A Model Of Landscapes

Terry Jones *

February 1, 1994

Abstract

The use of the term "landscape" is increasing rapidly in the field of evolutionary computation, yet in many cases it remains poorly, if at all, defined. This situation has perhaps developed because everyone grasps the imagery immediately, and the questions that would be asked of a less evocative term do not get asked. This paper presents a model of landscapes that is general enough to encompass most of what computer scientists would call search, though the model is not restricted to either the field or the viewpoint. It is particularly relevant to algorithms that employ some form of crossover, and hence to genetic algorithms and other members of the evolutionary computing family. An overview of the consequences and properties of the model establishes a connection with more traditional search algorithms from artificial intelligence, introduces the notion of a crossover landscape, and argues the importance of viewing search as navigation and structure.

*Santa Fe Institute, 1660 Old Pecos Trail, Suite A., Santa Fe, NM 87505, USA and Department of Computer Science, University of New Mexico, Albuquerque NM 87131, USA. Email: terry@santafe.edu

1 Introduction

The field of Artificial Intelligence (AI) has long recognized the importance of structure in search. Most of the classic AI search algorithms, see, for example, [6, 21, 30, 40], are either explicitly phrased as algorithms that search graphs, or can be described in terms of a search on a graph. The other component of search is navigation: the process of examining a structure for some purpose. Myriad navigation strategies, from random search to exhaustive search, have been developed. Some are designed to search specific structures, others can be more generally applied.

The importance of structure in evolutionary algorithms and in fields other than biology is beginning to be recognized, and the structures are commonly termed "landscapes". These fields include anthropology [20], chemistry [5, 32], economics [15, 18, 27], immunology [14, 22], physics [1, 37, 38] and computer science [3, 11, 12, 17, 23, 24, 26, 35]. The above references represent merely the tip of an iceberg and should not be taken as complete. The study of RNA landscapes for instance has produced many papers that adopt a landscape perspective.

The use of a landscape metaphor to develop insight or intuition about the workings of a complex process originated with the work of Sewall Wright in the early 1930s [41]. Wright used landscapes to provide imagery for his theory of speciation. According to Eldredge [4], Wright's fitness surface is:

> By all odds the most important metaphor in macroevolutionary theory of the past fifty years.

We develop a landscape model that, structurally, is simply a labeled directed graph. This provides a fundamental link to traditional AI search and also a framework for thinking about evolutionary algorithms in terms of structure and navigation. This separation allows us to examine the components of the algorithms individually, and to apply simple navigation strategies from AI to the landscape structures induced by the so–called genetic operators in evolutionary algorithms. The model is especially relevant to the "evolutionary" algorithms that employ some form of crossover. This includes genetic algorithms [7, 9], genetic programming [19] and evolution strategies [2, 29, 33].

A later section details connections with other versions of landscapes. This discussion is deferred as it cannot be properly understood before the model is presented.

2 Landscapes

In its most general form, a landscape, $\mathcal{L}$, as proposed here, is simply a directed graph, $\mathcal{G}_\phi$, whose vertices, V, are labeled with a real number and whose edges, E, are labeled with a probability. The sum of the probabilities on the edges leaving any vertex is 1. Any such graph shall be called a landscape, and anything hereafter referred to as a landscape is simply such a graph. The subscript, ϕ, will be explained below.

It is important to clearly understand the origin of the various components of a landscape. Initially, we are faced with some situation whose objects present some (possibly infinite) set.

This is deliberately vague, so as to avoid terms like "problem" and "solution". As examples, we might be interested, for whatever reason, in the set of n–tuples of real numbers, the set of binary strings of length n, the set of LISP S–expressions, the set of permutations of the integers one to ten, the set of legal chess positions, the set of spin configurations in a spin glass or the set of RNA sequences.

We will call the set of objects described above the *object space*, $\mathcal{O}$. If we wish to use some form of computation to address $\mathcal{O}$, we construct a representation of the objects in it. This representation determines a *representation space*, $\mathcal{R}$. There may be a bijection between $\mathcal{O}$ and $\mathcal{R}$, but $\mathcal{R}$ may also correspond to a proper subset of $\mathcal{O}$ or may include the whole of $\mathcal{O}$ as a proper subset[1].

Due to finite limits on computation, some object spaces, being infinite, will necessarily be under–represented by the chosen $\mathcal{R}$. An obvious and common example is object spaces that involve real numbers. The computational process is restricted to dealing with that part of $\mathcal{O}$ that is representable with the choice of $\mathcal{R}$. There is of course nothing to stop the process from adopting a new $\mathcal{R}$ at any point, for example see [25, 31, 34, 39].

The objects in $\mathcal{O}$ are of interest for some reason or reasons, and we will suppose that the degree to which an object is interesting or desirable can be expressed as a single real value. This is commonly called the *fitness* of the object in question. We will denote the fitness of an object $a \in \mathcal{R}$ by $f(a)$ and call $f : \mathcal{R} \rightarrow \Re$ the *fitness function.* For the purposes of this paper, we will assume that numerically higher fitness is desirable, though the opposite view is also common. In practice, f may be difficult to compute, and may be the result of some heuristic calculation, as is common in many problems in AI.

