Information Landscapes

Yossi Borenstein
University of Essex
yboren@essex.ac.uk

Riccardo Poli
University of Essex
rpoli@essex.ac.uk

ABSTRACT

We give a new interpretation to the concept of "landscape". This allows us to develop a new theoretical model to study search algorithms. Particularly, we are able to quantify the *amount* and *quality* of "information" embedded in a landscape and to predict the performance of a search algorithm over it. We conclude presenting empirical results for a simple genetic algorithm which strongly support this idea.

Categories and Subject Descriptors

F.2.0 [**Theory of Computation**]: Analysis of algorithms and problem complexity.

General Terms

Algorithms, Performance, Theory.

Keywords

Fitness landscape, Genetic Algorithm, Theory

1. Introduction

During the last 20 years many algorithms (metaheuristics) have been proposed to explore search spaces in an efficient way [1]. Usually a search algorithm tries to infer the position of good new solutions in the search space based on previously sampled solutions.

Many metaheuristics are inspired by powerful natural or physical processes. Ant Colony Optimization (ACO), Evolutionary Computation (EC) and Simulated Annealing (SA) are examples of such algorithms. ACO and EC are inspired by nature; SA is inspired by the annealing process of metals and glass.

These and other metaheuristics have been applied successfully to an ever increasing number of hard combinatorial optimization problems such as TSP, vehicle routing, job shop scheduling, and bin packing. However, in many cases, their remarkable empirical success is not associated to corresponding robust theoretical foundations. This, in turn, makes it difficult to match efficiently the right algorithm with the right problem or even to tune the parameters of a search algorithm in an optimal way without the need of extensive preliminary experimentation.

Metaheuristics are typically used in a black-box scenario. The analysis of black-box optimization algorithms usually follows one of two approaches. The first approach is the study of the properties of a particular algorithm and, so, it is algorithm-specific. The second approach, instead, focuses more on the problem itself, and in particular on the notion of fitness landscape. In the next subsections we exemplify these two approaches focusing, for the sake of brevity, on the Genetic Algorithm (GA). Since this is one of the most popular metaheuristics we think this is an interesting case-study.

GECCO'05, June 25-29, 2005, Washington, DC, USA.

1.1 Algorithm-specific approaches

During the last decade there have been several attempts to characterize GA behavior. The Building Block (BB) hypothesis [2] states that a GA tries to combine low, highly fit schemata. Following the BB hypothesis the notion of deception [2,3] has been defined. Epistasis variance [4] and epistasis correlation [5] try to assess the GA-hardness of problems from a theoretical genetics perspective. Sitewise optimization [6], a generalization of a fitness distance correlation (more on this below) and epistasis, was suggested as well. None of these methods has fully succeeded in giving a reliable measure of GA-hardness [6][7][8][9].

Fundamentally, the reason for this is that the intrinsic complexity of modern metaheuristics makes it difficult to explore their dynamics theoretically. Indeed, recent work published in the EC field suggests that, due to lack of proper general classification methods, empirical results should be used instead of theory [8][9].

1.2 Fitness landscape approaches

The second approach studies the properties of the fitness landscape. The notion of *fitness landscape* was introduced in theoretical genetics [10] as a way to visualize evolution dynamics. The intuition behind it is that fitness can be thought of as a "height function", orthogonal to the genome space and hence high fitness points are located in "peaks" and low fitness points in "valleys".

The landscape connects to an algorithm via the neighborhood structure. Properties of the landscapes are expected to affect the performance of the algorithm because the algorithm uses the *same* neighborhood structure to conduct its search. So, in theory one could infer GA hardness from these properties.

Isolation [3] and multimodality [4] were perhaps the first attempts to connect the fitness landscape to the complexity of a problem. Isolation might be sufficient but is not a necessary condition for a difficult landscape. Multimodality is neither necessary nor sufficient for a landscape to be difficult to search [13].

Fitness distance correlation [14] measures the hardness of a landscape according to the correlation between the distance from the optimum and the fitness of the solution. Despite good success, fitness distance correlation is not able to predict performance in some scenarios [15][6].

NK landscapes [16],[17] use the idea of epistasis in order to create artificial, arbitrary landscapes with a tunable degree of difficulty. However, even though NK landscapes are interesting from theoretical genetics point of view, their practical use is questionable [8].

