On the Correct Characterization of Fitness Landscape Neighborhood Topologies

Garrison W. Greenwood

Department of Electrical & Computer Engineering

Portland State University

Portland, OR 97207

Abstract

Knowledge about the fitness landscape structure aids in the design of effective search operators used in algorithms that solve difficult combinatorial optimization problems. Localized regions (i.e., neighborhoods) provide valuable structural clues.

Some researchers have claimed that the search operator "induces" the fitness landscape and therefore must be defined before any characterization of a landscape's structure is possible. In this paper we argue that such claims are flawed and can produce erroneous structural information. Formal proofs show that only a metric space interpretation of a fitness landscape can produce a correct characterization.

Keywords: fitness landscape, optimization, stochastic search

1 Introduction

There is an ongoing interest in the design of efficient algorithms capable of finding good solutions to difficult combinatorial optimization problems. Implicit is the idea that each solution has an associated real number value that reflects the quality of that solution. Any algorithm that "solves" an optimization problem must search the solution set for that particular solution with the highest quality value. Put another way, optimization problems are basically search problems.

Sewall Wright [24] introduced the concept of a *fitness landscape* to capture the underlying dynamics of evolutionary search. This landscape formulates the solution set as a collection of genotypes arranged in an abstract *genotype space*. The arrangement makes two genotypes adjacent if they differ by only a single mutation[1]. Each genotype is also assigned a real number, which indicates a fitness value. More formally, a fitness landscape consists of

- a large (albeit finite) set of solutions S
- a fitness function $f : S \rightarrow \Re_+$ (the positive real number line)
- the concept of a neighborhood between solutions

In the context of combinatorial optimization problems each genotype encodes all of the problem parameters needed to describe a solution. The solution constructed from the genotype is the phenotype and the fitness value indicates the solution quality. Searching for a solution is thus equivalent to exploring a fitness landscape for the highest peak. In the rest of this paper we use the terms genotype space, search space and solution set interchangeably.

The neighborhood concept allows us to interpret S as a vertex set of a graph Γ, where Γ is the solution set of a fitness landscape f [21]. Our definition of a neighborhood is similar to that used in metric space—i.e., the neighborhood of a genotype g_0 consists of all other genotypes within an open ball of radius δ centered at g_0. Of course, this implies some definition of "distance" in the fitness landscape[2].

The importance of a neighborhood cannot be understated. Many real-world optimization problems are NP-complete, which makes their genotype space exponentially large—exhaustive search is not practical. It therefore becomes paramount that the search operator be capable of exploring the landscape with minimal computational effort. The design of such an operator becomes more likely if the topology of the landscape is known, and it is the neighborhoods that provide that knowledge. For example, consider an arbitrary genotype g_0; its neighborhood consists of all genotypes within a small region encompassing g_0. If the fitness of these neighboring genotypes differs markedly from that of g_0, then the landscape is rugged—indicating a large number of local optima. Hence, the search operator should only make small moves over the landscape. Conversely, if fitness in the neighborhood does not differ much from that of g_0, then the landscape is smooth and large moves can be made with little chance of missing the global optima.

It is important to stress that the above definition of a neighborhood does not identify the search operator used by an algorithm to explore the fitness landscape. In other words, neighborhoods are defined solely by the genomic encoding; a genotype's neighbors are those genotypes that differ by only a bounded mutation, without regard to the underlying mechanism that caused the mutation. This interpretation is widely accepted by a large number of researchers—it is a canonical definition. Many researchers have used this definition with

[1] In biological terms, an alternative form of a gene is called an *allele*. Adjacent genotypes thus have only one allele that is different.

[2] This definition will usually be problem dependent.

abstract binary sequence spaces [4, 11, 13, 16, 17], while others have used it with more sophisticated landscape models [1, 2]. Nevertheless, some researchers have promoted the idea that a genotype alone cannot define the neighborhood—a search operator must be predefined because it somehow "induces" a neighborhood structure. (A partial list of recent examples espousing this idea includes [9, 18, 22].) Unfortunately, such notions are ill conceived and the purpose of this paper is to show why this is so. Specifically, we argue that using a search operator to help define neighborhoods or even the fitness landscape itself can lead to inconsistent and even contradictory results. We provide several examples to substantiate our argument.