At this point we have a representation space and a function that assigns values to points in that space. Nothing more. In particular, we do not yet have a landscape. The entire reason, presumably, for talking about a landscape is to use the powerful imagery it evokes. One is concerned with "peaks", "ridges" and "valleys" and the like. But what is a peak? A simple definition of a peak is a point whose neighbors are all "below" (or less fit) than it is. Thus a *sine qua non* of any discussion of landscapes is some notion of neighborhood. Without neighborhood, none of the terminology associated with landscapes makes any sense at all.

Neighborhood in our landscapes will be determined by the choice of an *operator*, ϕ. This is not an operator in the mathematical sense, the term is chosen as it is already so widely used in the field of evolutionary computation. An operator is better thought of as a stochastic process which, when given some vertex v in $\mathcal{G}_{\phi}$ as input, produces one of a set of possible outcomes, each of which is a vertex in $\mathcal{G}_{\phi}$, and each with a certain probability.

The (possibly infinite) set of outcomes (vertices) that the operator may produce on input v will be called the *neighborhood* of v under ϕ, and will be denoted by $N_{\phi}(v)$. If ϕ produces w from input v at time t, we will write $\phi_t(v) = w$. It will be simplest to assume for now that $N_{\phi}(v)$, the set of possible outcomes, does not change with time and that the probability

[1] If we are not performing any computation, there is no need for a representation space and we can simply set $\mathcal{R} = \mathcal{O}$.

that $\phi_t(v) = w$ is also independent of time[2]. If we are simply concerned with the event of ϕ producing w from v, and not with when this happened, we will drop the time subscript and simply write $\phi(v) = w$

If $w \in N_{\phi}(v)$ then $\mathcal{G}_{\phi}$, will contain a directed edge (v, w) labeled with the probability that $\phi_t(v) = w$. Thus the edges of the graph are induced by the operator ϕ, as indicated by the subscript.

We have accounted for the edges of $\mathcal{G}_{\phi}$, but what about the vertices? The usual notion of a landscape would simply assign each member of $\mathcal{R}$ to a vertex in $\mathcal{G}_{\phi}$. But this would prevent us from defining landscapes for operators that act on or produce more than a single element of $\mathcal{R}$. We will make an important generalization. We assume that an operator acts on a k–tuple of points from $\mathcal{R}$ (i.e. an element of $\mathcal{R}^k$) and produces an l–tuple, a member of $\mathcal{R}^l$.

Vertices in $\mathcal{G}_{\phi}$ will then correspond to either a k–tuple or an l–tuple of points from $\mathcal{R}$. Often, as is the case in mutation, k and l will both be one, and the distinction will not be needed. But all forms of crossover act on more than a single point of $\mathcal{R}$ as input, and most produce more than a single point.

Before elaborating on the consequences of this generalization, we can attend to the last component of our landscape, the real numbers that label the vertices. In general, we need two functions, $f_k : \mathcal{R}^k \rightarrow \Re$ and $f_l : \mathcal{R}^l \rightarrow \Re$. These are used to assign a real value to each vertex.

If the operator that induces the connectivity in $\mathcal{G}_{\phi}$ acts on and produces a single point in $\mathcal{R}$, then it is natural to label each vertex $v \in V$ with the fitness, $f(v)$, of the point in $\mathcal{R}$ to which it corresponds. In other words, if $k = 1$ and $l = 1$, it is natural (though of course not necessary) to let $f_k = f$ and $f_l = f$. Where $k > 1$ or $l > 1$, one might choose the maximum of the fitnesses of the corresponding points of $\mathcal{R}$ [3], their average [23], or some other function.

2.1 Terminology And Conventions

A landscape will be called *walkable* if $k = l$. If a landscape is walkable, an operator's outcome can be used as its next input. For example, the landscapes generated by the common forms of mutation are always walkable. Any form of crossover that takes two parents and produces two children induces a landscape that is walkable. Crossover that produces one child from two parents induces a landscape that is not walkable. We will also refer to operators as being walkable, meaning that they generate walkable landscapes.

A landscape will be called *symmetric* if $w \in N_{\phi}(v) \Rightarrow v \in N_{\phi}(w)$ and $P(\phi(v) = w) = P(\phi(w) = v)$. In words, a landscape is symmetric if the existence of a directed edge between v and w implies that there is also an edge from w to v, and that, at any point in time, both edges are labeled with the same probability. We will also refer to operators as being

[2]These assumptions are violated by evolution strategies, evolutionary programming and simulated annealing. They can be discarded once the basic landscape notions are understood.

symmetric, meaning that they generate symmetric landscapes. Most common operators in evolutionary algorithms are symmetric. If an operator is symmetric it will be convenient to consider $\mathcal{G}_\phi$ as undirected. Notice that a symmetric landscape is necessarily walkable since the above condition implies that $k = l$.

If a landscape is symmetric, it will make sense to talk about connectedness. Two vertices v and w in V are connected in $\mathcal{G}_\phi$ if there is at least one path between them. If every vertex is considered connected to itself, the relation "connected to" on V is an equivalence relation and thus induces equivalence classes. Each of these equivalence classes, together with the edges incident on the vertices in the class, defines a *connected component* of $\mathcal{G}_\phi$. If $\mathcal{G}_\phi$ has a single connected component, the landscape will be said to be *connected*. A connected landscape is necessarily walkable, but the converse is not true.

If ϕ produces each member of v's neighborhood with equal probability, for all $v \in V$, we will usually not label the edges of $\mathcal{G}_\phi$.

