A Measure of Landscapes

Wim Hordijk
Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM 87501
wim@santafe.edu

Abstract

This paper introduces a statistical fitness landscape analysis, based on Weinberger's random walk method and on a time series analysis known as the Box-Jenkins approach, to measure and express the correlation structure of fitness landscapes. The analysis has some additions to and advantages over previous methods for measuring this structure. The analysis is demonstrated on fitness landscapes constructed with Kauffman's NK-model, using two operators (point mutation and a form of crossover) and a combination of the two. Furthermore, the predictive value of the method is shown.

Keywords:

Fitness landscapes, correlation structure, correlation length.

1 Introduction

> *"We need a real theory relating the structure of rugged multi-peaked fitness landscapes to the flow of a population upon those landscapes.*
>
> *We do not yet have such a theory."*

This quote, from Stuart A. Kauffman (Kauffman, 1993), states the goal of this paper. Its purpose is to contribute another brick in the wall of this (badly needed) theory. The part of the theory this paper is focusing on is the structure of fitness landscapes.

There are several characteristics that make up the structure of a fitness landscape. For example, one could look at the distribution of local optima, their heights, the number of steps it takes to reach them, etc. Another important feature of fitness landscapes is the average *fitness difference* between points in the landscape. In other words, how much will the fitness change when a move is made from one point to another point in the landscape. This characteristic of a landscape is often expressed by the *correlation structure* of the landscape, which indicates the (average) amount of correlation between the fitness values of two points a certain distance apart in the landscape.

So far, different methods have been used to measure the correlation structure of a fitness landscape (Manderick et al., 1991; Lipsitch, 1991; Weinberger, 1990), but they basically all come up with one single number: the characteristic (auto)correlation of the landscape, or a correlation length derived from it. However, during a walk (or search) through the landscape, it could be very useful to have a way of *predicting* the fitness of a (next) point in the landscape given the fitness values of some previously visited points. For example, it turns out that during a search on a landscape just going uphill as fast as possible (i.e., steepest ascent) does not always lead to the best results (see for example (Forrest and Mitchell, 1993; Wilson, 1991)). Having a model that is able to predict the expected (average) increase in fitness can help in selecting a next point during the search, one that has a fitness increase closer to that predicted by the model, thus taking advantage of the structure of the landscape.

This paper introduces a statistical fitness landscape analysis, based on the random walk method used in (Weinberger, 1990), that provides a complete stochastic model that can be used as both a characterization of the correlation structure of the landscape, and as a predictor for fitness values of distant points in the landscape. The analysis is also based on a time series analysis known as the Box-Jenkins approach (Box and Jenkins, 1970). Here, the analysis is demonstrated on fitness landscapes constructed with the NK-model (Kauffman, 1989). In another paper (Hordijk, 1996) the analysis is also applied to another, less "well behaved" fitness landscape to demonstrate the generality of the method.

In the next section the concept and usefulness of fitness landscapes is explained in more detail. The section following that explains the NK-model as introduced by Kauffman (Kauffman, 1989). In Section 4, different statistical concepts and definitions are explained, which are then put together to come up with the complete statistical fitness landscape analysis. In Section 5 this landscape analysis is applied to different fitness landscapes constructed with the NK-model. Section 6 shows one particular example of how the models that result from the analysis can be used to make quantitative predictions about an actual search on the landscape. Finally, in Section 7, the major conclusions that can be drawn from the application of the landscape analysis are summarized.

2 Fitness Landscapes

The notion of a fitness landscape comes from biology (first introduced in (Wright, 1932)), where it is used as a framework for thinking about evolution. Biological organisms can be viewed by their *genotype*, which is the genetic "encoding" of the organism, or their *phenotype*, which is the actual form and behavior of the organism. An abstract notion of *fitness* can be assigned to each phenotype, which denotes its ability to survive and reproduce. Evolution is now viewed as a process that searches, by means of genetic operators like crossover and mutation, a fitness landscape of possible genotypes, looking for genotypes that encode highly fit phenotypes. In other words, evolution searches for "solutions", encoded in genotypes, for the problem of finding fit organisms that are capable of reproduction.

This landscape paradigm can be used for search in general. Given an optimization problem, the set of possible solutions to this problem can be encoded using words of a (usually) fixed length formed from some finite size alphabet. This encoding generates a *representation space*, which is a high dimensional space of all possible words of a particular length, each of which encodes a possible solution to the problem (in analogy with genotypes and phenotypes). Furthermore, there is a *neighborhood relation* that defines which points in the representation space are connected. This relation usually depends on the specific search operator (for example mutation) — or combination of search operators, for example crossover and mutation — that is used to search the space. In other words, points that are reachable from each other by one application of the search operator are connected. Finally, there is a fitness function that assigns a fitness value to each possible word, or point in the landscape. This fitness value can be seen as a measure of how good a solution, represented by that point in the landscape, is to the given problem.

Summarizing, a fitness landscape is defined by three things:

1. A representation space (all possible words in the encoding).
2. A neighborhood relation denoting which points in the representation space are neighbors.
3. A fitness function that assigns a fitness value to each point in the space.

Together, these three "ingredients" give rise to the visual image of a more or less mountainous landscape (consider the fitness of each point as its "height"), with valleys, ridges and peaks. Optimization, i.e. finding good solutions to a problem, is now a search on this landscape, looking for the highest peaks, which denote the best (the most fit) solutions. This makes the fitness landscape metaphor a useful one, because it gives rise to intuitively simple images that can help in understanding evolutionary processes or search processes in general. Furthermore, the above definition of a fitness landscape lends itself to rigorous mathematical analysis, recent attempts of which can be found in (Jones, 1995; Stadler, 1995).

Different landscapes differ in their *ruggedness*. A landscape with a few local peaks and small (average) fitness differences between neighboring points is called *smooth*. A landscape with a lot of local peaks and large fitness differences is called *rugged*. The more rugged a landscape is, the harder (in general) it will be to search on it for "good" solutions (high peaks), and the less information there will be about the fitness of distant points in the landscape. This provides another useful feature of fitness landscapes and measures of their structure: the structure of a landscape can reflect how easy or difficult it is for a search algorithm to find good solutions (Manderick et al., 1991; Mitchell et al., 1992).

3 The NK-Model

The *NK-model* was introduced by Kauffman to have a problem-independent model for constructing fitness landscapes that can gradually be tuned from smooth to rugged (Kauffman, 1989). The main parameters of the model are N, the number of genes in the genotype, i.e. the length of the strings that form the points in the landscape, and K, the number of other genes that epistatically influence a particular gene (i.e., the fitness contribution of each gene is determined by the gene itself plus K other genes). Kauffman showed that almost all properties of the model are independent of A, the number of possible values each gene can have, so here the simplest case of $A = 2$ (i.e., bit strings) will be used.

The fitness of a bit string b of length N is defined as follows. Suppose every bit b_i ($i = 1, \ldots, N$) in a bit string b is assigned a fitness f_i of its own. The fitness f_i of a bit b_i, however, does not only depend on the value (0 or 1) of the bit itself, but also on the value of K other bits b_j in the same bit string ($0 \leq K \leq N - 1$). These dependencies are called *epistatic interactions*. So the fitness contribution of one bit depends on the value of $K + 1$ bits (itself and K others), giving rise to a total of 2^{K+1} possibilities, called neighborhood configurations. Each of these neighborhood configurations is assigned a random fitness value, drawn from a uniform probability distribution between 0.0 and 1.0. Therefore, the fitness contribution f_i of bit b_i is specified by a list of 2^{K+1} random decimals between 0.0 and 1.0. This assignment of fitness values is repeated for every bit b_i, $i = 1, \ldots, N$ in the bit string b. So there are N lookup tables (one for each bit), each with 2^{K+1} entries.

Having thus assigned the fitness contributions for every bit in the string, the fitness of the entire bit string is now defined as the average of the fitness contributions of all the bits:

$$F = \frac{1}{N} \sum_{i=1}^{N} f_i$$

Table 1 gives an example, taken from (Kauffman, 1993), of a lookup table for N=3 and K=2. The fitness contributions for each bit and for each possible neighborhood configuration are assigned at random, and the total fitness of the string is the average of the fitness contributions of all the bits.

value of bit			fitness contribution			total fitness
1	2	3	w_1	w_2	w_3	$W = \frac{1}{N} \sum_{i=1}^{N} w_i$
0	0	0	0.6	0.3	0.5	0.47
0	0	1	0.1	0.5	0.9	0.50
0	1	0	0.4	0.8	0.1	0.43
0	1	1	0.3	0.5	0.8	0.53
1	0	0	0.9	0.9	0.7	0.83
1	0	1	0.7	0.2	0.3	0.40
1	1	0	0.6	0.7	0.6	0.63
1	1	1	0.7	0.9	0.5	0.70

Table 1: An example of a lookup table for N=3 and K=2. The fitness contributions for the three bits for each of the $2^{K+1} = 8$ neighborhood configurations are assigned at random. The fitness of the entire string is the average of the fitness contributions of the three bits.