2 Background

We begin by qualitatively characterizing a rugged fitness landscape. A landscape is considered "smooth" in the region surrounding a genotype if its neighboring points—i.e., genotypes with nearly identical alleles—differ in fitness by only a small amount. Conversely, a landscape is "rugged" if its neighboring points differ markedly in fitness. A promising statistical approach to characterizing landscapes was advanced by Weinberger [23] who suggested using a random walk to gather statistical information. Starting at some randomly chosen genotype g_t the walk next visits a randomly chosen neighbor. Repeating this process yields a sequence of fitness values $f_t, f_{t+1}, \ldots$. Weinberger assumed that since there is some underlying distribution of fitness values, a random walk in any direction is sufficient to gather statistics. The degree of correlation between two genotypes s steps apart in this random walk is given by the correlation function

$$R(s) = \frac{\langle f_t f_{t+s} \rangle - \langle f_t \rangle^2}{\sigma_f^2} \tag{1}$$

where $\langle \cdot \rangle$ means the expected value over all pairs s steps apart. If a high degree of correlation exists, then the landscape is smooth. Highly uncorrelated landscapes have a large number of local optima and any adaptive walk (i.e., a walk restricted to fitter neighbors) is likely to stop very quickly Kauffman [12]. These landscapes are presumed to be *statistically isotropic*. In other words, independent of where the random walk begins, the statistical information is invariant; a sufficiently long walk will infer any correlation present in the landscape. We have previously shown that isotropy does not hold for the class of constrained optimization problems where constraints determine if a genotype is feasible [5]. Nevertheless, the method of conducting walks is powerful and is instrumental to our discussion.

A mischaracterization of the true landscape topology can occur if one relies on a search operator to induce that landscape. Visualizing a genotype space as a metric space can show this. Some basic terminology of metric spaces is given below. More detailed information can be found in a variety of texts (e.g., see Mendelson

[15]).

A metric space is a set of points and a distance metric that indicates the closeness of pairs of points. Let X be a finite, non-empty set of points and define $\rho : X \times X \rightarrow \Re_{+}$.

Definition 1. The pair (X, ρ) is called a *metric space* provided that for all $x, y, z \in X$, the following properties:

(i) $\rho(x,y) \geq 0$

(ii) $\rho(x,y) = 0$ iff $x = y$

(iii) $\rho(x,y) = \rho(y,x)$

(iv) $\rho(x,z) \leq \rho(x,y) + \rho(y,z)$

ρ is a *metric* that quantifies the distance between two points in X. The last requirement above represents the well-known triangular inequality. The next two definitions describe special subsets of X.

Definition 2. Let (X, ρ) be a metric space. For any point $z \in X$ and any real number $\delta > 0$, let

$\mathrm{B}(z;\delta) = x : \rho(z,x) < \delta$

Then $\mathrm{B}(z;\delta)$ is called an *open ball* with center z and radius δ.

Definition 3. Let (X, ρ) be a metric space and let $a \in X$. A subset N of X is a *neighborhood* of a if there is a $\delta > 0$ such that

$\mathrm{B}(a;\delta) \subset N$

Definition 2 states that $\mathrm{B}(z;\delta)$ contains the set of points in X that are within a distance δ of the point z, while Definition 3 points out that a neighborhood contains all points of X that are sufficiently close to a. In particular, for every $\delta > 0$, $\mathrm{B}(a;\delta)$ is a neighborhood of a.

A natural distance measure to use with binary strings is the *hamming distance*, which measures the number of bit positions in which the two n-bit strings differ. It is easy to verify that all conditions from Definition 1 are satisfied; the genotype space comprised of n-bit binary strings can therefore be interpreted as a metric space. With $\delta = 2$, the open ball centered at the binary string x_0 is the neighborhood of x_0, and it includes all $x_k | \rho(x_0, x_k) = 1$ because any such x_k differs by only a single mutation. This is completely consistent with the canonical definition of neighborhood. Of course larger neighborhoods can exist by making $\delta > 2$.