To the best of our knowledge at present there is no measure of landscape difficulty which is able to capture precisely the hardness of the search. In our opinion, linking the properties of a fitness landscape with search hardness is problematic for various reasons.

Firstly, the main perspective when considering a fitness landscape should be the *information* that is available to the search algorithm in order to conduct the search. The needle in a haystack, for example, is considered to be a very difficult problem because it does not contain any information to guide the search. Intuitively it is arguable that the *quantity of information* available to an algorithm is very important in determining its performance. This, of course, cannot be the only factor. For example, even if there is a lot of information in a landscape, this information can be such to deceive an algorithm steering it away from global optima more often than not. Conversely, we can have a situation, e.g. in the case of a unimodal fitness function, where the information available rapidly and invariably guides the search towards good solutions. So, the *quality of the information* available to an algorithm is also crucial in determining performance. A limitation of the original fitness landscape approach is that it does not provide a way to quantify the amount of information available in a landscape nor to asses its quality. This is why there is a big gap between the notion of fitness landscape and the notion of search hardness. In this paper we will propose a solution to this problem based on a redefinition of the notion of landscape.

Secondly, the analysis of the fitness landscape assumes that the neighborhood structure and the fitness function are sufficient to describe the properties of the landscape. This is of course true, but the final goal when assessing search hardness is to describe the properties (e.g. the performance) of the search not those of the problem. Because different search algorithms use different operators which induce different landscapes [18] and sample those landscapes in different ways, evaluating the difficulty of the search without considering the details of the algorithm being used seems hard if not impossible.

This is what led Stadler and Stephenes to state [19] "Hill climbing on a static fitness landscape is an adequate metaphor in the case of evolution dominated by selection. We have emphasized in this paper that when this is not the case, then the utility of the landscape concept is much diminished."

Finally, the information given in a fitness landscape is redundant, at least for some of the modern metaheuristics. For example, often GAs use tournament selection as the selection paradigm. In such a case the absolute fitness of solutions is of no relevance to the algorithm: only the relative order between individuals is. So, analyses based on the full fitness landscape may end up being overly complicated.

1.3 A black box model of performance

As we discussed above, on the one hand the use of the concept of a fitness landscape to assess problem difficulty is problematic. On the other hand, there has also been a relative lack of success in constructing useful algorithm-specific theoretical models even when full knowledge of an algorithm and the problem at hand is available. This suggests that it might be more constructive to treat the *search algorithms* itself as a *black box.*

In this framework, the fitness function is the search algorithm, the input of the search algorithm is a fitness landscape (i.e. search space, neighborhood structure, and a fitness function) and the output is the performance of the search algorithm over the particular fitness landscape (figure 1).

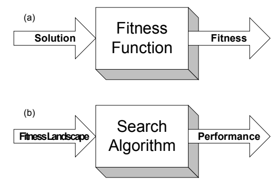


Figure 1. (a) The traditional black box model. (b) Our model: considering the *search algorithm* as a black box.

More formally, we consider the performance function $P : f, X, \chi \rightarrow \Re$ where X is a search space of a particular size, χ is the neighborhood structure which is defined over X and f is a fitness function defined over X . The performance function is related to an unknown search algorithm. *P* is any given measure of performance (e.g., the average number of fitness evaluations required to find the optimum).

In order to simplify the model we study the measure of performance for a search space of a given size, and for a particular neighborhood structure. This does not affect the generality of our model because both the size and the neighborhood structure are arbitrary.

It is important to emphasize that unlike the traditional black-box model, we are not particularly interested in optimization, i.e. in trying to find the fitness landscape (input) which provides maximum performance (output). Rather we are interested in analyzing the properties of the search algorithm.

The paper is organized as follows. In the next section we give a new interpretation to the traditional concept of a landscape. Since this is based on the quantity and quality of the *information* available to the search algorithm we call this landscape an *information landscape*. In section 3, an approximation measure to the performance function *P* is computed. We call this approximation a *performance landscape*. We conclude with empirical results and a discussion.

2. Information landscapes

A fitness landscape (X, χ , f) is traditionally defined by specifying its three basic ingredients [19]:

1. a set X of configurations
2. a notion χ of neighborhood, nearness, distance or accessibility on X, and
3. a fitness function $f : X \rightarrow \Re$

As stated in the introduction, given the disadvantages of the existing concept of a landscape we suggest a new concept which, in our opinion, is able to capture in a better way the knowledge that the algorithm actually uses in a black box scenario. Our new perspective is based on the following simple observations.

