5	0.6
111	0.1	0.7	0.2	0.6
110	0.5	0.3	0.6	0.9

$$f(x,k) = \frac{1}{N}\sum_{i=0}^{N-1} f_i(x_i,n(x_i,K(k)),k). \qquad (10.31)$$

Dynamic NKC landscapes can be formulated by introducing time–dependent $K(k)$ and/or $C(k)$ in the models given by Equations (10.29) and (10.30). The landscape in Equation (10.31) also shows that there are two ways for imprinting dynamics on a NK landscape. A first is to change the parameter $K(k)$ (and/or $C(k)$); a second is to change the fitness contributions $f_i(k)$. In some ways the former implies the latter as a changed $K(k)$ and $C(k)$ entails a recasting of the lookup tables and hence changes the fitness contributions as well. The main difference between both variants is that only changing the $f_i(k)$ but retaining the K and C somehow restricts the severity of changes. See for instance [6, 76] for experiments with dynamic NK landscapes. However, it should also be pointed out that dynamic NKC landscapes formulated in a similar fashion as the dynamic NK landscape of Equation (10.31) do not directly model coevolutionary dynamics in the sense of the discussion in Section 10.3.2. Clearly, they are dynamically coupled (and could be interpreted as codynamics), but how the coupling modeled here relates to the coupling via interacting populations is far from being clear. An attempt to clarify such relations are the deformable landscape considered next.

Table 10.3 Example of the NKC landscape with $N = 4$, $K = 1$ and $1 \leq C \leq 2$ and $x = 0110$, $y = 1001$. The KC neighborhood functions $n(x_i,K,C)$, $n(y_i,K,C)$ are symmetric and nearest neighbor interaction, namely $n(x_i,1,1) = n(x_i,1)|n(x_i,y_i,1) = x_i x_{i+1}|y_i$, $n(x_i,1,2) = x_i x_{i+1}|y_i y_{i+1}$, where again the periodic boundary conditions $x_N := x_0$. $y_N := y_0$ are observed. Here $a|b$ means string a concatenated with string b. The fitness contributions $f_i^x(K,C) := f_i(x_i,n(x_i,K,C))$ and $f_i^y(K,C) := f_i(y_i,n(y_i,K,C))$ are assigned as independent realizations of a random variable uniformly distributed on the interval $[0,1]$, refer to Table 10.4 as an example of a lookup table containing these values. We obtain the fitness function values $f^x(0110) = 0.4$, $f^y(1001) = 0.375$ for $K = 1, C = 1$ and $f^x(0110) = 0.4$, $f^y(1001) = 0.5$ for $K = 1, C = 2$.

	K	$= C$	$=$								
	1	1									
x	0	1	1	0		y	1	0	0	1	
i	$n(x_i,1,1)$				$f_i^x(1,1)$		$n(y_i,1,1)$				$f_i^y(1,1)$
0	$01\vert 1$				0.3		$10\vert 0$				0.6
1		$11\vert 0$			0.2			$00\vert 1$			0.3
2			$10\vert 0$		0.7				$01\vert 1$		0.1
3				$00\vert 1$	0.4					$11\vert 0$	0.5

	K	$= C$	$=$								
	1	2									
x	0	1	1	0		y	1	0	0	1	
i	$n(x_i,1,2)$				$f_i^x(1,2)$		$n(y_i,1,2)$				$f_i^y(1,2)$
0	$01\vert 10$				0.1		$10\vert 01$				0.9
1		$11\vert 00$			0.5			$00\vert 11$			0.5
2			$10\vert 01$		0.3				$01\vert 10$		0.4
3				$00\vert 11$	0.7					$11\vert 00$	0.2

Up to now, we have considered examples where the dynamics of the fitness landscape is either externally defined and subsequently imprinted on the landscape, or is internal due to a mathematical description of the landscape that explicitly depends on time. A fundamentally different way to initiate dynamics in a fitness landscape is