3 Discussion

Let G denote the genotype space corresponding to an optimization problem. The genotype representation we consider is the n-bit binary string, i.e., $x = \{0, 1\}^n \in G$. Our goal is to find a value of x that minimizes an objective function $f(x)$. The genotype space G consists of all x that can be encoded as an n-bit binary string.

A concrete example of how search operators induce inconsistent neighborhoods definitions comes from a popular stochastic search algorithm: the *genetic algorithm* (GA). The GA is a probabilistic search algorithm that maintains a population of individuals where each individual represents a unique solution. During each iteration, stochastic reproduction operators create new solutions, which are evaluated. This evolutionary process continues until a termination condition is satisfied. To fully appreciate the inconsistency of the neighborhood definition, a more detailed explanation of the GA is needed. The GA is typically implemented as follows:

1. The optimization problem is formulated and the objective function used to evaluate each potential solution—i.e., determine the "fitness" of that solution—is defined.

2. A population of candidate solutions is randomly initialized. Each candidate solution encodes the problem parameters, which uniquely describe that solution. Holland [8] suggested that a binary string be used as a data structure, where the number of bits depends on the degree of desired resolution.

3. Each candidate solution is decoded and the problem parameter values are provided to the objective function that renders a fitness value.

4. Each individual i is assigned a reproduction probability p_i proportional to its relative fitness with respect to the current population of individuals.

5. Parents are selected for reproduction according to the reproduction probabilities p_i. Offspring—i.e., new candidate solutions—are created by operators such as bit mutation or crossover (see below). This process is repeated until a new population is created.

6. The process is halted if sufficient computation time has been expended or if a sufficient quality solution has been found. Otherwise, go to step (3).

GAs attempt to emulate the evolutionary dynamics found in Nature. These dynamics are influenced by five primary forces: natural selection, mutation, gene flow, genetic drift, and non-random mating. The probabilistic selection method in GAs emulates natural selection and to a certain extent the non-random mating forces. The random initialization of the first population emulates gene flow. Genetic drift is essential for a GA to converge. This latter force is a key element of our argument and so it is worth discussing in more detail.

Genetic drift refers to the loss of genetic variation in a finite population due to random sampling errors. Its effects are inversely proportional to the population size—i.e., genetic drift occurs sooner in smaller populations. This loss in genetic variation has no particular direction, which means there is no way of predicting which genes will be lost. The only thing that is known with certainty is, over time, genetic drift will change allele frequencies within the population even if no other evolutionary forces are present.

In terms of GAs, this means eventually the same bits are fixed in every member of the population. Put another way, ultimately some alleles are entirely removed from the population. This phenomena is regularly observed in GAs (see [19] for a thorough analysis). It is also the reason why search operators cannot induce fitness landscapes.

The most widespread reproduction operator used in GAs is the crossover operator, which combines portions of two parents to produce an offspring. The simplest version is the one-point crossover operator. Two parents are randomly chosen (according to p_i) and the same bit position is randomly chosen in each. The segments are then swapped to produce two offspring. This process is shown in Figure 1.

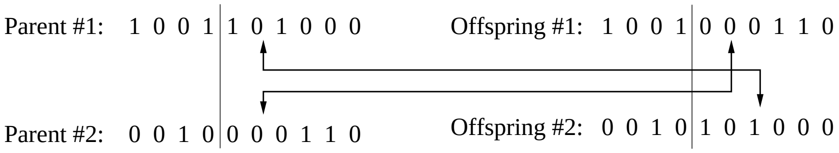
Parent #1: 1 0 0 1 | 1 0 1 0 0 0
Offspring #1: 1 0 0 1 | 0 0 0 1 1 0
Parent #2: 0 0 1 0 | 0 0 0 1 1 0
Offspring #2: 0 0 1 0 | 1 0 1 0 0 0

Figure 1: An example of one-point crossover used in a GA. The vertical line shows the crossover point.