The objective of a search algorithm is to find fit solutions in the search space. In order to do so, it has to sample the objective function. Algorithms that use tournament or rank selection use the fitness function only in order to choose, out of two or more tentative solutions, the one that will be the basis of the next step of the search.

The dynamics of search algorithms is often governed by one simple rule: the higher the fitness of a solution the higher the probability that it can lead the algorithm to other solutions (e.g., the offspring of such a solution) with similar or higher fitness. The underlying heuristic/assumption of the algorithm is that applying the search operators on solutions with high fitness values (as opposed to ones with low fitness) is more likely to yield solutions which are closer to the optimum (we only consider optimization problems). That is, implicitly we assume that the neighborhood structure has certain properties. In particular, that close solutions (defined by χ) have similar fitness.

Consider a hill climber on a unimodal landscape. The closer the initial search point to the optimum (w.r.t to the neighborhood structure) is, the better the performance of the algorithm will be. In other words, applying the search operator of a hill climber on a point which is closer to the optimum will give better results. In this scenario the *distance from the optimum* is a sufficient indicator to the performance of the algorithm.

This is not the case for more complex algorithms. For example, in certain cases a crossover operation can construct the optimum even if both parents are far away from the optimum. In these cases, the performance of a search algorithm depends on the neighborhood induced by the search operator (not any other notion of neighborhood which may have been provided with the fitness landscape). The performance of a memetic algorithm on the other hand, depends on *both* the search operators and the neighborhood structure. Generally, the performance is a function of the search operators and the values of various parameters controlling the algorithm.

Informally we refer to the chance of finding the optimum by applying the search operators on a point as the *effective distance* of that point from the optimum. For example, in the case of GA, given a particular population, the *effective distance* of a string from the optimum is related to the probability that given that this string is selected into the mating pool, the optimum will be found during the run. The higher the probability the smaller the *effective distance* will be. It is important to note that the *effective distance* is a function of the search operators, the parameters and the target solution (the optimum). It is not related, in any way to the *fitness* of the solution[1].

This leads us to the following conclusion:

1. Search algorithms based selection use the fitness function as a tool to *compare* solutions. (Naturally, this is true for algorithms using tournament selection or rank selection. We argue that this is approximately the case also for other selection schemes.) So, for a search space of size $|X|$ the fitness function can be decomposed into $|X|^2$ values representing the outcome of a comparison between every possible pair of solutions. This captures, in many cases, *all and only* the information that the search algorithm uses.
2. The fitness of a solution is not related to the effective distance of the solution from the optimum. However, the algorithm uses the fitness value as an indicator for such a distance. The performance of the algorithm depends therefore, on the correlation between the *relative fitness* (i.e. the *information* given by the fitness function) and the *effective distance* from the optimum.

The performance of an algorithm is therefore only a function of a $|X|\times|X|$ matrix M, in which each entry $m_{i,j}$ represents the relative order of solutions "*i*" and "*j*". We are now in a position to introduce our notion of *information landscape*.

Formally, an *information landscape* is the triple (X, χ, t) including

1. a set X of configurations
2. a notion χ of neighborhood, nearness, distance or accessibility on X, and
3. A stochastic information function $t: X \times X \rightarrow [0,1]$.

For every pair (x_i, x_j) of elements in X, t gives the probability that x_i is superior to x_j. The value of the function t can be viewed as the outcome of a stochastic tournament selection with tournament size two. Naturally, the function t can be represented as an $|X| \text{x} |X|$ information matrix M with entries $m_{i,j} = t(x_i, x_j)$. Note that when X is implied we can use the term *information landscapes* to denote M without ambiguity.

When a fitness function f is available, normally:

$$t(x_i, x_j) = \begin{cases} 1 \;\; if \; f(x_i) > f(x_j) \\ 0.5 if \; f(x_i) = f(x_j) \\ 0 \qquad otherwise \end{cases} \tag{1}$$

However, if the fitness function is noisy, t can take values other than 0, 0.5 and 1. In the rest of the paper we use the terms problem and landscape interchangeably. By landscape we mean an information landscape.