Table 10.4 Example of the lookup table for the NKC landscape with $N = 4$, $K = 1$ and $C = 1$, which is of dimension $N \times 2^{K+C+1}$

$n(x_i,1,1)$	$f_0^x(1,1)$	$f_1^x(1,1)$	$f_2^x(1,1)$	$f_3^x(1,1)$	$n(y_i,1,1)$	$f_0^y(1,1)$	$f_1^y(1,1)$	$f_2^y(1,1)$	$f_3^y(1,1)$
00|0	0.6	0.3	0.5	0.1	00|0	0.6	0.3	0.5	0.1
01|0	0.1	0.5	0.9	0.4	01|0	0.1	0.5	0.9	0.4
11|0	0.8	0.2	0.1	0.3	11|0	0.8	0.5	0.1	0.5
10|0	0.5	0.8	0.7	0.7	10|0	0.6	0.8	0.9	0.7
10|1	0.6	0.3	0.2	0.7	10|1	0.6	0.3	0.2	0.7
11|1	0.7	0.9	0.5	0.6	11|1	0.7	0.9	0.5	0.6
01|1	0.3	0.7	0.2	0.6	01|1	0.1	0.7	0.1	0.6
00|1	0.5	0.3	0.6	0.4	00|1	0.5	0.3	0.6	0.9

by the individuals of a population that inhabits it. This can be done by utilizing the time dependence of the individuals' fitness and/or the individuals' phenotype (that is, a location in the search space), or by using quantities that are derived from fitness and/or location. In doing so, each individual is considered as to represent a phenotypic realization of a genotypic point in the landscape that changes with time. In other words, we use the evolutionary dynamics and let it interact with the environmental dynamics. Instead of individuals, also whole species that belong to a group of species can be considered as far as they are described by a single quantity averaging (or otherwise aggregating) the phenotypic variety of the species. Either way, for describing the evolutionary dynamics of the population (or the group of species) we need a time scale, which usually is discrete generational time. In the following we assume that there is a linear relationship between the generational time and the landscape time k and to simplify even more we set this linear relationship to equality. Nevertheless, the discussion that follows applies likewise for any unique relationship between generational time and landscape time. So, we can use the landscape time to measure generational dynamics. With this time scale we can formulate via a generation transition function ψ (see e.g. [2], p. 64–65) how a population $P(k+1)$ at generation $k+1$ originates from a population $P(k)$ at generation k:

$$P(k+1) = \psi\left(P(k)\right), k \geq 0. \tag{10.32}$$

These generational dynamics can now be linked to the fitness landscape and also be considered as to change the topology of the fitness landscape with time. In other words, the population adapts to the fitness landscape and deforms it.

There are some motivating thoughts for such a model. A first is that the individual (or the species) utilizes the abilities and features connected to the phenotypic realization while at the same time interacts and competes with other individuals of the same or other species. This in turn leads to an adaption process in the interacting parties, diminishes the species' fitness and bulges in the fitness landscape for the given phenotypic realization. As a result, fitness can only be maintained by moving in the phenotypic space, which is also called the Red Queen effect. This is a coevolutionary process where the actions of one species interact with the fitness of another species and vice versa. If both species have separated fitness landscapes they are coupled as for instance in the NKC model of the Equations (10.29) and (10.30). If we describe the actions of both species with a single shared fitness landscape, we obtain a dynamically deformable landscape. A very simple way to model such a deformation of the fitness landscapes by the individuals that populate it, is to take a static fitness landscape $f(x)$ and deduct fitness at points related to the position of the population. The most straightforward way to do so is to deform the landscape at the points where the population is in genotypic space. With λ the number of individuals in the population $P(k) = (p_1(k), p_2(k), \ldots, p_\lambda(k))$, we can define a dynamic fitness landscape that is dynamically deformed depending on the individuals' locations as

$$f(x,k) = f(x) - \frac{1}{\lambda} \sum_{i=1}^{\lambda} \alpha_i p_i(x,k), \tag{10.33}$$