One further aspect of the NK-model characterizes how the K epistatic interactions for each bit are

chosen. Generally, this is done in one of two ways. The first way is by choosing them at random (without repetition) from among the other $N-1$ bits. This is called *random interactions*. It is important to note that no reciprocity in epistatic influence is assumed. This means that if the fitness of bit b_i (partly) depends on bit b_j, it is not necessary that the reverse also holds. The epistatic interactions for one bit are determined independent of the epistatic interaction of other bits, but once they are chosen, they remain fixed.

The second way of choosing the epistatic interactions for a bit is by taking the K neighboring bits. The $K/2$ bits on each side of a bit will influence the fitness of this bit. This is called *nearest neighbor interactions*. To make this possible *periodic boundary conditions* are taken into account. This means that the bit string is considered circular, so the first and the last bit are each others neighbors. Note that for K=0 and $K = N-1$, there is no difference between the two types of interactions. In the first case, the fitness of each bit depends only on its own value, and in the second case, the fitness of each bit depends on the value of all the bits in the string.

Landscapes resulting from the NK-model were originally constructed using point mutation as defining the neighborhood relation (i.e., bit strings that differ in the value of only one bit are neighbors). By increasing the value of K from 0 to $N-1$, these fitness landscapes can be tuned from smooth to rugged. When K is small, neighboring bit strings will have a relatively small difference in fitness, because the bits that are different in the two bit strings influence the fitness contribution of only a few bits in both strings. So only a few bits in both strings will have a different fitness contribution, which will have a small influence on the overall fitness of both strings.

When K is large, however, the differing bits of two neighboring strings will influence the fitness of a large number of bits in both strings, which will have a large influence on the overall fitness of both strings. When $K = N-1$, the fitness landscape will be completely random, because changing the value of only one bit changes the fitness contribution of all bits in the string, so the overall fitness will be completely different (however, the larger N is, the more the fitness values will cluster around 0.5).

As the results in later sections will show, this increase in ruggedness for increasing K relative to N, will also occur when other neighborhood relations, such as those induced by a crossover operator, are used. Therefore, the NK-model will be used here to construct fitness landscapes of different ruggedness. The statistical fitness landscape analysis, as introduced in the next section, will then be applied to these different landscapes.

4 The Statistical Fitness Landscape Analysis

The landscape analysis introduced here is partly based on Weinberger's method (Weinberger, 1990) using the autocorrelation function, and partly on a time series analysis known as the Box-Jenkins approach (Box and Jenkins, 1970). These separate issues are first addressed in several subsections, after which they will be combined into the complete statistical fitness landscape analysis.

4.1 The autocorrelation function

Weinberger (Weinberger, 1990) introduced a method to measure the correlation structure of fitness landscapes based on the *autocorrelation function.* The idea is to first perform a random walk on the landscape via neighboring points. The landscape Weinberger used is based on point-mutation, i.e., neighboring points are points in the landscape that differ in exactly one bit (in case of bit strings). Starting at a randomly chosen point, at every step one randomly chosen bit in the string is flipped. At each step the fitness of the genotype encountered is recorded. This way, a time series of fitness values is generated.

Next, the autocorrelation function is used to determine the correlation structure of this time series. The autocorrelation function ρ_i relates the value of two points in the time series which are i steps apart (called *times lags* in case of a time series). The autocorrelation for time lag i of a time series $y_t, t = 1, .., T$ is defined as:

$$\rho_i = Corr(y_t, y_{t+i}) = \frac{E[y_t y_{t+i}] - E[y_t]E[y_{t+i}]}{Var(y_t)}$$

where $E[y_t]$ is the expected value of y_t and $Var(y_t)$ is the variance of y_t. It always holds that $-1 \leq \rho_i \leq 1$. If $|\rho_i|$ is close to one, then there is much correlation between two values i steps apart; if it is close to zero, then there is hardly any correlation. Estimates r_i of these autocorrelations ρ_i are:

$$r_i = \frac{\sum_{t=1}^{T-i}(y_t - \overline{y})(y_{t+i} - \overline{y})}{\sum_{t=1}^{T}(y_t - \overline{y})^2}$$

where $\overline{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$ and $T \gg 0$ (Box and Jenkins, 1970; Granger and Newbold, 1986).

The autocorrelation function of this time series (or rather its estimate) is then taken as a measure of the correlation structure of the landscape. An important assumption made here is that the fitness landscape is *statistically isotropic.* This means that the statistics of the time series generated by a random walk are the same, regardless of the starting point. In other words, the landscape looks (globally) the same everywhere. The significance of statistical isotropy is that the random walk is "representative" of the entire landscape, and thus that the correlation structure of the time series can be regarded as the correlation structure of the landscape.

4.2 Correlation length

Different definitions for the *correlation length* of a landscape have been used in the literature (see for example (Weinberger, 1990; Manderick et al., 1991; Lipsitch, 1991)). In a statistical sense, the correlation length gives an indication of the largest "distance" (time lag) between two points at which the value of one point can still provide some information about the expected value of the other point. In other words, the correlation length τ is the largest time lag i for which there is still some correlation between two points i steps apart. Since the exact values of the ρ_i are not known, the estimates r_i have to be used to determine when the correlation becomes equal to zero.

In estimating the value of a variable, it is usual to provide a standard error[1] of the estimate, which can be used to determine whether the estimated value is significantly different from zero. For the r_i, the estimates of the ρ_i, the standard error is approximately $1/\sqrt{T}$ (Box and Jenkins, 1970; Granger and

[1] The statistical estimation of a variable will never be exact, but contains some uncertainty. The standard error of an estimated value gives an indication of the amount of this uncertainty.

Newbold, 1986), where T is the length of the time series. Following (Granger and Newbold, 1986) one can assume normality in moderately large samples, and thus comparison of the sample quantities with $\pm 2/\sqrt{T}$ gives a useful guide to statistical significance. In other words, whenever the value of r_i is outside the region $(-2/\sqrt{T}, +2/\sqrt{T})$ it can be considered significantly different from zero.

The definition of the correlation length τ of a time series can thus be given as one less than the first time lag i for which the estimated autocorrelation r_i falls inside the region $(-2/\sqrt{T}, +2/\sqrt{T})$, and thus becomes not significantly different from zero. This way, the correlation length τ is the largest time lag i for which the correlation between two points i steps apart is still statistically significant. This definition is different from the less general ones used in (Weinberger, 1990; Manderick et al., 1991), because their definitions depend on the assumption that the correlations fall off exponentially with increasing i. The above definition does not depend on this assumption.

4.3 Time series analysis

The method introduced by Weinberger is restricted to estimating the autocorrelation function of a time series of fitness values, and the corresponding correlation length. Having a time series at hand, however, automatically invites one to do a complete *time series analysis*. This involves identifying an appropriate model (usually within a certain class of models) that adequately represents the data generating process (in this case the random walk that generated the time series), then estimating the parameters of this model, and applying statistical tests to see how well the estimated model approximates the given data and what the explanatory and predictive value of the model is. In other words, a model of the form

$$y_{t+1} = f(y_t, y_{t-1}, \cdots, y_0)$$

is derived from the observed data, which can then be used to simulate the outcome of a random process similar to the one that generated the original data, or to predict future values in the time series.

Applying a time series analysis to a time series of fitness values generated by a random walk on a fitness landscape can thus give more information about the correlation structure of the landscape than just the estimated autocorrelations. The derived model can be used to simulate walks or searches on the landscape, or might even be used to predict future values during an actual search. Furthermore, the particular form of the identified model can give additional information about the process that generated the data.

4.4 The Box-Jenkins approach

The Box-Jenkins approach (Box and Jenkins, 1970) is a very useful standard statistical method of model building, based on the analysis of a time series $y_1, y_2, \ldots, y_T$, generated by a stochastic process. The purpose of the Box-Jenkins approach is to find an ARMA (autoregressive moving-average) model that adequately represents this data generating process. Once an adequate model is found, it can be used to make forecasts about future values, or to simulate a process similar to the one that generated the original data.