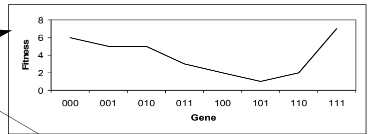
Given an information landscape we can construct the following rank-based fitness function:

$$f_{rank}(k) = \sum_j m_{k,j} \tag{2}$$

It is important to mention that not all the information landscapes can be associated to a fitness function (the information matrix may not induce a partial order). We will call *invalid* those information landscapes that can not be derived form a corresponding fitness landscape.

---

[1] The effective distance depends to some extent on the different solutions present in a particular population. However, on average the dependency is null.



(b)

(a)

Gene	Fitness
000	6
001	5
010	5
011	3
100	2
101	1
110	2
111	7

(c)

Gene	000	001	010	011	100	101	110	111
000		1	1	1	1	1	1	0
001	0		0.5	1	1	1	1	0
010	0	0.5		1	1	1	1	0
011	0	0	0		1	1	1	0
100	0	0	0	0		1	0.5	0
101	0	0	0	0	0		0	0
110	0	0	0	0	0.5	1		0
111	1	1	1	1	1	1	1	

Figure 2. Three ways of representing the information given to a search algorithm: a) a fitness function (represented as a vector) b) a graph, representing topological properties (fitness landscape) and c) a matrix representing the outcome of all possible comparisons between solutions (information landscape). The matrix presents symmetries with respect to the diagonal; the gray area marks the distinct elements.

Figure 2 gives an example of a fitness function, a fitness landscape and the matrix which represents our information landscape for a bit-string configuration space.

At this stage we consider only optimization problems with one global optimum. The objective of the search is to find it. For the purpose of this study we assume that the algorithm has a way of identifying the optimum (e.g., from its fitness) when this gets sampled. Further, we assume that once the optimum is found the search stops. Therefore, the entries of the information landscape which correspond to the optimum are of no relevance to the search. For this reason we exclude those entries from the information landscape (figure 2). More formally, only the set $X \setminus \{X_{target}\}$ is being considered.

Since $t(x_i, x_j) = 1 - t(x_j, x_i)$, the matrix (figure 2) presents symmetries with respect to the diagonal; the gray area marks the distinct elements of the information landscape. Diagonal elements (omitted for clarity) are all 0.5. In order to account for all this in a simple way we use a vector to store the relevant (above diagonal) entries in the matrix:

$$V = (v_1, v_2, ..., v_n) = (m_{1,2}, m_{1,3}, ..., m_{|X|-1,|X|})$$

where $|V| \equiv n = (|X|-1)(|X|-2)/2$.

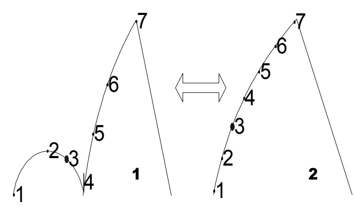
2.1 Quantity and Quality of Information

In the previous section we gave a formal definition of the information landscape. In this section we define a distance between two landscapes. From this we derive measurements of the *quality* and the *quantity* of information embedded in a landscape. These measurements are the main contribution of the information landscape compared with the traditional fitness landscape.

Let V_1, V_2 be two information landscapes. The distance between them is defined as:

$$D(V_1, V_2) = \frac{1}{n} \sum \left| V_{1_i} - V_{2_i} \right| \quad (3)$$

The distance defined in equation 3 is linearly dependant on the number of differing entries in the two landscapes. Figure 3 illustrates how the distance between two landscapes can be interpreted. Figure 3.A shows unimodal and bimodal landscapes. Figure 3.B gives the "bimodal" information landscape. The gray area represents the entries in the "bimodal" information landscape which differ from those of the unimodal one.



B

Difference from A2

Infomration Landscape of A1

Target Solution

i/j	1	2	3	4	5	6	7
1	0.5	0	0	0	0	0	0
2		0.5	1	1	0	0	0
3			0.5	1	0	0	0
4				0.5	0	0	0
5					0.5	0	0
6						0.5	0
7	1	1	1	1	1	1	0.5

Figure 3. (A) Bimodal and unimodal fitness landscapes; seven points are shown. (B) The bimodal information landscape (A1). The gray area indicates where the bimodal information landscape is different from the unimodal one.

In order to understand how the *quality* of the information can be quantified, consider the performance of a *hill climber* on these two landscapes. Note that for a hill climber, under this representation, the distance (as defined by the neighborhood structure) and the *effective distance* of a solution from the optimum coincide.