The most general form of crossover is m-point crossover. Here m identical bit positions are chosen in both parents. Suppose we label these positions $\{r_1, r_2, ..., r_m\}$. Then the offspring copies all bits from position 1 up to position r_1 from the first parent, all bits from position r_1 to r_2 from the second parent, all bits from position r_2 to r_3 from the first parent, and so on. In other words, alternating segments are taken from each parent to form the offspring.

Without loss in generality, we consider a simple GA where only the one-point crossover operator is used for reproduction. No form of mutation is used. Each solution x in the search space S is encoded as a fixed length binary string, so unique solutions have unique binary patterns. The GA runs for a large, but finite, number of generations.

But there is a problem. At any given iteration t, the GA is evolving a finite population $P(t)$ where $|P(t)| << |S|$. Individuals are selected from the previous population ($P(t-1)$) with a probability directly proportional to their fitness—i.e., highly fit individuals are chosen with a greater probability. Genetic drift will occur because the population size is finite, which means after running a GA for a period of time, every individual in the population will have the same one or more bits fixed at the same value.

The inescapable consequences of genetic drift helps to explain why one should not consider fitness landscapes to be induced by a search operator. If the GA uses only crossover for reproduction—i.e., no mutations are permitted—then no matter how many times crossover is subsequently applied, these bits will never change value. All fitness landscapes now become time dependent because the effects of genetic drift change gene frequencies alleles and not all search operators can restore lost alleles. For instance, if the left-most bit of every individual in the population is fixed at 0, then an infinite number of crossover operations will never produce an offspring with the left-most bit set to 1. *In effect, every genotype $g \in S$ of the form $1\#\#\#\#\cdots\#$ has disappeared from the search space*[3].

In reality these genotypes still do exist even though the search operator cannot visit them. Over time genetic drift causes the search space of a GA to shrink. This can be illustrated with the sequence

$$S_0 \supseteq S_1 \supseteq S_2 \supseteq \cdots \supseteq S_{k-1} \supset S_k \supseteq S_{k+1} \supseteq \cdots \supset S_m \supseteq \cdots \supset S_N = S_{N+1} = \cdots \qquad (2)$$

where S_t is the search space at generation t. Eventually at some generation k only a proper subset is formed because genetic drift has eliminated some of the alleles. Subsequent generations are subsets until some generation m where again genetic drift removes more alleles. This process continues until generation N where the search space contains only a single genotype—i.e., the GA's population has converged and is now homogeneous. However, the situation is completely different if the search operator allows gene flow. In this case no shrinkage of the search space occurs and the evolving sequence is

$$S_i \ = S_0 \ \ \forall\ i \qquad (3)$$

The inability of some search operators to compensate for genetic drift effects is fundamental to our argument that search operators do not induce landscapes that can be correctly characterized. This is stated formally by the following:

Theorem 1 *Let S be a search space of a GA. Then the topology of any $S' \subseteq S$ is completely independent of any search operator that moves through S'.*

Proof 1 *Let $\mathcal{O}$ be the set of search operators that move through S'. This set can be partitioned as* $\mathcal{O} =$

[3] '#' means either a 0 or a 1

$\mathcal{O}_g \cup \mathcal{O}_{ng}$ where O_g (O_{ng}) is the subset of operators that does (does not) introduce gene flow.

If $O_i \in O_{ng}$, then Eq. (2) shows how S' changes as the GA processes generations. But this effect is caused by genetic drift, which no $O_i \in O_{ng}$ can compensate for. Furthermore, repeated runs of the GA will not produce the exact same effect because genetic drift has no specific direction; the topology change is therefore independent of the search operator choice. Eq. (3) holds for every operator in O_g, which makes the topology independent of any operator in O_g. Thus, in either case, the topology is independent of the search operator.