An ARMA model is obtained by combining an autoregressive (AR) model and a moving-average (MA) model. An AR model of order p (AR(p)) has the form

$$y_t = \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + \varepsilon_t$$

where the stochastic variable ε_t is white noise, that is, $E[\varepsilon_t] = 0$, $Var(\varepsilon_t) < \infty$ for all t, and $Cov(\varepsilon_s, \varepsilon_t) = 0$ for $s \neq t$, so all ε_t are independent of one another. In words, each value y_t in an AR(p) process depends on p past values and some stochastic variable ε_t. An MA model of order q (MA(q)) has the form

$$y_t = \varepsilon_t + \beta_1 \varepsilon_{t-1} + \cdots + \beta_q \varepsilon_{t-q}$$

where ε_t is again white noise. In words, each value y_t in an MA(q) process is a weighted sum of members of a white noise series. An ARMA(p, q) model, then, is the sum of an AR(p) and an MA(q) model:

$$y_t = \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \cdots + \beta_q \varepsilon_{t-q}$$

The mean of a time series generated by any of these three models is zero. If this does not match the original data, then either a constant c can be added to the model, or the mean of the original data can be subtracted from all data points.

In economics the Box-Jenkins approach is used frequently when a model is needed to make forecasts about future values of some (partly) stochastic variable, for example the price of some commodity, or the index of industrial production (see for example (Granger and Newbold, 1986; Judge et al., 1988)). The approach consists of three stages:

1. *Identification*, in which a choice is made for an appropriate ARMA model, by looking at the estimated autocorrelations and partial autocorrelations of the time series.
2. *Estimation*, in which the parameters of the chosen model are estimated.
3. *Diagnostic checking*, in which the estimated model is tested for adequateness.

Identification

At the identification stage an appropriate model, within the class of ARMA models, is specified on the basis of the *correlogram* and the *partial correlogram*. The correlogram of a time series y_t is a plot of the (estimated) autocorrelations (the r_i as given in Section 4.1) of this series against the time lag i. The partial correlogram is the plot of the (estimated) partial autocorrelations of the time series against the time lag. The ith partial autocorrelation can be interpreted as the estimated correlation between y_t and y_{t+i}, after the effects of all intermediate y's on this correlation are taken out. The partial autocorrelation of order K, a_{KK}, can be found by solving the set of simultaneous linear equations

$$\rho_i = \sum_{j=1}^{K} a_{Kj} \rho_{i-j}, \quad i = 1, 2, \ldots, K$$

Estimates of a_{KK} can be found by replacing the ρ_i with the r_i in the above equation.

The choice of an appropriate model is now made on the following basis (Box and Jenkins, 1970; Judge et al., 1988):

- If the correlogram tapers off to zero and the partial correlogram suddenly cuts off (i.e., becomes equal to zero) after some time lag, say p, then an appropriate model is AR(p).
- If the correlogram cuts off after some time lag, say q, and the partial correlogram tapers off to zero, then an appropriate model is MA(q).

- If neither diagram cuts off after some time lag, but both taper off, then an appropriate model is ARMA(p,q). The values of p and q have to be inferred from the particular pattern of the two diagrams.

To determine the (possible) cut-off point for the estimated autocorrelations or partial autocorrelations, they are compared to $\pm 2/\sqrt{T}$, which is an appropriate bound for significance of the estimates, as explained in section 4.2 ($1/\sqrt{T}$ is also the approximate standard error for the estimates of the partial autocorrelations, where T is again the length of the time series). Whenever the estimated (partial) autocorrelations are outside the $\pm 2/\sqrt{T}$ significance bound, they can be considered to be significantly different from zero.

Estimation

Once the appropriate model is chosen, the parameters of this model are estimated. This is achieved by using the estimates of the autocorrelations of the given time series. From these values, estimates for the parameters of the model can be derived (see (Box and Jenkins, 1970; Granger and Newbold, 1986; Judge et al., 1988)).

There is always some uncertainty in the estimation of a parameter. Therefore, it is important to know whether a parameter is *significant* or whether it might as well be considered to be equal to zero. As a measure of significance of an estimated parameter the t-statistic is used here. To test whether an estimated parameter is significantly different from zero, the t-statistic is calculated as the estimated value of the parameter divided by its estimated standard error. This value can then be compared to that of the t-distribution at an appropriate confidence level to determine whether the estimated parameter can be considered significant. Here a 95% confidence level will be assumed. For large sample sizes, the t-distribution at this confidence level attains a value of 1.96, or approximately 2. So an estimated parameter will be considered significant if its t-statistic (in absolute value) is larger than 2.

As a measure of "goodness of fit" of the estimated model, the *coefficient of determination* R^2 is used (Judge et al., 1988). This value is a measure of the proportion of the total variance in the data that is accounted for by the explanatory variables in the estimated model ($0 \leq R^2 \leq 1$). A value of R^2 close to one means that the explanatory variables can explain the observed data very well. A value of R^2 close to zero means that the stochastic component of the model plays a dominant role (or it could be that there exist more explanatory variables than there are currently in the model; this will usually not be the case here, because it is assumed that in the identification stage an appropriate model is already chosen). Another interpretation of R^2 is that it is the square of the correlation coefficient between the observed data (y_t) and the data produced (estimated) by the model ($\hat{y}_t$):

$$R^2 = \frac{\left[\sum_{t=1}^{T}(y_t - \bar{y})(\hat{y}_t - \bar{y})\right]^2}{\sum_{t=1}^{T}(y_t - \bar{y})^2 \sum_{t=1}^{T}(\hat{y}_t - \bar{y})^2}$$

where $\bar{y} = \sum_{t=1}^{T} y_t$. So if there is a high correlation between the observed and model data (R^2 close to 1), then the model agrees very well with the observed data.

Diagnostic checking

Before the estimated model is used, it is important to check that it is an adequate one. The usual test is to fit the model on the data and calculate the autocorrelations of the residuals (the difference between the observed values (y_t) and those predicted by the model ($\hat{y}_t$)). These residuals should be white noise (i.e., uncorrelated, zero-mean variables), meaning that all these residual autocorrelations should not be significantly different from zero. To check this, the residual autocorrelations are compared with the same $\pm 2/\sqrt{T}$ bound (as with the autocorrelations and partial autocorrelations). Another test is to fit a

slightly higher-order model and then check that the extra parameters are not significantly different from zero (using the t-statistic). Thus, if for example an AR(p) model is estimated, an AR($p+1$) model is also estimated and the extra parameter (α_{p+1}) should not be significant.

Summarizing, the Box-Jenkins approach is used to find an appropriate model, within the class of ARMA-models, for a given time series generated by some stochastic process, to make forecasts for, or simulations of, the process that generated the original data.

There are several motivations for using the Box-Jenkins approach here. First of all, it is based on calculating the autocorrelations, so it is related to earlier methods for measuring the correlation structure of a landscape. However, the proposed analysis has to offer more than these earlier methods. It also provides a complete (stochastic) model that can be used to make quantitative predictions or perform simulations. Finally, the model that results from using the approach indicates the amount of memory in the process that generated the data (by the order p of the autoregressive part of the model). As was mentioned in the introduction, this might be helpful for improving search methods by using this memory to calculate an expected fitness increase at each step, instead of going uphill as fast as possible.

4.5 The statistical fitness landscape analysis

Using the concepts and methods introduced in the previous subsections, the complete statistical fitness landscape analysis can now be put together.

In the fitness landscape framework, as described in Section 2, there is a specific search operator (or combination of operators) that defines the neighborhood relation for the fitness landscape. This operator (or combination) is used to perform a random walk on the landscape. The algorithm is as follows: Start at a random point in the landscape and record its fitness. Then apply the operator(s) once to this point, resulting in a new point which is a neighbor of the former one, and record the fitness of this new point. Repeat applying the operator(s) and recording the fitness of each new point for a certain number of steps. This way a time series of fitness values is generated.

Next, the Box-Jenkins approach is applied to this time series, providing the estimated autocorrelations of this time series, a stochastic model that adequately represents the data-generating process (i.e., the random walk), and some statistics indicating the explanatory and predictive value of the estimated model. Furthermore, from the estimated autocorrelations a correlation length can be derived. All these statistics together can then be used to express the correlation structure of the fitness landscape, or to make quantitative predictions about search on the landscape.