A landscape will be called *natural* if it is symmetric, $k = 1$, $l = 1$, $f_k = f$ and $f_l = f$. A simple example is the mutation landscape where vertices of $\mathcal{G}_\phi$ are labeled with the fitnesses of the points in $\mathcal{R}$ to which they correspond.

2.2 Three Operator Classes And Their Landscapes

The three classes of operators most commonly encountered all rely on the objects in $\mathcal{R}$ being composed of some (not necessarily fixed) number of components. For example, a bit string has some number of bits, an n–tuple of real numbers consists of n individual reals and the parse tree representing a LISP S–expression consists of some number of internal nodes and leaves. These components can typically be modified individually or in subsets, possibly according to some constraints.

The simplest operator class we will consider could (narrowly) be called the bit–flipping operators[3]. These act on and produce a single element of $\mathcal{R}$. They randomly choose a component of their input and modify it in some slight way. The outcome of this operator always differs in exactly one component from the input. The simplest example of this sort of operator is the operator that flips a bit in a bit string. If $\mathcal{R}$ is $\{0,1\}^n$, the bit–flipping operator induces a graph that is a hypercube of dimension n. Each edge in the graph will be labeled with probability $1/n$.

The second class is the mutational operators. These operators also act on and produce a single element of $\mathcal{R}$. They modify each component of their input with a given, typically small, probability. Thus a mutational operator's output is a modification of its input according to some probability distribution. The outcome may be identical to the input, it may contain minor changes or may even be different from the input in every component. If we again consider binary strings of length n and assume that each bit is flipped with a fixed probability p, then the graph induced is the complete graph K_{2^n}. An edge in the graph between vertices that represent binary strings whose Hamming distance is d will be labeled with a probability

[3] This name is really only appropriate when $\mathcal{R}$ is $\{0,1\}^n$. In fact, operators of this class can be trivially constructed for any of the representations for which we can define mutational operators.

$p^d q^{n-d}$. Notice that this is a very different landscape to that generated by the conceptually similar bit–flipping operator.

The final class is the crossover operators. These typically act on an element of $\mathcal{R}^2$. Crossover operators combine the components from their inputs to produce their output. As an example, a crossover operator in an evolution strategy might receive two n–tuples of real numbers as input and its outcome might be a single n–tuple whose elements are the averages of the corresponding components of the inputs [2]. There are many forms of crossover for representation spaces such as n–tuples of reals, permutations of integers and fixed length strings over some alphabet.

3 Consequences Of The Model

The definition presented above has a number of consequences that are not found in other landscape models. The landscape does not need any notion of dimensionality, and it does not need a distance metric. In some cases of course, these will be present, and there is nothing to say that they should not be taken advantage of when they are. But the model does not require them, and so we can construct landscapes for Tic–Tac–Toe, sliding block puzzles, bit strings of length n, traveling salesperson tours and LISP S–expressions with equal ease. These two consequences are further discussed below in the section on advantages of the model.

Two properties of the current model will seem particularly strange. Firstly, as described above, a landscape may not be *walkable*. As mentioned, an example is any form of crossover that produces one child from two parents. Such an operator cannot be used to conduct a walk on the landscape as the output of the operator cannot be simply used as the next input.

Secondly, landscapes may not be connected. A simple example is the landscape induced by a (non–genetic programming) crossover operator that produces two children from two parents. Consider a vertex of the landscape that corresponds to two points of $\mathcal{R}$ that are identical. The vertex will be connected to itself (with probability 1), and nothing else. A more specific example of a landscape that is not connected is seen by considering the vertex $(011, 010)$ of V where $\mathcal{R} = \{0, 1\}^3$. Clearly no form of crossover can transform this input to a pair of points either of which begins with a one, e.g. $(100, 000)$. Equally clearly, no composition of crossovers will be able to do that either. Thus $(011, 010)$ is not connected to any vertex in the landscape which contains a member of $\mathcal{R}$ that starts with a 1 (in fact, this vertex is also connected only to itself). The complete landscape for one–point crossover on binary strings of length 3 is shown in Figure 1.

The possibility that landscapes may not be connected means that on these landscapes there is no general notion of distance between landscape vertices. This does not mean that a metric cannot be defined, just that there may be no natural one (such as the length of the shortest path between the vertices concerned). The model does not require a distance metric to exist, though the absence of one might make certain statistics meaningless or impossible to compute. Within a connected component of a landscape, one can always use the length of the shortest path between two vertices as a distance metric.