where the α_i are some weighting factors. However, such a description has some serious disadvantages. It only deforms the landscape at exactly the points where the individuals are situated from generation to generation but not their vicinity. Also, the deforming process happens at exactly the moment when the individual arrives at the search space point and is not depleted at any later point in time. For these reasons it would be desirable to have some smoothening, both in the spatial and the temporal aspect. These ideas are addressed in the deformable fitness landscapes studied by Ebner and co–workers [17, 19], see also Chapter 12 of this book. These works consider fitness landscapes that are populated by several species, but this type of modelling could also be applied to individuals of a population. For the spatial smoothing, each individual $p_i(k)$ of the population $P(k)$ is surrounded by a negative Gaussian hill $-\exp\left(-(x-P(k))^T A(x-P(k))\right)$, that bulges in the landscape. The positive definite matrix A can be used to adjust the dilation of the spatial deforming. The temporal smoothing is also modeled by a Gaussian function, $\exp\left(\frac{i-k-\tau_{lat}}{2\sigma^2}\right)$, which runs over a temporal counter i and assigns that for the point in landscape time $(k+\tau_{lat})$ the spatial negative Gaussian hill deforms the landscape maximally and this effect phases out for smaller and larger times. The τ_{lat} is a latency to move the effect of the deformation forward in time and σ can be used to tune the timely deforming process. Hence, we have a time–dependent deformable fitness landscape based on a static landscape $f(x)$ as

$$f(x,k) = f(x) - \frac{1}{\alpha} \sum_{i=0}^{i_{end}} \exp\left(-\frac{(i-k-\tau_{lat})^2}{2\sigma^2}\right) \exp\left(-\frac{1}{2}(x-P(k))^T A(x-P(k))\right), \tag{10.34}$$

with $\alpha = \sigma(2\pi)^{\frac{n+1}{2}} \det(A)^{\frac{1}{2}}$ and i_{end} the final run time of the landscape. With Equation (10.34) we have a dynamic fitness landscape whose dynamics is population–based as $f(x,k+1)$ only depends on the values of the population $P(k+1)$ at this generation. This closely resembles the codynamic fitness landscape defined by Equation (10.27).

This population dynamics can follow any law as set out by Equation (10.32) and may include any genetic operator. A convenient way to generate population dynamics for experimenting with the deformable fitness landscape that is simple and computationally inexpensive is a hill climbing process, which was used in [17, 19]. In these works three types of hill climbing dynamics were considered that led to the following updating rules. All these updating rules use the landscape's gradient in order to model the effect that the population tries to escape regions where the landscape's fitness undergoes change. The first rule uses the sign of the landscape's gradient, i.e.

$$P(k+1) = \alpha\, sgn \left(\left. \frac{\partial f(x,k)}{\partial x} \right|_{x=P(k)} \right) \cdot P(k), \tag{10.35}$$

where *sgn* is the element–wise sign operator and α a weighting factor. A second rule sets the movement of the population proportional to the gradient, i.e.

$$P(k+1) = \alpha \left. \frac{\partial f(x,k)}{\partial x} \right|_{x=P(k)} \cdot P(k) \tag{10.36}$$

and a third rule integrates the gradient over time, i.e.

$$P(k+1) = \left(\sum_{i=1}^{k} \alpha_i \left. \frac{\partial f(x,i)}{\partial x} \right|_{x=P(i)} \right) \cdot P(k). \tag{10.37}$$

Detailed experiments with such deformable landscapes are given in Chapter 12 of this book.

10.4 Conclusions

10.4.1 Hierarchy of Fitness Landscapes

In this chapter we have dealt with landscapes whose fitness values change with time. It was shown that such time–dependent landscapes may occur in describing two major evolutionary processes. A first is in treating conditions apart from the genetic makeup of a population that change with time and massively influence the evolutionary outcome. Most prominently, this means that fitness of a phenotypic realization is not constant over the time frame of consideration and creates a dynamic environment in which the evolutionary process takes place. This leads to dynamic fitness landscapes and may imply solving a dynamic optimization problem. A second evolutionary process deals with situations where several populations (or at least several individuals) interact in a cooperative or competitive way and hence mutually influence fitness in a timely fashion. This leads to coevolution and creates fitness landscapes for each population that are codynamic. These landscapes are coupled via dynamic fitness evaluation and repercussions from the interdependent fitness allocation can be seen as to deform the landscapes.