This method can be applied with every possible search operator or combination of search operators, as long as the search operator is well defined, meaning that applying the operator to a genotype (or possibly a set of genotypes) yields a valid new genotype (or set of genotypes), on which the operator can be applied again. Most, if not all, known operators in evolutionary algorithms have this property: selection (going from a population of genotypes to a new population), mutation (going from one genotype to another), crossover (going from one genotype to another, or from a pair of genotypes to another pair, depending on the exact implementation of the operator), etc. Each well defined operator induces a fitness landscape, and the fitness landscapes of these separate operators, and of combinations of operators, can be analyzed by the above method.

Furthermore, if a landscape is not isotropic, the random walk can be restricted to a subspace of the

landscape that is locally isotropic, and several subspaces of the same landscape can be compared to one another. This is, for example, done in (Hordijk, 1996). However, the difficulty here is to identify these isotropic subspaces, which might impose a limitation on the method.

5 The Correlation Structure of NK-Landscapes

In this section the landscape analysis, as introduced in the previous section, is applied to several landscapes constructed with the NK-model. Bit strings of length 100 ($N = 100$) are used, with the following values for K: 0, 2, 5, 25, 50, and 95. Both the random interaction model and the nearest neighbor interaction model are used. Three different neighborhood relations are used, based on the following search operators:

1. Point mutation
2. Random-mate random-child one point crossover (RMRC crossover)
3. A combination of 1 and 2.

This gives rise to a total of 6 (K-values) $\times$ 2 (types of interactions) $\times$ 3 (neighborhood relations) = 36 different fitness landscapes.

The search operators are defined as follows (operating on bit strings):

Point mutation: At every step one randomly chosen bit in the bit string is flipped ($0 \leftrightarrow 1$).

Random-mate random-child one-point crossover: At every step a mate is randomly chosen out of all possible bit strings. A crossover point is selected at random, and the parts of the current bit string and the mate after this crossover point are exchanged, creating two children. One of these two children is selected at random (with equal chance) for the next step.

Combination of crossover and mutation: At every step first RMRC crossover is applied to the current bit string, after which point mutation is applied to the resulting string.

Since the main interest here is the "global" structure of a landscape, crossover is defined this way so that the walk is not restricted to a specific part of the landscape (which will happen if, for example, a fixed mate is used).

On each of the 36 different landscapes, the operator (or combination) that defines the neighborhood relation for that landscape is used to perform a random walk of 10,000 steps. The next three subsections present the results of applying the complete fitness landscape analysis to the different landscapes.

5.1 Results for point mutation

The landscape
Figure 1 gives an example of a fitness landscape for bit strings of length 3, with randomly assigned fitness

values, and where the neighborhood relation is defined by the operator point mutation. Bit strings that differ in the value of only one of the bits are each others neighbors (indicated by an edge between the two points). In general, the fitness landscape for point mutation is simply the boolean hypercube.

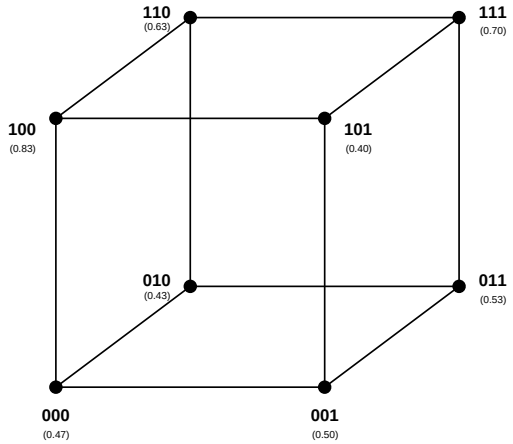

Figure 1: An example of a fitness landscape for bit strings of length 3, where the neighborhood relation is defined by point mutation. Fitness values (randomly assigned) are shown in parentheses.

Identification

The correlograms for the different values of K are given in Figures 2 (random interactions) and 3 (nearest neighbor interactions), together with the $\pm 2/\sqrt{T}$ bound for significance (± 0.02 for T=10,000). The correlograms all taper off to zero (except for K=95 with nearest neighbor interactions; here the autocorrelations almost immediately drop to zero). Table 2 gives the correlation lengths (as defined in Section 4.2) for the different values of K. Both the plots and the table show that there is little difference between the random interaction version and the nearest neighbor interaction version of the NK-model for point mutation.

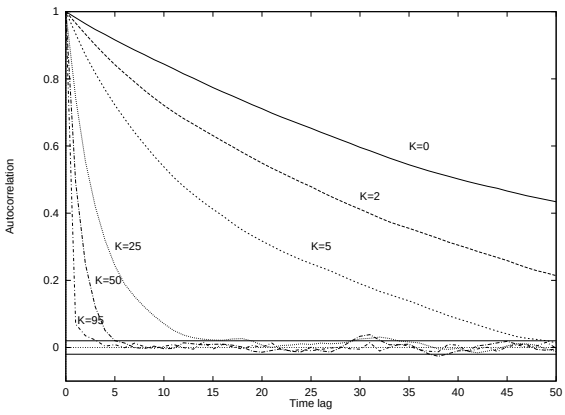

Figure 2: The first 50 autocorrelations for point mutation for different values of K with random interactions.

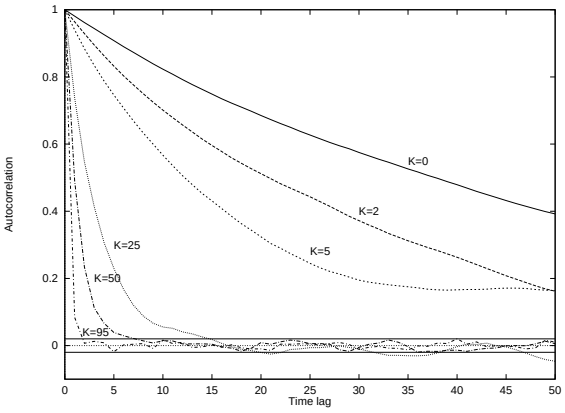

Figure 3: The first 50 autocorrelations for point mutation for different values of K with nearest neighbor interactions.

To see which model is the most appropriate, the partial correlogram for K=0 with random interactions, is shown in Figure 4. This plot shows that the first partial autocorrelation is almost equal to one, and thus well outside the ± 0.02 significance bound. The other partial autocorrelations are all within this bound (apart from one minor exception). So it is clear that the partial correlogram cuts off after one time lag,

K	random interactions	nearest neighbor interactions
0	>50	>50
2	>50	>50
5	49	>50
25	18	14
50	5	7
95	3	1

Table 2: The correlation lengths for the operator point mutation for different values of K.

and thus an AR(1) model is the most appropriate choice here. The partial correlograms for the other values of K and for nearest neighbor interactions look similar, except for K=95 with random interactions. In this case, the first two partial autocorrelations are outside the significance bound, suggesting an AR(2) model.

Estimation

In all cases an AR(1) model of the form $y_t = c + \alpha_1 y_{t-1} + \varepsilon_t$ is estimated, except for K=95 with random interactions, where an AR(2) model of the form $y_t = c + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \varepsilon_t$ is estimated. The constant is added because the mean of the time series is not equal to zero. Table 3 shows the results of the estimation. The t-statistics (see Section 4.4) of the estimated parameters are shown in parentheses.

	random interactions					nearest neighbor interactions				
K	c	α_1	α_2	$Var(\varepsilon_t)$	R^2	c	α_1	α_2	$Var(\varepsilon_t)$	R^2
0	0.01 (9.59)	0.98 (518.20)	-	0.00002	0.96	0.01 (9.94)	0.98 (510.85)	-	0.00002	0.96
2	0.02 (13.08)	0.97 (375.48)	-	0.00005	0.93	0.02 (13.47)	0.96 (364.39)	-	0.00005	0.93
5	0.03 (18.46)	0.93 (261.59)	-	0.00010	0.87	0.03 (17.55)	0.94 (275.72)	-	0.00010	0.88
25	0.13 (38.35)	0.74 (110.91)	-	0.00038	0.55	0.13 (38.86)	0.74 (109.00)	-	0.00038	0.54
50	0.25 (58.29)	0.49 (56.41)	-	0.00064	0.24	0.25 (58.40)	0.49 (56.21)	-	0.00063	0.24
95	0.45 (66.09)	0.07 (6.76)	0.03 (3.04)	0.00082	0.01	0.46 (91.76)	0.08 (8.42)	-	0.00082	0.01

Table 3: The results of the estimation for point mutation for different values of K.