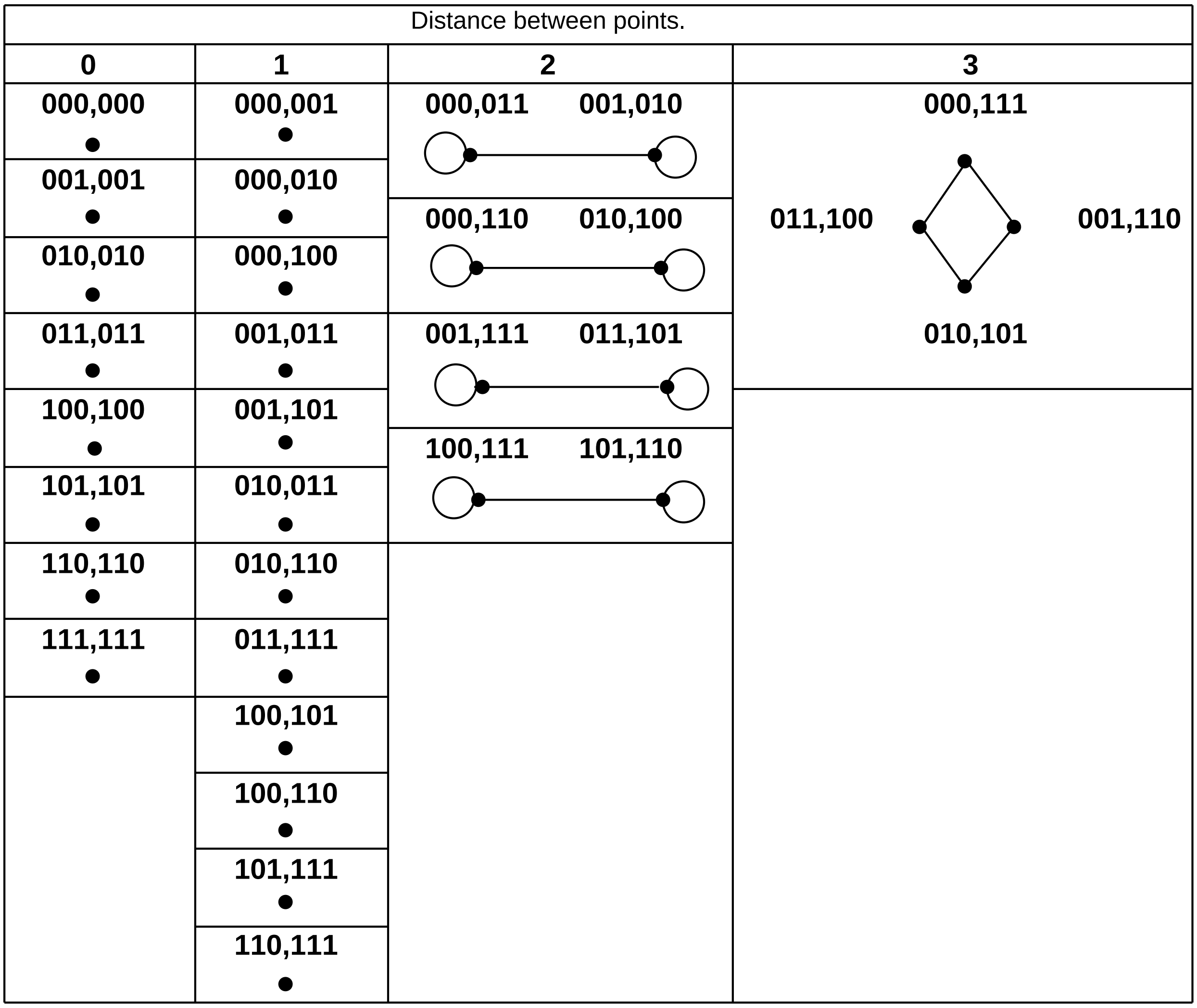

Figure 1: The one–point crossover landscape for binary strings of length 3. All edge probabilities are 1/2.

Because a landscape may not be walkable, general statistics describing properties of landscapes may be restricted to using the information gained from repeated single applications of the operator that generated the landscape. This sort of statistic was employed in [23], even though they were actually considering a walkable landscape.

4 Implications For Genetic Algorithms

An important consequence of this model of landscapes is that every operator used by an algorithm defines its own landscape. For this reason, genetic algorithms should be seen as operating on (at least) a pair of landscapes, the mutation landscape and the crossover landscape. These landscapes will have different characteristics for different problems, and can be studied independently.

When one hears people talk of "the landscape", the object of reference is most commonly the mutation landscape or the hypercube landscape that a bit–flipping operator induces. Crossover is typically employed far more frequently than mutation in a genetic algorithm, yet it has been assigned a supporting role in the landscape structure defined by mutation. To say that crossover is making large jumps on the landscape (the mutation landscape is implied), is to examine crossover outside its own context. What if the algorithm involved does not even use mutation? What relevance does the mutation landscape have then for the algorithm or for crossover? It makes as much sense to say that mutation is making large jumps on the crossover landscape. There is some sense to these statements, but it is worth considering each operator in light of the landscape structure it defines. This bias is a difficult thing to detect.

Another illustration can be found in the interpretation commonly given to the notion of a peak on a landscape. When people talk of a peak, they are referring to a peak on the mutation landscape. Yet crossover is an operator too, and a far more frequently used one. Surely it should be entitled to some notion of peak? The notion of a peak cannot be divorced from the notion of operator. A vertex in $\mathcal{G}_\phi$ can be a peak under one operator and not under another. The mutation landscape has peaks, the crossover landscape has peaks. To regard peaks as something defined only in terms of mutation is simply biased.

5 Advantages Of The Model

- The model establishes a point of contact with traditional AI search. The contact is at a fundamental level, that of the graph, and has practical consequences (see below).

- The model does not make the assumptions that prevent other notions of landscape from being more widely used. For example, the landscape does not need some fixed dimensionality, nor does it need a distance metric between the objects that compose the landscape. For these reasons, it is possible to view many AI problems as being problems on landscapes of this type. For instance, it is simple to imagine a landscape for chess or Rubik's cube. The navigational task on these landscapes may differ, and the operators may be complex, but the underlying structures are the same. The model is also useful within the field of evolutionary computing. It applies as well to genetic programming as to genetic algorithms. Statistics that can be calculated for a landscape in one paradigm can be calculated in exactly the same way for another. The model also provides a framework for thinking about hillclimbing, simulated annealing, evolution strategies and evolutionary programming.