Consider the entry $m_{3,4}$ in figure 3(b), the value 1 indicates that solution "3" wins in a comparison between "3" and "4". The underlying assumption of the algorithm is that a solution with higher fitness is more likely to lead eventually to the optimum. In this example, this piece of information is misleading. The same entry for the unimodal landscape (Figure 3.A.2) is 0, which indicates that "4" has a better chance of leading to the global optimum than "3".

Under ideal conditions, an algorithm would want to choose the solutions on which to base the next step of the search according to their effective distance from the optimum. In practice, this information is not available and so the algorithm chooses according to their fitness. The assumption being the higher the fitness, the closer the global optimum. Thus, when evaluating whether to choose solution *j* or solution *i,* if $m_{i,j}$>0.5, the search algorithm will pick *i,* the assumption being that *i* is effectively closer to the optimum than *j*. That is, the information landscape has a dual interpretation. On the one hand it simply represents the relative order of the solutions w.r.t the fitness. On the other hand, it represents a table of *assumed* relative effective closeness to the optimum. Naturally (as illustrated in the previous example), not all assumptions will be correct. We expect the *performance* of the algorithm to correlate with the number of correct assumptions.

In order to define the quality of the information we make a distinction between positive information and misleading one. Positive information refers to a tournament which returns the solution *effectively* closer to the optimum.

Following this reasoning, there is a landscape which contains only correct assumption. The *distance* from this landscape will give the *amount* of *negative* information embedded in any other landscape.

Different search algorithms have different optimal landscapes, hence, once the optimal landscape is known, this method can be applied to any search algorithm. In [20] we used this notion in order to develop a measure of GA hardness.

Naturally, when the outcome of the tournament is random (i.e. an entry in the information landscape equals 0.5) the information is neither positive nor negative: it is absent. The *amount* of information available to an algorithm is defined therefore, as the fraction of entries, in the information landscape which differ from 0.5.

We refer to this amount as the *degree* $d^{0.5}$ of the information landscape. Formally, it is the distance between a landscape and the landscape where all matrix elements are 0.5, normalized to the range [0,1]:

$$d^{0.5}(V) = \frac{2}{n}\sum |v_i - 0.5| \tag{4}$$

If there is no information in a landscape, a search algorithm is not expected to perform better than random search. The degree to which it can outperform random search depends on the amount of information. This can be measured using equation 4.

The information however can be either positive or negative. The amount of negative information can be measured by the distance from an optimal landscape. If such a landscape is not available an approximation of the information landscape of an empirically known easy problem can be used for this purpose.

3. Performance landscape

Given the definition of the information landscape, we can now define *P*, in a proper way. We consider the performance function $P: V, X, X_{trgt}, \chi \rightarrow \Re$ where X is a search space of a particular size, χ is the neighborhood structure which is defined over X and V is the information landscape. Since by definition the information landscape does not include the entries related to the target solution (optimum) X_{target} we have to supply this value separately. The value of *P* is a measure of performance (e.g. the number of average fitness evaluations it takes in order to find the optimum).

Since in this paper we assume that everything but V is constant, *P* can be written as $P: V \rightarrow \Re$. Our objective in this section is to build a useful approximation of the performance function.

P is a complicated function of *n* variables for which we have no explicit formulation. However, this function can be estimated using machine learning techniques if a suitable training set is available. A train set would have to be made up of pairs of the form (*V*, *P*) where *V* is a particular (input) information landscape and *P* is the target performance measure. This can be estimated by averaging the performance recorded by running an algorithm a suitably large number of times over the particular landscape in question. Naturally, for this approach to work the training set should include a representative set of problems, that is, it should include easy, difficult and random problems.

To illustrate this approach, in this paper we consider the simplest possible approximation. We start by assuming that *P* is a linear function of the *n* variables in *V* (the entries of the information matrix). Using empirical results over a sample of the possible information landscape*s* as observations, we can apply a multivariate linear regression in order to estimate the performance function:

$$P(V) \cong c_0 + \sum c_i v_i \tag{5}$$

In many cases a linear dependency for the specific algorithm might not hold over the whole space of possible information matrices. In that case, assuming that *P* is differentiable, we can use a truncated Taylor expansion in order to approximate *P* in the neighborhood of any reference information landscape V_0. That is:

$$P(V) \cong P(V_0) + \nabla P(V)\big|_{V=V_0} (V - V_0) \tag{6}$$

Therefore:

$$P(V) \cong c_0 + \sum c_i (v_i - v_{0_i}) \tag{7}$$

where $c_i = \left.\frac{\partial P(V)}{\partial v_i}\right|_{V=V_0}$ and $c_0 = P(V_0)$ is the performance of the algorithm on the reference landscape V_0.