The table shows that all parameters are significant (t-statistic > 2). It also shows clearly that the value of the α_1 parameter decreases and the variance of the error term ($Var(\varepsilon_t)$) increases as K increases. Note that the value of α_1 decreases linearly with increasing K. Furthermore, the table shows that the R^2 decreases as K increases, so the explanatory variables in the estimated model are less capable of explaining the variance in the observed data for higher values of K. These results verify the fact that the landscapes become more rugged for higher K-values. The table, just as the correlograms, shows little difference between random and nearest neighbor interactions for point mutation.

Diagnostic checking

Figure 5 shows the first 25 residual autocorrelations for K=0 with random interactions for the estimated model. This plot shows that they are all well within the ± 0.02 significance bound (apart from one minor exception). The plots for the other values of K and for nearest neighbor interactions look similar.

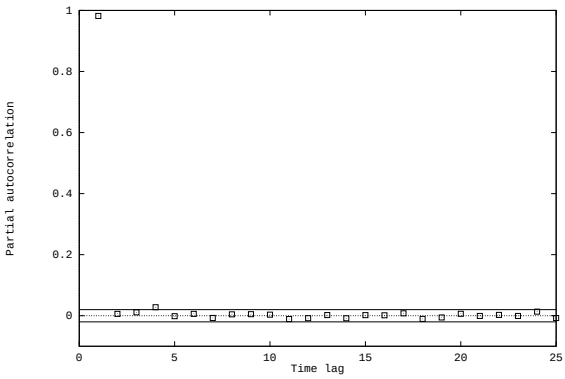

Figure 4: The first 25 partial autocorrelations for point mutation for $K = 0$ with random interactions.

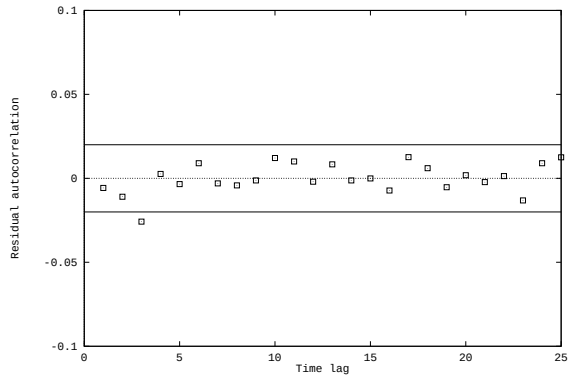

Figure 5: The first 25 residual autocorrelations for point mutation for $K = 0$ with random interactions.

To verify once more the adequateness of the estimated models, an AR(2) model is estimated for all cases (AR(3) for K=95 with random interactions). The results are presented in Table 4.

	random interactions		nearest neighbor interactions	
K	α_2	t-statistic	α_2	t-statistic
0	0.01	0.62	-0.01	-0.55
2	-0.01	-1.10	-0.01	-1.44
5	-0.01	-0.72	0.01	0.64
25	0.01	0.53	0.01	0.96
50	0.01	0.81	-0.01	-0.75
95	0.02 (α_3)	1.80	-0.00	-0.01

Table 4: The results of the overestimation of the chosen model for point mutation for different values of K.

As expected, the extra parameter is not significantly different from zero in all cases. So it seems that the identified and estimated models for point mutation on the different landscapes are adequate in all cases.

It turns out that in all cases except one, an AR(1) model is the most appropriate for expressing the correlation structure of the fitness landscapes. This means that the fitness of the current point, y_t, only depends on that of the previous point, y_{t-1}, and there is no further memory in the process. The fact that landscapes generated by the NK-model, using point mutation to define the neighborhood relation, give rise to AR(1) models was already suggested in (Weinberger, 1990) and is verified here. Section 6 will give an example of how the resulting models can be used in predicting actual search results on the landscape.

5.2 Results for RMRC crossover

The landscape

Figure 6 gives an example of a fitness landscape for bit strings of length 3, with randomly assigned fitness values, and where the neighborhood relation is defined by the operator RMRC crossover. Bit strings that can be reached from each other by one application of one-point crossover with a random mate are each others neighbors (indicated by an edge between the two points). Note that this landscape looks different from the familiar hypercube. In this crossover landscape, a point is connected to all other points, except for those that have a different value for both the first and the last bit. Furthermore, each point is also connected to itself. However, the probabilities of reaching different points from one particular point are not the same. Starting from a particular point, some points have a higher probability of being reached than other points. In a more elaborate model introduced in (Jones, 1995), the edges in the landscape graph are labeled with these "transition" probabilities.

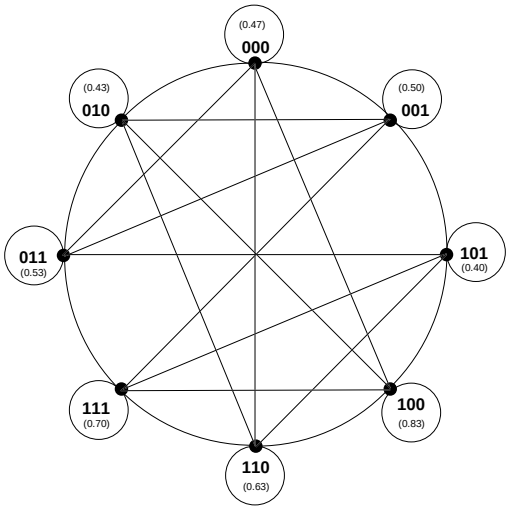

Figure 6: An example of a fitness landscape for bit strings of length 3, where the neighborhood relation is defined by RMRC crossover. Fitness values (randomly assigned) are shown in parentheses.

Identification

The correlograms for RMRC crossover for the different values of K are presented in Figures 7 (random interactions) and 8 (nearest neighbor interactions). The correlograms taper off to zero, but much faster than for point mutation, and for higher K-values, random interaction, they drop off to zero right away. The plots show furthermore that for RMRC crossover there is some difference between random interactions and nearest neighbor interactions. This is shown more clearly in Table 5, which shows the correlation lengths for the different values of K.

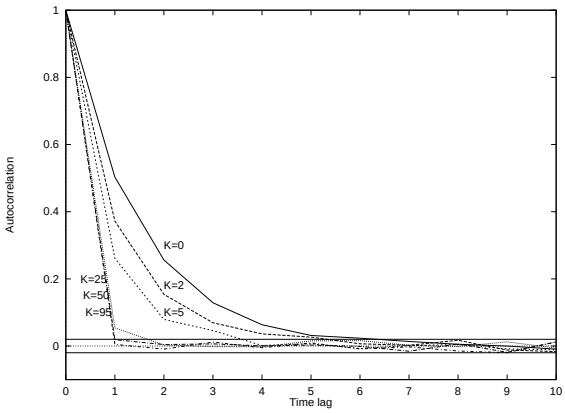

Figure 7: The first 10 autocorrelations for RMRC crossover for different values of K with random interactions.

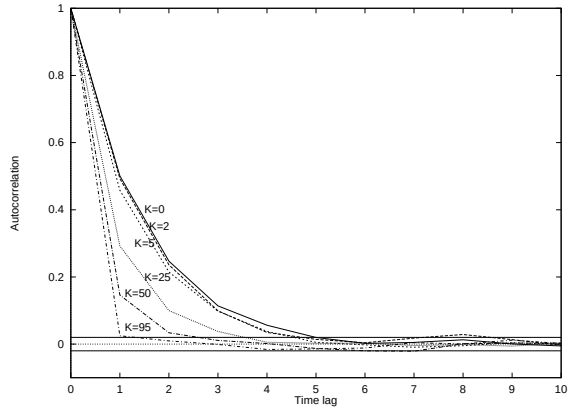

Figure 8: The first 10 autocorrelations for RMRC crossover for different values of K with nearest neighbor interactions.

K	random interactions	nearest neighbor interactions
0	6	4
2	5	4
5	3	4
25	1	3
50	0	2
95	0	1

Table 5: The correlation lengths for RMRC crossover for different values of K.