In the last two sections, we have studied such landscapes and have seen that the dynamics may have three sources: internal, external and population–based. The distinction between external, internal and population–based dynamics has, apart from characterizing the source of changes in the fitness landscape, implications for the specification of the dynamics of neighboring points in the search space. For internal dynamics, the landscape's time evolution law of Equation (10.11) states how the fitness of all points in the search space and all their neighboring points change with time. In other words, the changes are explained for every search space point. With external dynamics, we only define changes of selected and characteristic features in the landscape. Here, the changes are explicitly explained for only a discrete subset of search space points. Their neighborhood may change too, but according to the same time regime as the points themselves. Population–based dynamics extends to landscapes for each population involved that mutually influence each other in a codynamic way. This can be interpreted as features of one landscape that drive other landscapes externally. However, there is no separation between that what drives and that what is driven because the process alternates between all involved landscapes and the drive is not directed to selected and pre–defined topological features of the landscapes.

To put these facts into a wider context, there has been an attempt to draw a connection to spatially extended systems and to establish a hierarchy of fitness landscapes [59], which is based on a hierarchy of spatio–temporal dynamics [14, 26] (see Table 10.5). The hierarchy comes from the different combinations of discretizing space and time in the fitness landscape. For the landscape being static, the search space can be either continuous or discrete (binary), which results in problem classes 1 and 2. If the discrete search space consists additionally of a finite number of elements, the corresponding optimization problem is a combinatorial one. Class 3 are discrete fitness functions with discrete dynamics. These dynamics can be external, as for instance the XOR generator [78], see Equation (10.17), or dynamic combinatorial optimization problems such a dynamic knapsack, dynamic royal road or dynamic bit–matching [70]. An example of internal dynamics are fitness landscapes constructed from coupled map lattices (CML). A class 3 problem with population–based drive would be a combinatorial optimization problem solved by a coevolutionary algorithm. Continuous fitness functions with discrete dynamics form class 4. Examples with external dynamics include the moving peak benchmark defined in Equation (10.12), but also other similar problems such a dynamic sphere,

Table 10.5 Hierarchy of fitness landscapes; S: static, D: discrete, C: continuous; Dynamics: N: no dynamics, I: internal, E: external, P: population–based

Class	Space	Time	Possible model	Dynamics
1	*D*	S	Discrete fitness function	N
2	*C*	S	Continuous fitness function	N
3	*D*	*D*	Discrete fitness function with discrete dynamics	I, E, P
4	*C*	*D*	Continuous fitness function with discrete dynamics	E, P
5	*D*	*C*	Lattice of coupled ordinary differential equation (ODE)	I
6	*C*	*C*	Partial differential equation (PDE)	I

dynamic Ackley, dynamic Rosenbrook etc. The coevolutionary problem considered in Section 10.3.2 as well as decomposition problems with a continuous evaluation function (see e.g. [79]) fall into class 4 with population–based dynamics. To define internal dynamics for class 4 problems would mean to have a description as in Equation (10.11) for a non–countable number of points in the fitness landscape, which is impractical for any numerical calculation. Therefore, class 4 problems with internal dynamics do not play a role in the literature. Class 5 and 6 fitness landscapes have continuous time and discrete or continuous search spaces. Possible models for such dynamic fitness landscapes are lattices of ODEs or PDEs. The mathematical formulation involves internal continuous–time dynamics. Such models are suitable for formulating general fitness landscapes and mainly pervade modeling and studying fundamental properties of evolution, but rarely to be found in studies in evolutionary computation. The reason for this might be that, as mentioned before, the discrete population dynamics of an evolutionary algorithm is best linked to fitness landscapes with discrete time characteristics. In addition, both types of mathematical description do, at least not in general, have an analytic solution. Any numerical calculation involves a discretization of time and/or space, for instance in numerical integrating ODEs and PDEs. Hence, both classes can, at least from a numerical point of view, be reduced to class 3 problems. As coevolutionary algorithm drive the codynamic landscapes by interacting populations, the resulting population–based dynamics is always discrete–time. However, codynamics also mean that the landscapes for each population are dynamical slices through an overall landscape. As a consequence, the shared static fitness landscape might, for instance, be of class 2, but the codynamic landscapes for each population are class 4.