- The model invites a point of view that seems uncommon in the field of evolutionary computation, though not in AI. This is a view of search as navigation and structure. Once this move is made and once the various components present in a genetic algorithm are identified, it becomes natural to ask questions about these components. If we regard many search algorithms as being navigation and structure, then it is possible to break algorithms into their pieces and collect the parts. From a collection of these pieces, we could choose (say) one navigation strategy and one structure and use them as the basis of a new algorithm. An obvious target here is to combine a hillclimbing strategy with a crossover landscape from genetic algorithms. This approach has been used with success on Holland's recent Royal Road "challenge" [10]. A simple hillclimber consistently climbs four of the five levels in that problem using approximately the same number of evaluations as a standard genetic algorithm [13]. The success of this approach raises important questions about the genetic algorithm as a navigation strategy, at least on this problem.

- The one operator, one landscape view reveals the very different landscapes that are constructed by various operators. This invites statistical analysis of the landscapes, as has been done in [17, 23, 24, 35, 38]. Such analysis has the advantage of being independent of any particular navigation strategy. For this reason, it may be possible to demonstrate that a particular operator creates difficult (in some sense) landscapes for some problem types. Such results might go a long way towards resolving the never–ending debates on the virtues of certain operators. Statistics such as these would be very useful as indicators of potential difficulty (or ease) of a problem for an operator and could simultaneously be a guide to the construction, at run time, of hybrid search algorithms.

6 Limitations Of The Model

As presented, the model assumes that the worth of objects in $\mathcal{O}$ can be represented as a single number. In many situations, it may be more natural to assign a set of values to a point in $\mathcal{O}$. As presented, the model assumes also that navigation strategies are content to navigate on the basis of a single value at each vertex of the landscape. This may not be the case, and is certainly not the case in pareto optimality. The model can be easily extended to remove both these assumptions. These generalizations were not presented for the sake of brevity. One might also generalize to operators that accept and produce variable numbers of objects from $\mathcal{O}$.

At first glance, it appears that the model is not general enough to cover situations in which landscape vertices do not correspond to sets of points in $\mathcal{R}$, but also to *partial* objects. An example of a partial object is a sub–tour in the graph of a TSP problem. This shortcoming may be simply a matter of perspective. If $\mathcal{O}$ is thought of as including the partial objects (and, possibly, the complete objects), we can proceed as before, as long as the function f can produce an indicator of the worth of a partial object. A search structure that involves partial solutions to the matrix chain association problem and the traveling salesperson problem has been developed by Helman [8].

It can be argued that the model is not particularly useful in situations where the nature of the operators changes in the course of search. For example, the change of mutation vector for an individual in evolution strategies and evolutionary programming, the change of edge probabilities when the temperature falls in simulated annealing or after inversion in a genetic algorithm, or where the representation space is changed, e.g. in dynamic parameter encoding [31] and delta coding [25]. There is some truth in this. However, if one ceases to regard a search structure as something necessarily fixed for all time, the model is still potentially useful. For example, a statistic, say correlation length, might be computed for the landscapes generated by a range of different temperature settings in a simulated annealing problem. This might provide useful information about when the search could be expected to make good progress (thus guiding the choice of cooling schedule) and it might prompt a comparison with a hillclimber or other algorithm on one of the landscapes. These situations make the landscape something of a moving target, but the targets can be studied individually.

More generally, an algorithm might change any aspect of its behavior, for example the fitness function or the representation space, mid–run, and thereby shift its attention to new landscapes. This might be done very frequently. The model is not intended to be general enough to cover every conceivable algorithm, and so its usefulness will be limited in some situations. It is intended to be general enough to cover a wide range of current situations while remaining relatively simple.

7 Other Viewpoints

There are many ways to look at things, and none of them is necessarily right. In science at least, a perspective is “good” to the extent that it provides insight, raises (and answers) questions, and so on. There are many ways to formulate landscape models. The most common point of view found in the genetic algorithm domain regards an n–dimensional hypercube as “the landscape”, and views crossover as making jumps on it and mutation as taking single steps on it. Paradoxically, this landscape structure, from the point of view proposed here, is not the structure induced by either operator! There has been a considerable amount of thinking within this framework and a lot of progress has been made.

The model presented here argues that this point of view, despite its history and the use it has been, is more complex than need be and, more importantly, contains and promotes a biased view of the two operators. The one operator, one landscape view builds independent and very different structures for mutation and crossover and insists that we give each its due. Studying the effect of using both in a population–based algorithm is of course still necessary, but this view allows us to address a simpler problem, the study of the individual structural components of such algorithms.

Another view of landscapes regards an entire population of objects as a point on some landscape. This model can also be traced to Wright [42]. Under this view, the landscape is usually thought of as having some number of dimensions and the axes represent gene frequencies in the population. This view of landscape is captured by the model presented here, but as an object of study it is vast. Progress has been made studying the genetic algorithm from this perspective, see for example [36].

One might also view selection as an operator and therefore as having its own landscape. This is perfectly valid, and statistics calculated on a selection landscape might lead to results concerning rates of convergence. Our preference has been to concentrate on simpler operators, but it should be clear that this is not necessary.