Theorem 2 *A correct topological characterization of $S' \subseteq S$ can be computed from a metric space interpretation of S.*

Proof 2 *A metric space interpretation of S contains all possible genotypes so a statistical analysis such as given in Eq. (1) will be accurate. Furthermore, the interpretation does not require identifying a search operator. The proof follows from Theorem 1.*

Genetic drift ultimately leads to irreversible alterations in the neighborhood structure. The believe that search operators induce landscapes gives a distorted view of a neighborhood that is more than just a mild inconvenience. In fact, the inability of a search operator to visit some points introduces real difficulties into the quest for an optimal solution. Specifically,

- The inability to visit some genotypes within a neighborhood makes that neighborhood incomplete in the sense that some genotypes have, in effect, been removed from the search space—perhaps even including the globally optimal one. Nevertheless, these genotypes do actually exist.

- Proofs exist that show stochastic search algorithms—such as evolutionary programs—will converge to the global optimum if given enough running time [7]. However, these proofs presume the search operator used is ergodic. Incomplete neighborhoods destroy any pretense of ergodicity, thereby negating these proofs.

- Statistical analysis of the landscape can serve as a basis of choosing suitable search operators [12, 14]. The removal of some genotypes alters the topology of the neighborhood, which yields erroneous statistical values. Any new search operators chosen from this incorrect correlation value may not perform as expected.

4 An Example

In particular we want to expand on this last point and illustrate how incomplete neighborhoods can mischaracterize a landscape neighborhood. Consider the following decision problem:

> *Let $I = \{i_1, ..., i_n\}$ be a randomly chosen set of positive integers from the interval* [1,1000] *and choose another positive integer called the goal. Are there five distinct elements in I that will total exactly to the goal?*

This problem is an instance of the *subset sum problem*, which is NP-complete [3]. Each solution is an integer array containing five elements from I and a 1-mutant neighbor has only a single element incremented or decremented by 1. We consider a solution to be infeasible if either (a) the total exceeds the goal or (b) the solution has a number i such that $i \in [1, 1000]$ but $i \notin I$. The fitness of a feasible solution equals the total of the five numbers; the fitness of an infeasible solution is set to -500. (We used 2500 for the goal.) Observe that the fitness for feasible solutions is strictly positive while infeasible solutions form "sinkholes" of extremely low fitness in the landscape. These sinkholes are intrinsic to fitness landscapes corresponding to constrained optimization problems [5].

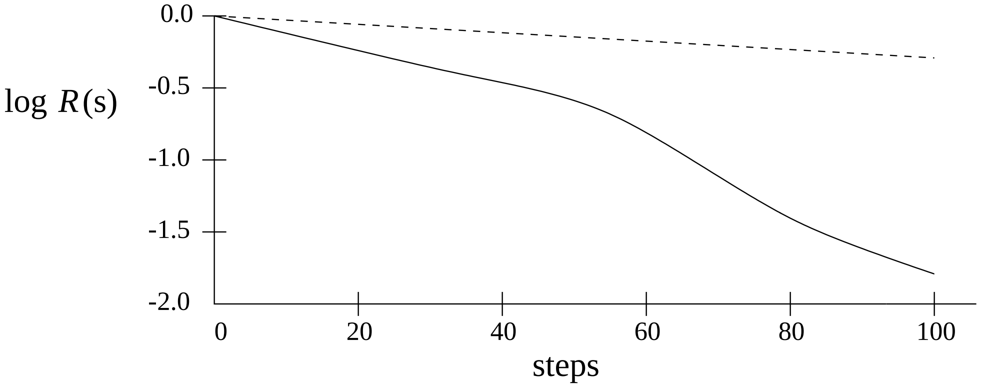

Figure 2: Correlation in the subset sum problem fitness landscape from random walks started at the same initial solution and restricted to an open ball of radius 20. The solid (dashed) line is from a 2000 step walk conducted with the O1 (O2) operator. Note that plot is semi-logarithmic.

We are going to characterize two landscapes for this problem: one is complete in the metric sense while the other has say the fifth position in the integer array fixed. (The second landscape is a subset of the first landscape.) Both landscapes are induced by search operators, but the second one is missing genotypes due to genetic drift. We are going to characterize the landscapes by gathering statistics from a long random walk on the landscapes. Denote O1 (O2) as the operator that walks between 1-mutant neighbors in the first (second) landscape. Both walks are 2000 steps long and begin at the same point. Furthermore, both walks are restricted to a neighborhood defined by an open ball of radius 20 centered at the start point. In order to make the comparison as fair as possible, we recorded the array position that was mutated to create each step for the O1 walk, and mutated the same position in the same way during the same step to create the

O2 walk[4]. Figure 2 shows the correlation obtained from one such walk. It is clear that the two operators do not perceive the same neighbor structure. The characterization by the O1 walk is more accurate because it traverses a complete neighborhood.