The table shows that the correlation dies out for random interactions, whereas for nearest neighbor interactions there still remains some correlation as K increases. This difference can be explained by the fact that one-point crossover is less disruptive in case of nearest neighbor interactions. If the epistatic interactions in the NK-model are spread randomly across the genotype, then every possible crossover point will affect the epistatic relations of almost all bits in the string, especially for larger values of K. But if the epistatic interactions are the K neighboring bits, then only the epistatic relations of the bits in the vicinity of the crossover point are affected (this is stated in another way in the *Schema Theorem* for genetic algorithms, which implies that schemata of short defining length have a higher chance to survive under one-point crossover than longer ones; see (Holland, 1992; Goldberg, 1989)). Thus, if more epistatic relations stay intact, more bits will keep the same fitness, and the fitness of the entire genotype in the next step will be more correlated to that of its parents.

Looking at the partial correlogram for K=0 (random interactions), as shown in Figure 9, it can be concluded that an AR(1) model is the most appropriate model since only the first partial autocorrelation is well outside the significance bound. The partial correlograms for the other values of K and for nearest neighbor interactions look similar, except for K=50 and 95 with random interactions. In these cases, all partial autocorrelations are within the ± 0.02 bound, suggesting just a white noise model.

Estimation

As with point mutation, an AR(1) model with a constant is estimated here, except for K=50 and 95

with random interactions, where a white noise model is estimated. The results are presented in Table 6 (t-statistics shown between parentheses).

	random interactions				nearest neighbor interactions			
K	c	α_1	$Var(\varepsilon_t)$	R^2	c	α_1	$Var(\varepsilon_t)$	R^2
0	0.26 (57.52)	0.50 (58.21)	0.00034	0.25	0.23 (57.57)	0.50 (57.93)	0.00036	0.25
2	0.32 (67.56)	0.37 (40.06)	0.00061	0.14	0.27 (58.21)	0.49 (56.62)	0.00056	0.24
5	0.37 (76.42)	0.26 (27.04)	0.00073	0.07	0.27 (60.85)	0.46 (51.54)	0.00061	0.21
25	0.47 (94.56)	0.05 (5.41)	0.00084	0.01	0.35 (73.87)	0.29 (30.56)	0.00075	0.09
50	0.49 (97.92)	-	0.00082	-	0.43 (86.14)	0.15 (14.80)	0.00083	0.02
95	0.50 (99.31)	-	0.00085	-	0.49 (97.38)	0.02 (2.48)	0.00082	0.01

Table 6: The results of the estimation for RMRC crossover for different values of K.

All parameters appear to be significant. The table shows that for a specific value of $K > 0$ the estimated value of α_1 is larger for nearest neighbor interactions than for random interactions (for the extreme of K=0 there is no difference between random and nearest neighbor interactions, see Section 3). This indicates that there is indeed more correlation for one-point crossover on a landscape with nearest neighbor interactions than on a landscape with random interactions. Furthermore, for a specific value of $K > 0$, the R^2 values are also larger for nearest neighbor interactions, which means that the estimated model has more explanatory and predictive power than it does for random interactions. For K=50 and 95 with random interactions, the value of R^2 can not be computed, because there is no variance in the estimated data $\hat{y}_t$ (which are all just equal to the constant c in the model). As with point mutation, the value of the α_1 parameter decreases, and the variance of the error term increases with increasing K, indicating an increase in ruggedness of the landscapes.

Diagnostic checking

Figure 10 shows the first 25 residual autocorrelations for K=0 (random interactions). All are well within the ± 0.02 significance bound. The plots for larger K values and nearest neighbor interactions look similar.

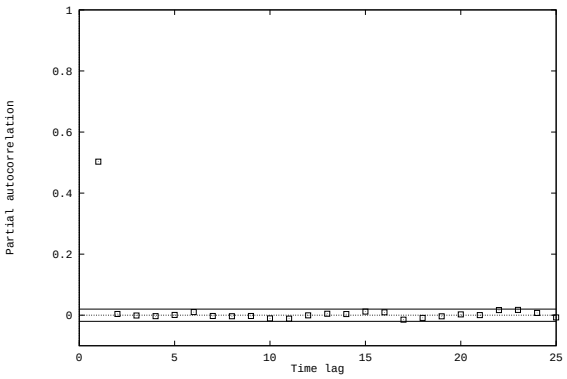

Figure 9: The first 25 partial autocorrelations for crossover for $K = 0$ with random interactions.

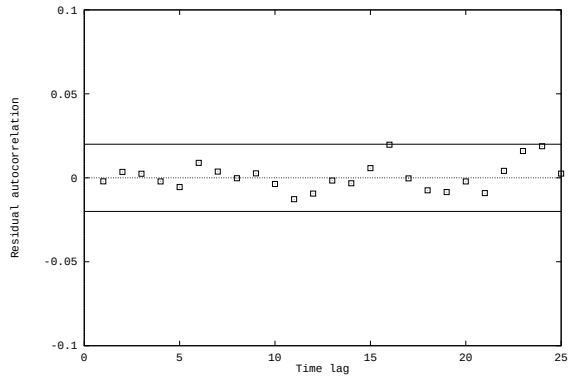

Figure 10: The first 25 residual autocorrelations for crossover for $K = 0$ with random interactions.

Table 7 gives the results of the overestimation, the extra check on the adequacy of the chosen model. The table shows that all extra parameters are not significantly different from zero, indicating once more that the AR(1) model is adequate (except for K=50 and 95 with random interactions, which are just white noise processes).

	random interactions		nearest neighbor interactions	
K	α_2	t-statistic	α_2	t-statistic
0	0.01	0.41	-0.01	-0.62
2	0.02	1.77	-0.01	-0.87
5	0.01	1.20	0.01	0.61
25	0.00	0.01	0.02	1.58
50	0.02 (α_1)	1.92	0.01	1.30
95	0.00 (α_1)	0.48	0.01	0.91

Table 7: The results of the overestimation of the chosen model for RMRC crossover for different values of K.

In the case of RMRC crossover, all models turn out the be AR(1) models, with the exception of two cases, where the appropriate model is just a white noise model. The correlations and correlation lengths are smaller than in the case of mutation, and there is a clear difference between random interactions and nearest neighbor interactions.

5.3 Results for RMRC crossover and mutation

The landscape

Figure 11 gives an example of a fitness landscape for bit strings of length 3, with randomly assigned fitness values, and where the neighborhood relation is defined by a combination of the operators RMRC crossover and point mutation. Bit strings that can be reached from each other by a consecutive application of RMRC crossover and point mutation are each others neighbors (indicated by an edge between the two points). It turns out that the landscape graph is the complete graph including loops at each point. Each point can be reached from each other point, but again with different probabilities (not shown here).

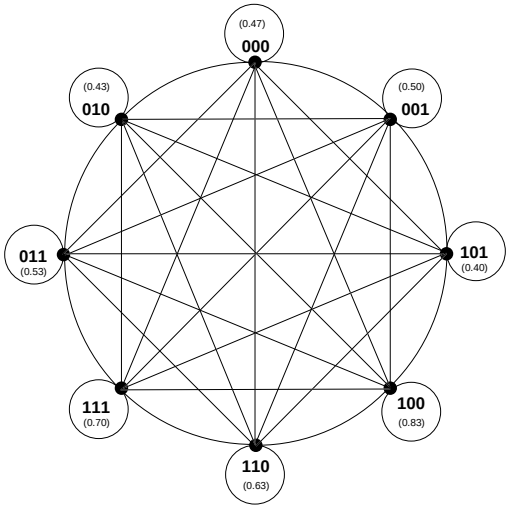

Figure 11: A fitness landscape for bit strings of length 3, where the neighborhood relation is defined by a combination of RMRC crossover and point mutation. Fitness values (randomly assigned) are shown in parentheses.

Identification

The correlograms for the combination of RMRC crossover and point mutation for the different values of K are presented in Figures 12 (random interactions) and 13 (nearest neighbor interactions). The correlograms taper off to zero, except for the highest K-values, where they drop off to zero at once. The plots show furthermore that for the combination of RMRC crossover and point mutation there is some difference between random interactions and nearest neighbor interactions (as with RMRC crossover alone). Table 8 gives the correlation lengths for the different values of K.