From a practical point of view, even when using multivariate linear regression, we will use equation 7 with $v_{0_i} = 0.5$. This is done for two reasons, Firstly, in our framework, the value 0.5 indicates the lack of information for a particular entry. However, when computing the *performance landscape* (see below), it is more convenient to use 0 as such an indicator. Secondly, it illustrates the performance of the algorithm on any particular problem w.r.t. a random problem, or a random landscape.[2]

Whether using multivariate linear regression or the Taylor expansion around a reference landscape, the performance *P* depends on the coefficients c_i. We denote the vector $C = (c_i)$ (or its corresponding matrix form) as the *performance landscape*.

The importance of the linear approximation of *P* is twofold. Firstly, it gives us a simple model to predict the performance of search algorithms in general. Secondly, it can be used to make prediction about the performance of a search algorithm on an unseen problem. This, in turn, may make it possible to classify problem hardness, under some conditions.

From equation 7 the following observations can be made:

- If $c_i = 0$ then the entry in the information matrix represented by v_i is not relevant to the search.
- If $c_i > 0$ then $\max(c_i(v_i - v_{o_i})) = c_i(1 - v_{o_i})$ and this is achieved for $v_i = 1$

---

[2] In principle this represents a flat landscape. However, in practice, it means that each solution has an equal probability to win or loose a tournmanet. In [21] we showed that the performance on a radom landscape is different from the performance on a flat landscape. Therefore, rather than using a flat landscape, we use a random one.

- If $c_i < 0$ than $\max(c_i(v_i - v_{o_i})) = -c_i v_{o_i}$ and this is achieved for $v_i = 0$.

So, for a given performance landscape *C* we should expect our algorithm to provide best performance on the following information landscape:

$$V_{\max} = \left(\arg\max_{v_i} [c_i(v_i - v_{o_i})]\right) \quad (8)$$

Moreover, when $V_o \equiv 0.5$, we should expect to see symmetry in the performance of the algorithm with respect to its performance on a random problem as stated in the following

Theorem: *Let* $\underline{1}$ *be an information landscape with all (above diagonal) entries equal to 1, and* V_0 *be an information landscape with all entries equal to 0.5. The negation of an information landscape V is defined as:*

$$\overline{V} \equiv \underline{1} - V \quad (9)$$

Then :

$$P(\overline{V}) - P(V_0) = P(V_0) - P(V) \quad (10)$$

Proof: In a neighborhood of V_0

$$P(V) \cong P(V_0) + \sum c_i(v_i - 0.5) \Rightarrow$$

$$P(V_0) - P(V) \cong -\sum c_i(v_i - 0.5) = \sum c_i(-v_i + 0.5)$$

$$= \sum c_i(1 - v_i - 1 + 0.5) = \sum c_i(\overline{v}_i - 0.5)$$

$$= P(V_0) + \sum c_i(\overline{v}_i - 0.5) - P(V_0) = P(\overline{V}) - P(V_0)$$

Interestingly the approximate *performance of a search algorithm on an information landscape can be interpreted as the scalar product of the information landscape and the performance landscape*. (The former represents the search problem, the latter the performance characteristics of the search algorithm.)

Indeed, in the case of the Taylor expansion,

$$P(V) \cong c_0 + \sum_i c_i(v_i - 0.5) =$$

$$= \sum_i c_i v_i + \underbrace{c_0 - \sum_i c_i v_{0_i}}_{\text{constant}} = C \cdot V + \text{constant}$$

In the case of multivariate linear regression

$$P(V) \cong \sum c_i v_i + \text{constant} = C \cdot V + \text{constant}$$

The scalar product interpretation can help us visualize the connection between the information landscape and the performance landscape. To a first order approximation, the performance is simply the magnitude (with sign) of the projection of one vector onto the other. *V* represents the information on a problem. *C* represents the sensitivity of the search algorithm to different pieces of information. Figure 4 shows how this can be used to compare the performance of different search algorithms on a particular problem (Figure 4.A) and the performance of a search algorithm on different problems (Figure 4.B).