5 Final Remarks

Not everyone believes distance metrics are necessary. Stadler, et. al [20] studied folding of RNA sequences into secondary structures and concluded that phenotypes organized according to genetic accessibility produces a space that lacks a metric. They do admit combinational optimization problem solutions often fit into a regular structure such as a hypercube (where the notion of distance makes perfect sense), but they go on to claim the notion of "neighborhood" does not require a distance measure but instead relies on a genotype to phenotype mapping. However, they studied a only small set of similar biological problems and tried to broadly apply their findings to the evolutionary search for solutions to arbitrary combinatorial optimization problems. Their arguments about evolutionary searches of genotype spaces are flawed in three ways:

1. Neighborhoods in hypercubes are completely independent of the genotype to phenotype mapping. An example of a regular topological space suitable for any combinatorial problem using binary strings for a genotype—and many problems fall into this category—is Kauffman's NK landscape [12]. A number of methods for analyzing the topology of NK landscapes have been proposed, but none of them depend on any particular mapping. Indeed, in Kauffman's original work the phenotypes were randomly assigned.

2. Evolutionary algorithms (GAs, evolution strategies and evolutionary programming) are widely used to conduct evolutionary searches over the genotype space of combinatorial problems. Some evolutionary algorithms use panmictic recombination and tournament selection—even though no counterparts for these dynamics exist in Nature. This shows algorithm designers do not necessarily believe evolutionary searches must exactly mimic what Nature does.

3. Nature modifies RNA shapes by mutating an underlying sequence—i.e., shape information is indirectly encoded because the genotype contains sequence information instead of explicit shape information. Stadler, et. al argue evolution based on spaces does not require the structure inherent in metric spaces because "nearness" can be defined in terms of set relations. Evolutionary algorithms use genotypes to encode representations of problem solutions. However, indirect encoding schemes are extremely rare. Metric accessibility topologies are commonplace for combinatorial problems because distance measures are easily defined with direct encodings. Direct encodings are preferred for bioinformatics problems.

[4] Any mutations to the fifth position were ignored in the O2 walk.

> For example, evolutionary algorithms have been extensively used to search for solutions to protein folding problems (see [6] for an excellent survey of recent work). None of the genotypes used with these evolutionary algorithms encoded residue sequences; instead they directly encoded secondary and tertiary structure information such as backbone or side-chain torsion angles. Neighborhoods in these circumstances are defined in terms of a distance measure and not sequence similarity.

The key point here is no compelling motive exists for treating a search space as a plain set devoid of any structure. There is nothing wrong with considering a search space to be equivalent to a metric space. Indeed, Stadler et. al never demonstrated how (or if) a point set topology formalism, when compared against a metric space formalism, leads to improved designs for evolutionary search operators or improved performance in evolutionary searches. Search spaces for combinatorial problems—even those without binary strings for genotypes—fit quite naturally into a regular structure where neighborhoods are defined by distances. (A technique for embedding the search space of any combinatorial problem into a k-ary N-cube is described in [10].) However, Stadler et. al do agree with us in one regard: they state genotype spaces induced by certain types of crossover operators are inconsistent with metric distances.

It is reasonable to question if landscape characterization really helps to choose effective search operators. We believe, for finite populations, it is only worthwhile to use search operators that allow some gene flow. This is because operators that do not permit gene flow have no ability to compensate for the unavoidable genetic drift effects.

It is important to note that completely eliminating genetic drift effects is to be avoided. Indeed, without some genetic drift effects the search algorithm would never converge to the optimal solution. The goal is to maintain some balance between genetic drift and gene flow. That is, the search should have enough genetic drift to converge, but also enough gene flow to escape from local optima.