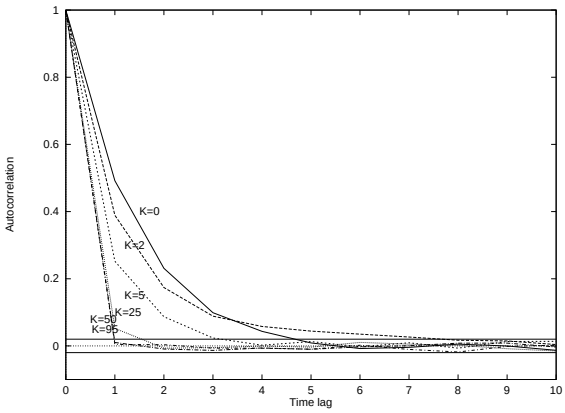

Figure 12: The first 10 autocorrelations for the combination of RMRC crossover and point mutation for different values of K with random interactions.

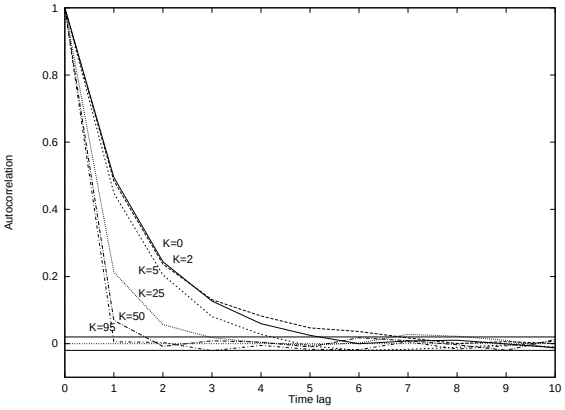

Figure 13: The first 10 autocorrelations for the combination of RMRC crossover and point mutation for different values of K with nearest neighbor interactions.

Again, there is still some correlation left for nearest neighbor interactions at K=50, while for random interactions the correlation has died out. But here also for nearest neighbor interactions the correlation eventually dies out as K increases.

K	random interactions	nearest neighbor interactions
0	4	5
2	7	6
5	3	4
25	1	2
50	0	1
95	0	0

Table 8: The correlation lengths for the combination of RMRC crossover and point mutation for different values of K.

Looking at the partial correlogram for K=0 (random interactions), as shown in Figure 14, it can be concluded that an AR(1) model is the most appropriate model since only the first partial autocorrelation is well outside the significance bound. The partial correlograms for most other values of K and for nearest neighbor interactions look similar. For K=2 and 5 with random interactions, also the second partial autocorrelation is outside the bound, suggesting an AR(2) model. For K=50 and 95 with random interactions, and K=95 with nearest neighbor interactions, all partial autocorrelation are within the bound (as are the autocorrelations for those cases), suggesting a white noise model.

Estimation

The results of the estimation of the identified models are presented in Table 9 (t-statistics shown between parentheses).

	random interactions					nearest neighbor interactions				
K	c	α_1	α_2	$Var(\varepsilon_t)$	R^2	c	α_1	α_2	$Var(\varepsilon_t)$	R^2
0	0.27 (58.34)	0.49 (56.49)	-	0.00030	0.24	0.26 (58.09)	0.49 (56.92)	-	0.00034	0.24
2	0.31 (53.76)	0.38 (37.9)	0.03 (2.65)	0.00058	0.15	0.26 (58.81)	0.48 (55.41)	-	0.00057	0.23
5	0.37 (59.58)	0.25 (24.57)	0.03 (2.51)	0.00078	0.06	0.28 (61.63)	0.45 (50.12)	-	0.00066	0.20
25	0.47 (94.75)	0.05 (5.21)	-	0.00084	0.01	0.39 (80.39)	0.21 (21.81)	-	0.00081	0.05
50	0.50 (1733.32)	-	-	0.00083	-	0.47 (93.17)	0.07 (6.93)	-	0.00081	0.01
95	0.50 (1733.36)	-	-	0.00083	-	0.50 (1750.52)	-	-	0.00082	-

Table 9: The results of the estimation for the combination of RMRC crossover and point mutation for different K-values.

Again all parameters appear to be significant. Similarly to RMRC crossover alone, for a specific value of $K > 0$ the estimated value of α_1 is larger for nearest neighbor interactions than for random interactions (as are the R^2 values). The results of this estimation are not really a surprise, considering the results for the two operators separately, but they are included here merely to show that the proposed landscape analysis is also possible for combinations of search operators. The results agree nicely with what can be expected from the separate results, showing the generality and consistency of the method.

Diagnostic checking

Figure 15 shows the first 25 residual autocorrelations for K=0 (random interactions). All are well within the ± 0.02 significance bound. The plots for larger K values and nearest neighbor interactions look similar.

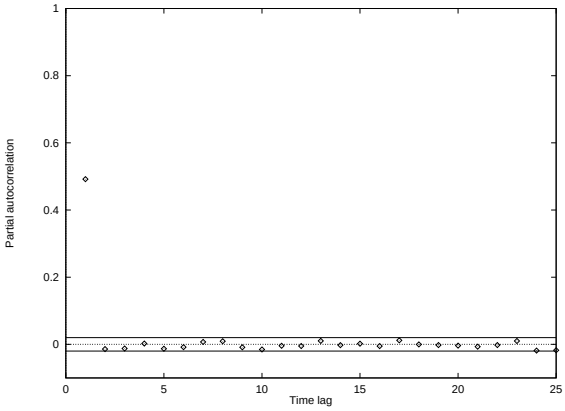

Figure 14: The first 25 partial autocorrelations for the combination of RMRC crossover and point mutation for $K = 0$ with random interactions.

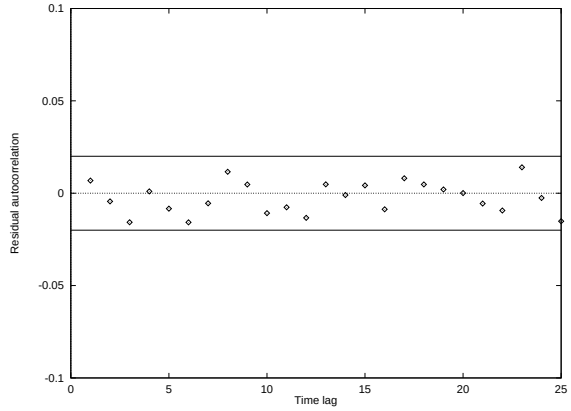

Figure 15: The first 25 residual autocorrelations for the combination of RMRC crossover and point mutation for $K = 0$ with random interactions.

Table 10 gives the results of the overestimation. The table shows that all extra parameters are not significantly different from zero, indicating once more that the estimated models are adequate.

	random interactions		nearest neighbor interactions	
K	α_2	t-statistic	α_2	t-statistic
0	-0.01	-1.41	-0.00	-0.04
2	0.01 (α_3)	1.46	0.00	0.24
5	-0.00 (α_3)	-0.43	0.00	0.48
25	-0.01	-0.93	0.01	1.18
50	0.01 (α_1)	1.03	-0.01	-1.28
95	0.00 (α_1)	0.46	0.01 (α_1)	0.58

Table 10: The results of the overestimation of the chosen model for the combination of RMRC crossover and point mutation for different values of K.

For the combination of RMRC crossover and point mutation, there are several models possible — white noise, AR(1) or AR(2) — depending on the exact value the parameter K in the NK-model. The results are more or less what could be expected from the results of the two operators separately, but this shows that the analysis also works well when a combination of operators is used, which is more realistic in terms of most evolutionary algorithms.

6 Landscape structure and search

As was claimed earlier, one advantage of the analysis as introduced and shown here is that it provides a complete stochastic model that can be used to make quantitative predictions about search on the

landscape. This section will show one particular example of how the resulting models can be used to make such predictions.

An actual search on landscapes generated with the NK-model is performed with an implementation of hill climbing which will be called Random Neighbor Ascent Hill Climbing (RNAHC; this method was used in (Kauffman, 1993)). The algorithm works as follows: Start with a random point in the landscape and calculate its fitness. Calculate the fitness of all the one-mutant neighbors (i.e., all the points that differ in the value of only one bit), and record the ones that are fitter than the current point. Choose one of the fitter neighbors at random, and move to that point. Repeat this procedure of climbing uphill until no fitter neighbors can be found. The algorithm is given a total of 10,000 function evaluations (calculating the fitness of a point is one evaluation), so when after reaching a local optimum this number of evaluations is not reached yet, a new uphill walk from a random point is started, until the 10,000 function evaluations are reached.