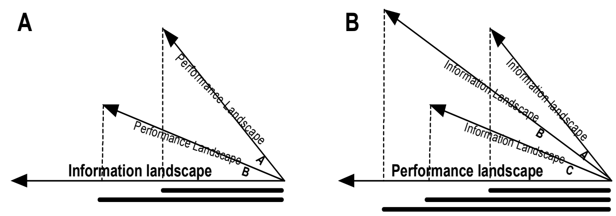


Figure 4. (a) The information landscape represents a particular problem. The two thick lines represent the performance of two different search algorithms on this problem. (b) The performance landscape represents a particular search algorithm. The solid lines represent the performance of the algorithm on 3 different problems.

4. Empirical Results

Giving empirical support to our ideas is not simple. For example, in order to do so, one must analyze the performance of an algorithm over all possible landscapes which is practically impossible except for landscapes of a very small size. In the following subsections we give an exhaustive analysis for a small landscape (8 possible solutions) for a simple GA, we corroborate theorem 1 and we give empirical evidence that it holds for also in more realistic scenarios.

4.1 Exhaustive analysis

We used a simple GA with onepoint crossover used with 100% probability. Due to the small size of the landscape, we had to choose a measure of performance which is highly sensitive. We chose the takeover time (i.e. the time require for the entire population to converge to the target solution) as the performance measure. We used a population size of 14. The maximum number of generations was 500. The search on each landscape was repeated 1000 times. The results are the average of those runs. The target solution (global optimum) was excluded from the first generation.

In a first experiment, we measured the mean takeover time for *all possible valid landscapes* of degree 1. It is important to emphasize that the target solution is fixed. Therefore, we were measuring the performance of a GA on all possible landscapes *given that* the optimum is at a particular position in the search space. Since the size of the landscape is 8, overall we have (8-1)! =5040 possible landscapes.

In order to estimate the performance landscape (equation 7) we used multivariate linear regression on the results obtained from running the GA over all such landscapes. Figure 5 plots the predicted line against the empirical results. For easier visualization landscapes were ordered on the basis of their predicted performance (from best to worst). The correlation coefficient between observed and predicted performance is 0.935, which indicates that a linear approximation to the performance function is actually rather accurate.

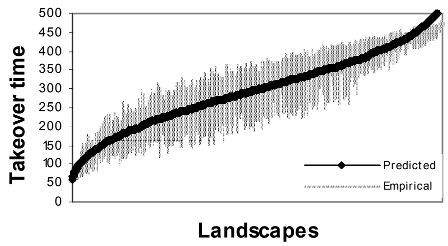


Figure 5. The predicted performance against the empirical performance of a GA on all possible valid landscape of size 8.

In order to verify whether the linear approximation to *P(V)* generalizes well, we sampled 1000 additional landscapes out of the entire space of possible information landscapes (i.e. including invalid landscapes). Figure 6 plots the predicted performance (using the formerly calculated coefficients) against the empirical results for the new landscapes. The correlation between prediction and observation is still very high (0.923), suggesting good generalization.

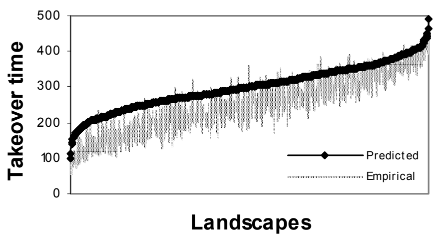


Figure 6. The predicted performance against the empirical performance of the test group (1000 random valid and invalid landscapes). In both cases the empirical plot is sorted. For each landscape represented by the empirical plot, the black solid curve shows the value predicted by the linear model.

Finally, figure 7 plots the performances of the different landscapes sorted from the worst to the best against the performance of the negation of those landscapes. We can clearly see the symmetry predicted by theorem 1 with respect to the performance over a random landscape ($P(V_0)$ in equation 10). This was measured as the average performance of 100 random landscapes. The correlation between the landscapes and their negations is -0.775.