8 Related Work

There are three other pieces of work whose relevance to this model can now be considered.

Weinberger [37, 38] provides a precise definition of a fitness landscape. His landscape is also a graph (though finite), whose vertices are labeled using some real valued function. He introduces the autocorrelation function and shows its use as a measure of landscape fitness correlation or "ruggedness". This measure has been widely used [16, 17, 23, 35]

The model presented here generalizes that model by allowing operators that act on and produce tuples of points in $\mathcal{R}$. In addition, our model introduces the representation space, allows the graph to be infinite and/or unconnected, and assigns probabilities to edges which is necessary when considering mutation or crossover. Figure 2 illustrates this common situation with crossover. Weinberger only considered operators that would assign an equal probability to each outgoing edge from a vertex, and so had no need for probabilities.

Manderick *et al.* [23] used Weinberger's landscape and autocorrelation function to examine the relationship between the statistical structure of landscapes and the performance of genetic algorithms on the same landscapes. This important paper illustrates how successfully statistics concerning landscape structure can predict algorithm performance.

Manderick *et al.*, following Weinberger, viewed a landscape as something defined by an operator that acted on and produced a single point of $\mathcal{R}$. Weinberger's autocorrelation function can be calculated for these simplest of operators. They recognized crossover as an operator without making the generalization of the model described above. To deal with this, they defined a measure of operator correlation which they applied successfully to one–point crossover on NK landscapes[4] and to four crossover operators for TSP. This statistic

[4]The NK "landscapes" are not actually landscapes, as defined by the current model. They provide $\mathcal{O}$ (and $\mathcal{R}$ is obvious) and a fitness function. The crucial missing ingredient is an operator. The majority of

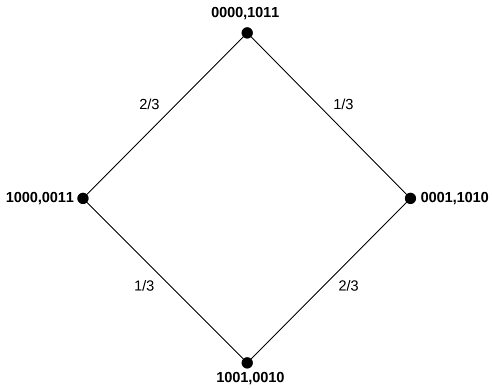

Figure 2: A fragment of the one–point crossover landscape for binary strings of length 4, showing unequal transition probabilities.

is calculated by repeated use of the operator from randomly chosen starting points. Thus, they were actually computing statistical measures about the crossover landscape without recognizing it as a landscape. Had they done so, they might have used the autocorrelation function, since they dealt with the walkable landscapes generated by crossover operators that take two parents and produce two children.

I have argued above that the most generally applicable landscape statistic will be based on fitness values obtained from repeated single operator applications from randomly chosen starting points. A measure such as autocorrelation is not computable for some landscapes because they have no distance metric. This is true for all non–walkable landscapes.

Recently, Culberson [3] has independently conceived of a crossover landscape. His structure, which he calls a *search space structure*, is also a graph and the vertices correspond to a population of points from $\{0,1\}^l$. He examines populations of size two, which creates a graph that corresponds to the largest connected component of the crossover landscape generated by one–point crossover in our model. That component (like all others) is a hypercube. It contains vertices that correspond to all possible pairs of binary strings of the form $(a, \overline{a})$. Culberson shows that the structure of the component is isomorphic to the hypercube generated by the bit–flipping operator for strings of length $l-1$. He demonstrates how to transform a problem that appears hard for one operator into a problem that appears hard for the other. This provides further evidence of the importance of structure for search and of how that structure is induced by the choice of operator.

9 Usefulness Of The Metaphor

The term "landscape" has something powerfully seductive about it. The imagery it evokes is so appealing, that further thought can be completely suspended. An important question to

work on these problems has been with respect to mutation, and references to NK "landscapes" always imply a mutational viewpoint.

ask is why we would want to use such a term. The answer to this is presumably that we hope to use the imagery (e.g. peaks, ridges, valleys etc.) to enhance our understanding of some process, to develop new ideas for exploring spaces and to stimulate questions about processes operating on these structures. All of this tends to rely rather heavily on the simple properties that we see in physical three dimensional landscapes. It is not clear just how many of the ideas scale up to landscapes with hundreds or thousands of dimensions. It is quite possible that the simplicity and beauty of the metaphor is actually damaging in some instances, for example by diverting attention from the actual process or by suggesting appealing, simple and incorrect explanations. All of this has been put very well by Provine [28] (pp. 307–317), which should be required reading for people interested in employing the metaphor.

The ambiguities surrounding the term and its use originated with Wright, and were not identified until 1985 [28]. These problems can also be found in the field of evolutionary computing. Given this, it is worth asking whether it is better to abandon the term or to use it and try to be more precise about what is actually meant. There is something to be said for abandoning it – after all, in just about every formulation, a landscape is simply a graph. On the other hand, it seems unlikely that the term will just go away. In addition, the metaphor, however distant it may sometimes be from reality, *has* given rise to new ideas and intuitions. This paper has opted to adopt the term, with the hope that it will lessen, rather than increase, the vagueness with which it is applied.