10.4.2 Future Research Directions

Research on fitness landscapes has intensified recently and those advances are the topic of this book. Research on fitness landscapes that depend on time is an even younger field and still a considerable step away from maturity. Therefore, it comes as no surprise that some interesting questions remain unanswered yet. However, this is also connected with the hope that the present study might serve as a baseline for further development. Two directions appear to be particularly interesting. A first is linked to one of the major purposes of fitness landscapes: to provide a notion of how difficult it is for an evolutionary search algorithm to solve a given optimization problem. One way to accrue such knowledge is by landscape measures. For static measures theoretical and practical considerations have let to reliable and applicable results, see e.g. [24, 67] but also Chapters 4, 5 and 8 of this book. There are some examples of works on measures for landscapes that depend on time [21, 29, 58, 59]. However, their main focus is on applying measures of static landscapes (that is, measures for topological features) to the dynamic situation. This somehow masks the effect the dynamics has on the landscape and hence on problem hardness. It would be desirable to establish additional measures for dynamic problem hardness. A second direction is studying landscapes of specific problems and by doing so closing the gap between theory and application. Again, this is a path that has been successfully followed for static landscapes [38, 54, 71]. Hence, this approach seems to be also promising for dynamic and codynamic landscapes.

References

[1] Alba, E., Sarasola, B.: Measuring fitness degeneration in dynamic optimization problems. In: Di Chio, C., et al. (eds.) Applications of Evolutionary Computation - EvoApplications 2010, pp. 572–581. Springer, Berlin (2010)

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

[3] Bascompte, J., Jordano, P.: Plant-animal mutualistic networks: the architecture of biodiversity. Annu. Rev. Ecol. Evol. Syst. 38, 567–593 (2007)

[4] Bosman, P.A.N., La Poutré, H.: Computationally intelligent online dynamic vehicle routing by explicit load prediction in an evolutionary algorithm. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 312–321. Springer, Heidelberg (2006)

[5] Bosman, P.A.N.: Learning and anticipation in online dynamic optimization. In: Yang, S., Ong, Y.S., Jin, Y. (eds.) Evolutionary Computation in Dynamic and Uncertain Environments, pp. 129–152. Springer, Heidelberg (2007)

[6] Brabazon, A., Silva, A., de Sousa, T.F., O'Neill, M., Matthews, R., Costa, E.: A particle swarm model of organizational adaptation. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 12–23. Springer, Heidelberg (2004)

[7] Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 1999, pp. 1875–1882. IEEE Press, Piscataway (1999)

[8] Buckling, A., Rainey, P.B.: Antagonistic coevolution between a bacterium and a bacteriophage. Proc. R. Soc. Lond. B 269, 931–936 (2002)

[9] Bull, L.: Coevolutionary species adaptation genetic algorithms: a continuing SAGA on coupled fitness landscapes. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 322–331. Springer, Heidelberg (2005)

[10] Chazottes, J.R., Fernandez, B.: Dynamics of Coupled Map Lattices and of Related Spatially Extended Systems. Springer, Heidelberg (2005)

[11] Chellapilla, K., Fogel, D.B.: Evolving neural networks to play checkers without relying on expert knowledge. IEEE Trans. Neural Netw. 10, 1382–1391 (1999)