Figure 16 shows the result of this search on a landscape with N=100 and K=25 with random interactions, averaged over 100 different runs (the solid line labeled "RNAHC"). The best fitness value found so far is plotted against the number of function evaluations. The dashed line labeled "AR(1) prediction" shows the prediction made (also averaged over 100 runs) using the AR(1) model that resulted from the analysis for this landscape for point mutation:

$$y_t = 0.13 + 0.74 y_{t-1} + \varepsilon_t \qquad Var(\varepsilon_t) = 0.00038$$

where ε_t is normally distributed (as is implied in an ARMA model) with zero mean and variance 0.00038 (as given by the analysis).

In the RNAHC algorithm each uphill step consists of 100 function evaluations (calculating the fitness of all the 100 neighboring points), and a fitter neighbor is chosen at random. This is simulated using the above AR(1) model as follows: Start with a fitness value y_0 around 0.5 (which is the expected fitness of a random point in a landscape generated with the NK-model). Calculate y_{t+1} using the given AR(1) model, drawing an ε_{t+1} from a normal distribution with mean zero and variance 0.00038. Choosing a fitter neighbor in the hill climber is simulated here by taking the absolute value of ε_{t+1} (giving a positive error term). If this new y-value is larger than the previous one, then accept it, otherwise keep the old value. Repeat this procedure for 100 steps (equivalent to 10,000 function evaluations).

As Figure 16 shows, this quantitative prediction using the AR(1) model comes fairly close to the real result. Of course there are still some differences between the two processes that are not taken into account in the prediction — for example the fact that the real hill climber can start at a new random point after getting stuck on a local optimum. But even despite those differences, the plot shows clearly that the model that results from the landscape analysis does have predictive value for an actual search on the landscape.

7 Conclusions

The purpose of this paper is to introduce a statistical analysis that can be used to measure and express the correlation structure of fitness landscapes. It is based on estimating the autocorrelations of a time series of fitness values (as introduced in (Weinberger, 1990)) but has several additional advantages:

1. The analysis makes use of a well known and understood time series analysis known as the Box-

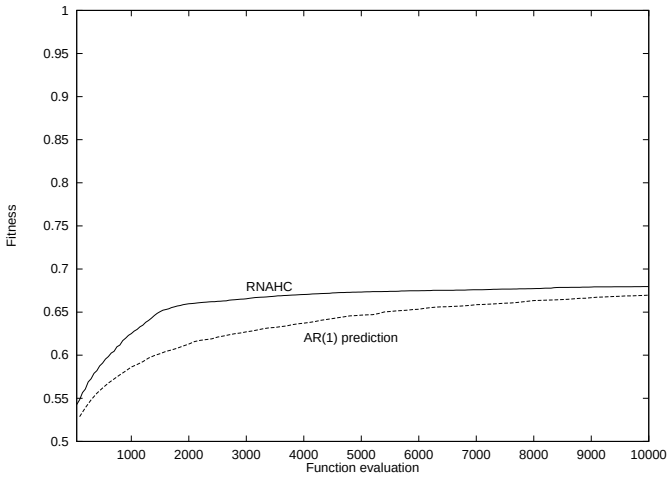

Figure 16: Comparison of an actual search result (RNAHC) with the prediction using the model that resulted from the landscape analysis (AR(1) prediction) on a fitness landscape generated with the NK-model with N=100 and K=25 and random interactions.

Jenkins approach. This makes it easy to apply the analysis and interpret the results, and gives more information than just the autocorrelations.

2. The analysis provides a complete (stochastic) model that can be used to make quantitative predictions or do simulations, something not possible with only the autocorrelations.

3. The model that results from using the analysis indicates the amount of memory in the process that generated the data, again something that can not be derived from the autocorrelations alone.

The analysis is demonstrated here on fitness landscapes that are constructed with the NK-model. Since NK-landscapes are fairly well understood, the results from applying the analysis are not that surprising. The correlation structure of most NK-landscapes can be expressed by an AR(1) model (as was already suggested in (Weinberger, 1990), and is validated here), but in some cases the appropriate model is just a white noise model or an AR(2) model, the latter indicating that there is more memory in the data generating process. This is something that could not be derived from looking at the autocorrelations only.

An interesting observation that follows from this is: Since there is more memory in the process, it should be possible to improve the search algorithm using this additional memory, for example by calculating an expected fitness increase at each step, and go to a next point that has a fitness increase close to the expected one. This way, the algorithm can take advantage of the correlation structure of the landscape as expressed by the given model. This points to one particular way of how the introduced analysis can help in designing better evolutionary algorithms.

It is also shown here that the analysis can be applied with a combination of search operators, which is more interesting when the correlation structure of a fitness landscape is coupled to the performance of some evolutionary algorithm that searches the landscape (which is the eventual goal, as stated in the quote at the beginning of this paper). The analysis is not restricted to single operators, but can be applied using any combination of well defined search operators.

Furthermore, one particular example of how the models that result from the analysis can be used to make quantitative predictions about search on the landscape is shown. This makes the analysis a useful one, and gives again an advantage over the correlation analyses used so far. Further work is required to show how general this predictive power is for other evolutionary algorithms.

Finally, in (Hordijk, 1996) the landscape analysis is applied to another, less well behaved and less well understood fitness landscape, resulting from searching for a cellular automaton that can perform a certain non-trivial computation. This landscape turns out to be highly anisotropic, and the analysis is applied to subspaces of the landscape. The analysis still gives consistent results in these cases, showing the generality of the method.

Acknowledgments

This paper resulted from a M.Sc. thesis done at the Erasmus University in Rotterdam, The Netherlands (Hordijk, 1994), under the supervision of Bernard Manderick. The paper itself was completed at the Santa Fe Institute. This work is supported by the National Science Foundation (grant NSF IRI-9320200) and the Department of Energy (grant DE-FG03-94ER25231). The author would like to thank Melanie Mitchell and Jim Crutchfield for invaluable support and for providing the opportunity to do this work. Furthermore, helpful conversations with Terry Jones and Peter Stadler are gratefully acknowledged.

References

Box, G. E. P. and Jenkins, G. M. (1970). *Time Series Analysis, Forecasting and Control.* Holden Day.

Forrest, S. and Mitchell, M. (1993). Relative Building-Block Fitness and the Building-Block Hypothesis. In Whitley, D., editor, *Foundations of Genetic Algorithms 2*, pages 109–126. Morgan Kaufmann.

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

Granger, C. W. J. and Newbold, P. (1986). *Forecasting Economic Time Series.* Academic Press, 2nd edition.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems.* MIT Press, 2nd edition.

Hordijk, W. (1994). Population Flow on Fitness Landscapes. Master's thesis, Erasmus University Rotterdam, The Netherlands. Available via http://www.santafe.edu/~wim.

Hordijk, W. (1996). Correlation Analysis of the Synchronizing-CA Landscape. In *Proceedings of the CNLS conference on Landscape Paradigms in Physics and Biology.* In press.

Jones, T. C. (1995). *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, University of New Mexico, Albuquerque NM, USA.

Judge, G. G. et al. (1988). *Introduction to the Theory and Practice of Econometrics.* John Wiley & Sons, 2nd edition.

Kauffman, S. A. (1989). Adaptation on Rugged Fitness Landscapes. In Stein, D., editor, *Lectures in the Sciences of Complexity*, pages 527–618. Addison-Wesley.

Kauffman, S. A. (1993). *Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press.

Lipsitch, M. (1991). Adaptation on Rugged Landscapes Generated by Iterated Local Interactions of Neighboring Genes. In Belew, R. K. and Booker, L. B., editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 128–135. Morgan Kaufmann.

Manderick, B., Weger, M. d., and Spiessens, P. (1991). The Genetic Algorithm and the Structure of the Fitness Landscape. In Belew, R. K. and Booker, L. B., editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150. Morgan Kaufmann.

Mitchell, M., Forrest, S., and Holland, J. H. (1992). The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance. In Varela, F. J. and Bourgine, P., editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 245–254. MIT Press.

Stadler, P. F. (1995). Towards a theory of landscapes. In Lopés-Peña, R., Capovilla, R., García-Pelayo, R., Waelbroeck, H., and Zertuche, F., editors, *Complex Systems and Binary Networks*. Springer Verlag.

Weinberger, E. D. (1990). Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference. *Biological Cybernetics*, (63):325–336.

Wilson, S. W. (1991). GA-Easy Does Not Imply Steepest-Ascent Optimizable. In Belew, R. K. and Booker, L. B., editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 85–89. Morgan Kaufmann.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. In Jones, D., editor, *Proceedings of the Sixth International Congress on Genetics*, volume 1, pages 356–366.