10 Conclusion

This paper presented a general model of landscapes and an overview of its consequences and relevance to genetic algorithms. The model views a landscape as a directed graph whose edges and vertices are labeled, though a vertex is not constrained to correspond to exactly one point from the representation space. It was argued that the operators in evolutionary algorithms, including crossover, each generate a landscape; that these landscapes have differing qualities; and that each can and should be studied in its own right, independent of any navigation strategy. Thus genetic algorithms are seen as operating on multiple landscapes. Defining a landscape as a graph establishes a contact with more traditional search algorithms from artificial intelligence, many of which are explicitly designed to search labeled graphs. The paper advocates a view of search as composed of navigation and structure, with the structure provided by landscapes. The advantages and limitations of the model were briefly discussed and the model was compared to other work on landscapes.

11 Acknowledgements

Thanks to Joseph Culberson, Walter Fontana, Stephanie Forrest, Ron Hightower, John Holland, Stuart Kauffman, Melanie Mitchell, Jesús Mosterín, Una–May O'Reilly, Richard Palmer, Gregory Rawlins, Derek Smith, and Peter Stadler for the many provocative conversations over the last two years that have led me to the ideas in this paper. Ron Hightower

also helped to make this paper more readable. Many thanks to the Santa Fe Institute for providing an environment that makes this kind of communication possible.

References

[1] Amitrano, C., Peliti, L. and Saber, M. [1987] "A Spin–Glass Model of Evolution" In Perelson, A. and Kauffman, S. (Eds.) *Molecular Evolution on Rugged Landscapes: Proteins, RNA and the Immune System*, Santa Fe Institute Studies in the Sciences of Complexity, Vol IX. Addison–Wesley, Redwood City, CA. pp. 27–38.

[2] Bäck, Thomas, Hoffmeister, Frank, and Schwefel, Hans–Paul [1991] "A Survey of Evolution Strategies" In Richard K. Belew and Lashon B. Booker (Eds.) *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, pp. 2–9.

[3] Culberson, Joseph C. [1994] "Mutation–Crossover Isomorphisms and the Construction of Discriminating Functions" Accepted by *Evolutionary Computation.*

[4] Eldredge, N. [1989], *Macroevolutionary Dynamics: Species, Niches and Adaptive Peaks*, McGraw–Hill.

[5] Fontana, W., Stadler, P. F., Bornberg–Bauer, E. G., Griesmacher, T., Hofacker, I., Tacker, M., Tarazona, P., Weinberger, E. D. and Schuster, P. [1993], "RNA folding and combinatory landscapes" *Physical Review E*, Vol. 47, Number 3, pp. 2083–2099.

[6] Ginsberg, Matt [1993], *Essentials Of Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA.

[7] Goldberg, David E. [1989], *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley.

[8] Helman, Paul [1988], "An Algebra For Search Problems And Their Solutions" In L. Kanal and V. Kumar (Eds) *Search In Artificial Intelligence*, pp. 28–90. Springer Verlag, NY.

[9] Holland, John H. [1975], *Adaptation in Natural and Artificial Systems*, University of Michigan Press.

[10] Holland, John H. [1993], "Royal Road Functions" Internet Genetic Algorithms Digest vol. 7, issue 22, Aug 12th 1993.

[11] Horn, Jeffrey, Goldberg, David E. and Deb, Kalyanmoy [1992], "Research Note: Long Path Problems for Mutation–Based Algorithms" Illinois Genetic Algorithm Library report 92011, Champaign–Urbana IL.

[12] Jones, Terry and Rawlins, G. J. E. [1993] "Reverse Hillclimbing, Genetic Algorithms and the Busy Beaver Problem" In S. Forrest (Ed.) *Genetic Algorithms: Proceedings of The Fifth International Conference (ICGA 1993)*, Morgan Kaufmann, San Mateo, CA.

[13] Jones, Terry [1994] "Computational Landscapes" Ph.D. dissertation (in preparation). Department of Computer Science, University of New Mexico, Albuquerque NM.

[14] Kauffman, Stuart A., Weinberger, Edward D. and Perelson, Alan S. [1987], "Maturation of the Immune Response Via Adaptive Walks on Affinity Landscapes" In Alan Perelson (Ed.) *Theoretical Immunology*, Santa Fe Institute Studies in the Sciences of Complexity, Addison–Wesley, vol. 2, part one, pp. 349–382.

[15] Kauffman, Stuart A. [1988], "The Evolution of Economic Webs" In Philip W. Anderson, Kenneth J. Arrow and David Pines (Eds.) *The Economy As An Evolving Complex System*, Santa Fe Institute Studies in the Sciences of Complexity, Addison–Wesley, vol. 5, pp. 125–146.

[16] Kauffman, Stuart A. [1989], "Adaptation on Rugged Fitness Landscapes" *Lectures in the Sciences of Complexity*, Ed. D. Stein, Addison–Wesley Longman, vol. 1, pp. 527–618.

[17] Kinnear, K. E. Jr. [1994] "Fitness Landscapes and Difficulty in Genetic Programming" Submitted to EC'94, The IEEE Conference on Evolutionary Computing.

[18] Kollman, Ken, Miller, John H. and Page, Scott E. [1993] "Political Parties and Electoral Landscapes" Santa Fe Institute Technical Report 93–01–003, Santa Fe Institute, Santa Fe NM.