[12] Cheng, H., Yang, S.: Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 562–571. Springer, Heidelberg (2010)

[13] Cheng, H., Yang, S.: Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks. Engineering Applications of Artificial Intelligence 23, 806–819 (2010)

[14] Crutchfield, J.P., Kaneko, K.: Phenomenology of spatiotemporal chaos. In: Hao, B. (ed.) Directions in Chaos, vol. 1, pp. 272–353. World Scientific, Singapore (1987)

[15] Cruz, C., Gonzlez, J.R., Pelta, D.A.: Optimization in dynamic environments: a survey on problems, methods and measures. Soft Computing 15, 1427–1448 (2011)

[16] de Jong, E.D., Polack, J.B.: Ideal evaluation from coevolution. Evolutionary Computation 12, 159–192 (2004)

[17] Ebner, M., Watson, R.A., Alexander, J.: Co–evolutionary dynamics on a deformable landscape. In: Zalzala, A., Fonseca, C., Kim, J.H., Smith, A., Yao, X. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2000, pp. 1284–1291. IEEE Press, Piscataway (2000)

[18] Ebner, M.: Coevolution and the red queen effect shape virtual plants. Genetic Programming and Evolvable Machines 7, 103–123 (2006)

[19] Ebner, M., Watson, R.A., Alexander, J.: Coevolutionary dynamics of interacting species. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 1–10. Springer, Heidelberg (2010)

[20] Goh, C.K., Tan, K.C.: A competitive–cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Trans. Evolut. Comp. 13, 103–127 (2009)

[21] Hordijk, W., Kauffman, S.A.: Correlation analysis of coupled fitness landscapes. Complexity 10, 42–49 (2005)

[22] Jansen, T., Wiegand, R.P.: The cooperative coevolutionary (1+1) EA. Evolutionary Computation 12, 405–434 (2004)

[23] Jones, T.: Evolutionary algorithms, fitness landscape and search. PhD thesis, The University of New Mexico, Albuquerque (1995), `http://www.santafe.edu/media/workingpapers/95-05-048.pdf` (retrieved November 11, 2012)

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

[25] Kaneko, K.: The coupled map lattice. In: Kaneko, K. (ed.) Theory and Application of Coupled Map Lattices, pp. 1–49. John Wiley, Chichester (1993)

[26] Kaneko, K., Tsuda, I.: Complex Systems: Chaos and Beyond. Springer, Heidelberg (2001)

[27] Kardar, M., Parisi, G., Zhang, Y.C.: Dynamic scaling of growing interfaces. Phys. Rev. Lett. 56, 889–892 (1986)

[28] Kashtan, N., Noor, E., Alon, U.: Varying environments can speed up evolution. Proc. Natl. Acad Sci USA (PNAS) 104, 13711–13716 (2007)

[29] Katada, Y., Handa, Y.: Tracking the Red Queen effect by estimating features of competitive co–evolutionary fitness landscapes. In: Fogel, D.B. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2010, pp. 4417–4424. IEEE Press, Piscataway (2010)

[30] Katzav, E., Cugliandolo, L.F.: From coupled map lattices to the stochastic Kardar–Parisi–Zhang equation. Physica A371, 96–99 (2006)

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

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

[33] Kauffman, S.A., Weinberger, E.D.: The NK Model of rugged fitness landscapes and its application to the maturation of the immune response. J. Theor. Biol. 141, 211–245 (1989)

[34] Kauffman, S.A., Johnsen, S.: Coevolution to the edge of chaos: Coupled fitness landscapes, poised states, and coevolutionary avalanches. J. Theor. Biol. 149, 467–505 (1991)

[35] Lin, S.C., Goodman, E.D., Punch, W.F.: A genetic algorithm approach to dynamic job shop scheduling problems. In: Bäck, T. (ed.) Proc. Seventh International Conference on Genetic Algorithms, pp. 481–488. Morgan Kaufmann, San Francisco (1997)