The conclusion one should draw from the above discussion is that search operators do not define neighborhoods within genotype space, but instead define reachability subspaces—i.e., subspaces containing points that can be visited by one execution of the search operator. We have shown these reachability subspaces do not always fit the canonical definition of a neighborhood, which introduces a number of problems including mischaracterization of the landscape's topology. Conversely, defining neighborhoods solely in terms of the genetic encoding assures all genotypes are present—which retains the global optimum—and provides the necessary framework for search algorithm ergodicity. The inability of a search operator to visit specific points says a lot more about the efficacy of the operator than it does about the landscape's topology.

References

[1] M. Conrad and W. Ebeling. Evolutionary thinking and the structure of fitness landscapes. *Biosys.*, 27:125–128, 1992.

[2] W. Fontana, D. Konings, P. Stadler, and P. Schuster. Statistics of RNA secondary structure. *Biopolymers*, 33:1389–1404, 1993.

[3] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W.H. Freeman, 1979.

[4] S. Govindarajan and R. Goldstein. The foldability landscape of model proteins. *Biopolymers*, 42:427–438, 1997.

[5] G. Greenwood and X. Hu. Are landscapes for constrained optimization problems statistically isotropic? *Physica Scripta*, 57:321–323, 1998.

[6] G. W. Greenwood and J. M. Shin. On the evolutionary search for solutions to the protein folding problem. *in Evolutionary Computation in Bioinformatics*, G. Fogel and D. Corne (Eds.), Morgan-Kauffman, 2002.

[7] G. Greenwoood and Q. Zhu. Convergence in evolutionary programs with self-adaptation. *Evol. Comput.*, 9(2):147–158, 2001.

[8] J. Holland. *Adaptation in Natural and Artificial Systems.* Univ. of Michigan Press, 1975.

[9] W. Hordijk. A measure of landscapes. *Evolutionary Computation*, 4(4):335–360, 1996.

[10] X. Hu, G. Greenwood, and S. Ravichandran. Modeling epistatic interactions in fitness landscapes. *Proc. Congress on Evol. Comput.*, pages 932–938, 2000.

[11] T. Jongeling. Self-organization and competition in evolution: a conceptual problem in the use of fitness landscapes. *J. Theo. Biol.*, 178:369–373, 1996.

[12] S. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford Univ. Press, 1993.

[13] W. MacReady. Tailoring mutation to landscape properties. *Proc. Evolutionary Prog. VIII*, pages 355–364, 1998.

[14] B. Manderick and P. Spiessens. How to select genetic operators for combinatorial optimization problems by analyzing their fitness landscape, in. *Computational Intelligence: Imitating Life*, J. Zurada et. al (Eds.), IEEE Press:170–181, 1994.

[15] B. Mendelson. *Introduction to Topology.* Dover Publ., 3rd edition, 1990.

[16] K. Niklas. Morphological evolution through complex domains of fitness. *Proc. Natl. Acad. Sci.*, 91:6772–6779, 1994.

[17] A. Perelson and C. Macken. Protein evolution on partially correlated landscapes. *Proc. Natl. Acad. Sci.*, 92:9657–9661, 1995.

[18] C. Reeves and C. Höhn. Modern heuristic search methods, in. pages 99–110, 1996.

[19] A. Rogers and A. Prügel-Bennett. Genetic drift in genetic algorithm selection schemes. *IEEE Trans. Evolutionary Comput.*, 3(4):298–303, 1999.

[20] B. Stadler and P. Stadler. Generalized topological spaces in evolutionary theory and combinatorial chemistry. *J. Chem. Inf. Comput. Sci.*, 42:577–585, 2002.

[21] P. Stadler. Fitness landscapes arising from the sequence structure maps of biopolymers. *Theochem*, 463:7–20, 1999.

[22] V. Vassilev, T. Fogarty, and J. Miller. Information characteristics and the structure of landscapes. *Evolutionary Computation*, 8:31–60, 2000.

[23] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *J. Biol. Cyber.*, 63:325–336, 1990.

[24] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. *Proc. 6th Int'l Congress on Genetics*, page 356, 1932.