[19] Koza, John R. [1992] *Genetic Programming : On the programming of computers by means of natural selection*, MIT Press, Cambridge MA.

[20] Lansing, J. Stephen and Kremer, James N. [1994] "Emergent Properties of Balinese Water Temple Networks: Coadaptation on a Rugged Fitness Landscape" In Christopher G. Langton (Ed.) *Artificial Life III* Santa Fe Institute Studies in the Sciences of Complexity, Vol IX. Addison–Wesley, Reading MA. pp. 201–223.

[21] Luger, George F. and Stubblefield, A. [1989] *Artificial intelligence and the design of expert systems*, Benjamin/Cummins, Redwood City CA.

[22] Macken, C. A. and Perelson, A. [1987], "Affinity Maturation on Rugged Landscapes" In Perelson, A. and Kauffman, S. (Eds.) *Molecular Evolution on Rugged Landscapes: Proteins, RNA and the Immune System*, Santa Fe Institute Studies in the Sciences of Complexity, Vol IX. Addison–Wesley, Redwood City, CA. pp. 93–118.

[23] Manderick, B., De Weger, M. and Spiessens, P. [1991], "The Genetic Algorithm and the Structure of the Fitness Landscape" In Richard K. Belew and Lashon B. Booker (Eds.) *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, pp. 143–150.

[24] Mathias, Keith and Whitley, Darrell [1992] "Genetic operators, the fitness landscape and the traveling salesman problem" In R. Männer and B. Manderick (Eds.) *Parallel Problem Solving from nature, 2* Elsevier Science Publishers B.V.

[25] Mathias, Keith and Whitley, Darrell [1993] "Remapping Hyperspace During Genetic Search: Canonical Delta Folding" In L. Darrell Whitley (Ed.) *Foundations of Genetic Algorithms 2* Morgan Kaufmann, San Mateo, CA, pp. 167–186.

[26] Mitchell, M., Forrest, S. and Holland, J. H. [1991], "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance" In *Proceedings of the First European Conference on Artificial Life*, MIT Press, Cambridge MA.

[27] Palmer, Richard [1988] "Statistical Mechanics Approaches to Complex Optimization Problems" In Philip W. Anderson, Kenneth J. Arrow and David Pines (Eds.) *The Economy As An Evolving Complex System*, Santa Fe Institute Studies in the Sciences of Complexity, Addison–Wesley, vol. 5, pp. 177–193.

[28] Provine, William B. [1986], *Sewall Wright And Evolutionary Biology*, University of Chicago Press, Chicago IL.

[29] Rechenberg, I. [1973] *Evolutionsstrategie: Optimierung techniquer Systeme nach Prinzipien der biologischen Evolution*, Frommann–Holzboog Verlag, Stuttgart, 1973.

[30] Rich, Elaine [1983] *Artificial Intelligence*, McGraw–Hill.

[31] Schraudolph, N. N. and Belew, R. K. [1992] "Dynamic Parameter Encoding for Genetic Algorithms" *Machine Learning* vol. 9 no. 1 (June), pp. 9–21.

[32] Schuster, Peter and Stadler, Peter F. [1993] "Landscapes: Complex Optimization Problems and Biopolymer Structures" Santa Fe Institute Technical Report 93–11–069, Santa Fe Institute, Santa Fe NM.

[33] Schwefel, H.–P. [1977], "Numerische Optimierung von Computer–Modellen mittels der Evolutionsstrategie" *Interdisciplinary Systems Research*, Vol. 26. Birkhäuser, Basel.

[34] Shaefer, Craig G. [1987] "The ARGOT Strategy: Adaptive Representation Genetic Optimizer Technique" In Grefenstette, John J. (Ed.) *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates. Hillsdale NJ. pp. 50–55.

[35] Stadler, Peter, F. and Schnabl, Wolfgang [1992], "The landscape of the traveling salesman problem" *Physics Letters A* Vol. 161. pp. 337–344.

[36] Vose, Michael, D. [1993] "Modeling Simple genetic Algorithms" In L. Darrell Whitley (Ed.) *Foundations of Genetic Algorithms 2* Morgan Kaufmann, San Mateo, CA, pp. 63–73.

[37] Weinberger, E. D. [1990], "Correlated and uncorrelated fitness landscapes and how to tell the difference" *Biological Cybernetics*, vol. 63, pp. 325–336.

[38] Weinberger, E. D. [1991], "Measuring Correlations in Energy Landscapes and Why It Matters" In H. Atmanspacher and H. Scheingraber (Eds.) *Information Dynamics*, Plenum Press, New York, pp. 185–193.

[39] Whitley, D., Mathias, K. and Fitzhorn, P. [1991], "Delta Coding: An Iterative Search Strategy for Genetic Algorithms" In Richard K. Belew and Lashon B. Booker (Eds.) *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, pp. 77–84.

[40] Winston, Patrick Henry [1992] *Artificial Intelligence*, 3rd Ed. Addison–Wesley.

[41] Wright, Sewall [1932], "The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution" *Proceedings of the Sixth International Congress of Genetics*, vol. 1, pp. 356–366. (This paper is reprinted in [43].)

[42] Wright, Sewall [1935], "Evolution in populations in approximate equilibrium" *Journal of Genetics*, vol. 30, pp. 257–266.

[43] Wright, Sewall [1986] *Evolution: Selected Papers*, William B. Provine, Ed. University of Chicago Press, Chicago IL.