[36] Ma, K., Jianga, J., Yanga, C.B.: Scaling behavior of roughness in the two–dimensional Kardar–Parisi–Zhang growth. Physica 378, 194–200 (2007)

[37] Mendes, R., Mohais, A.: DynDE: Differential Evolution for dynamic optimization problems. In: Corne, D. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2005, pp. 2808–2815. IEEE Press, Piscataway (2005)

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

[39] Meyers, L.A., Bull, J.J.: Fighting change with change: adaptive variation in an uncertain world. Trends in Ecology & Evolution 17, 551–557 (2002)

[40] Miranda, V.G., Aarão Reis, F.D.A.: Numerical study of the Kardar–Parisi–Zhang equation. Phys. Rev. 77, 031134–1–6 (2008)

[41] Morrison, R.W., De Jong, K.A.: A test problem generator for non–stationary environments. In: Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 1999, pp. 2047–2053. IEEE Press, Piscataway (1999)

[42] Morrison, R.W., De Jong, K.A.: Triggered hypermutation revisited. In: Zalzala, A., et al. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2000, pp. 1025–1032. IEEE Press, Piscataway (2000)

[43] Nguyen, T.T., Yao, X.: Benchmarking and solving dynamic constrained problems. In: Tyrrell, A. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2009, pp. 690–697. IEEE Press, Piscataway (2009)

[44] Nguyen, T.T., Yang, S., Branke, J.: Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation 6, 1–24 (2012)

[45] Oliehoek, F.A., de Jong, E.D., Vlassis, N.A.: The parallel Nash memory for asymmetric games. In: Cattolico, M. (ed.) Proc. Genetic and Evolutionary Computation Conference, GECCO 2006, pp. 337–344. ACM Press, New York (2006)

[46] Panait, L., Luke, S.: Time–dependent collaboration schemes for cooperative coevolutionary algorithms. In: Potter, M.A., Wiegand, R.P. (eds.) 2005 AAAI Fall Symposium on Coevolutionary and Coevolving Systems. AAAI Press, Palo Alto (2005)

[47] Panait, L., Luke, S.: Selecting informative actions improves cooperative multiagent learning. In: Nakashima, H., Wellman, M.P., Weiss, G., Stone, P. (eds.) Proc. Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2006), pp. 760–766. ACM Press, New York (2006)

[48] Panait, L., Luke, S., Harrison, J.: Archive–based cooperative coevolutionary algorithms. In: Cattolico, M. (ed.) Proc. Genetic and Evolutionary Computation Conference, GECCO 2006, pp. 345–352. ACM Press, New York (2006)

[49] Popovici, E., Bucci, A., Wiegand, R.P., de Jong, E.D.: Coevolutionary principles. In: Rozenberg, G., Bäck, T., Kok, J.N. (eds.) Handbook of Natural Computing, pp. 987–1033. Springer, Heidelberg (2010)

[50] Popovici, E., de Jong, K.A.: Understanding competitive co–evolutionary dynamics via fitness landscapes. In: Luke, S. (ed.) 2004 AAAI Fall Symposium on Artificial Multiagent Learning. AAAI Press, Palo Alto (2005)

[51] Popovici, E., de Jong, K.A.: Understanding cooperative co–evolutionary dynamics via simple fitness landscapes. In: Beyer, H.G., O'Reilly, U.M. (eds.) Proc. Genetic and Evolutionary Computation Conference, GECCO 2005, pp. 507–514. Morgan Kaufmann, San Francisco (2005)

[52] Popovici, E., de Jong, K.A.: The dynamics of the best individuals in co–evolution. Natural Computing 5, 229–255 (2006)

[53] Potter, M.A., de Jong, K.A.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation 8, 1–29 (2000)

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

[55] Richter, H.: Behavior of evolutionary algorithms in chaotically changing fitness landscapes. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 111–120. Springer, Heidelberg (2004)

[56] Richter, H.: A study of dynamic severity in chaotic fitness landscapes. In: Corne, D. (ed.) Proc. Congress on Evolutionary Computation, IEEE CEC 2005, pp. 2824–2831. IEEE Press, Piscataway (2005)

[57] Richter, H.: Evolutionary optimization in spatio–temporal fitness landscapes. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 1–10. Springer, Heidelberg (2006)

[58] Richter, H.: Coupled map lattices as spatio–temporal fitness functions: Landscape measures and evolutionary optimization. Physica 237, 167–186 (2008)

[59] Richter, H.: Evolutionary optimization and dynamic fitness landscapes: From reaction-diffusion systems to chaotic CML. In: Zelinka, I., Celikovsky, S., Richter, H., Chen, G. (eds.) Evolutionary Algorithms and Chaotic Systems. SCI, vol. 267, pp. 409–446. Springer, Heidelberg (2010)

[60] Richter, H.: Memory design for constrained dynamic optimization problems. In: Di Chio, C., et al. (eds.) EvoApplicatons 2010, Part I. LNCS, vol. 6024, pp. 552–561. Springer, Heidelberg (2010)

[61] Richter, H., Dietel, F.: Solving dynamic constrained optimization problems with asynchronous change pattern. In: Di Chio, C., et al. (eds.) EvoApplications 2011, Part I. LNCS, vol. 6624, pp. 334–343. Springer, Heidelberg (2011)

[62] Richter, H., Yang, S.: Memory based on abstraction for dynamic fitness functions. In: Giacobini, M., et al. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 596–605. Springer, Heidelberg (2008)

[63] Richter, H., Yang, S.: Learning behavior in abstract memory schemes for dynamic optimization problems. Soft Computing 13, 1163–1173 (2009)

[64] Rosin, C.D., Belew, R.K.: New methods for competitive coevolution. Evolutionary Computation 5, 1–29 (1997)

[65] Simões, A., Costa, E.: Variable–size memory evolutionary algorithm to deal with dynamic environments. In: Giacobini, M., et al. (eds.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 617–626. Springer, Heidelberg (2007)

[66] Simões, A., Costa, E.: Evolutionary algorithms for dynamic environments: Prediction using linear regression and Markov chains. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN X 2008. LNCS, vol. 5199, pp. 306–315. Springer, Heidelberg (2008)

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

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

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

[70] Stanhope, S.A., Daida, J.M.: (1+1) Genetic algorithm fitness dynamics in a changing environment. In: Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 1999, pp. 1851–1858. IEEE Press, Piscataway (1999)

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

[72] Tinós, R., Yang, S.: A self–organizing random immigrants genetic algorithm for dynamic optimization problems. Genetic Programming and Evolvable Machines 8, 255–286 (2007)

[73] van Hemert, J., La Poutré, J.A.H.: Dynamic routing problems with fruitful regions: Models and evolutionary computation. In: Yao, X., et al. (eds.) PPSN VIII 2004. LNCS, vol. 3242, pp. 692–701. Springer, Heidelberg (2004)

[74] Watson, R.A., Pollack, J.B.: Coevolutionary dynamics in a minimal substrate. In: Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E. (eds.) Proc. Genetic and Evolutionary Computation Conference, GECCO 2001, pp. 702–709. Morgan Kaufmann, San Francisco (2001)

[75] Weicker, K.: Performance measures for dynamic environments. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN VII 2002. LNCS, vol. 2439, pp. 64–73. Springer, Heidelberg (2002)

[76] Wilke, C.O., Martinetz, T.: Adaptive walks on time-dependent fitness landscapes. Phys. Rev. E60, 2154–2159 (1999)

[77] Yang, S.: Non–stationary problem optimization using the primal-dual genetic algorithm. In: Sarker, R., Reynolds, R., Abbass, H., Tan, K.C., Essam, D., McKay, R., Gedeon, T. (eds.) Proc. Congress on Evolutionary Computation, IEEE CEC 2003, pp. 2246–2253. IEEE Press, Piscataway (2003)

[78] Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft Computing 9, 815–834 (2005)

[79] Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Information Science 178, 2985–2999 (2008)