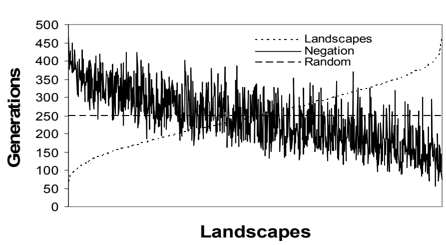


Figure 7. Empirical plot of the performance over all possible valid landscapes. The continuous line represents landscapes. The dotted line gives the performance on the negated information landscape for each point in the previous line.

4.2 Realistic scenario

In the previous subsection we conducted a through analysis, but on a very small landscape. The results strongly support the theorem developed in the previous section. In this subsection we verify that these results hold for a more realistic scenario.

The size of the search space in this experiment is 16 bits (65,536 points), the mutation rate is 0.15, the population size is 200 and the maximum number of generations is 600. This time as a performance measure we use the average number of generations required to find the global optimum.

Naturally, for a landscape of such a size we could not repeat the previous experiment where we explored all possible information landscapes. However, we can still verify whether theorem 1 holds. In order to do so, we measured the performance of a GA over 100 problems with an increasing level of difficulty (multi modality). For each level of difficulty (number of local optima), figure 8 plots the average of 500 different problems against the average of the negation of those problems. Theorem 1 (equation 10) predicts a symmetry between a landscape and its negation with respect to the performance over a random landscape. The performance over a random landscape was measured as the average of 20 random landscapes of the same size. The correlation coefficient, -0.96, clearly shows that the symmetry exists.

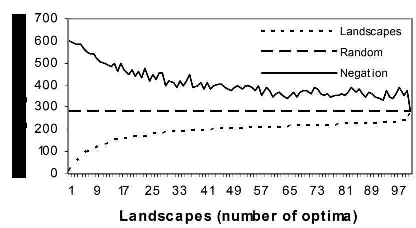


Figure. 8. Empirical plot of the performance over 100 selected problems. The dotted line represents landscapes. The solid line gives the performance on the negated landscape for each point in the previous line.

5. Discussion

A search algorithm uses *a priori* knowledge or assumptions about a problem in the hope to solve it in an efficient way. If a problem is not consistent with those assumptions, the performance cannot be expected to be good. Our notions of performance and information landscapes allow us to make this intuition explicit. A search algorithm is an information processing procedure, a problem (information landscape) is its input and a solution is its output. If the problem does not have enough information, the search algorithm will not be able to perform, to some degree, better than random search. If the problem has useful information (to guide the algorithm) the search algorithm will perform better than random search. Deceptive information (one that leads the algorithm away from the global optimum) will cause the performance to be *worse* than random search.[3]

[3] Any realistic landscape is far too big to be searched exhaustively. Usually, only a logarithmic fraction of it is actually sampled. Deceptive information may lead the search away from the global optimum for

The information landscape was introduced as a new concept of landscape. It enables to measure explicitly the amount of information present in a landscape through a simple notion of distance. Furthermore, given an "optimal" (or alternatively an empirically known easy) landscape, the amount of *positive* information can also be measured.

To emphasize one of the most important differences between fitness landscapes and information landscapes we quote Stadler and Stephens [19] again: "When considering evolution under the effect of multiple genetic operators the fitness that traditionally appears in a mathematical representation of selection is survival fitness or viability. Its role and importance relative to other operators, such as mutation and recombination, must be carefully considered". Unlike the fitness landscape, our information landscape framework explicitly separates the selection operator from the others. The role of selection is to choose between one or more competing solutions according to the probability that the combined effect of the other operators using this solution as an input will provide (eventually) the optimum.

The information landscape throws away some of the information available in the fitness landscape. In particular it ignores the absolute fitness values. However, the information matrix represents a partial relative order of the search space. It includes, for example, all the information necessary to do selection for tournament selection with tournaments size 2. Also when the information landscape is induces by a fitness function, the matrix defines implicitly the absolute rank of each solution and so it includes all the information available to tournament selection with other tournaments size as well as rank selection**.** We postulate that an information landscape is also a good first-order approximation of the information available to search algorithms that use absolute fitness values, like a GA with fitness-proportionate selection.

An information landscape represents a problem. A problem can be difficult for one search algorithm and easy for another. The performance landscape represents an algorithm. It can be used in order to interpret the information embedded in the information landscape. It also indicates that the *quality* of the information is a "subjective" measurement. We gave an empirical method to calculate the performance landscape for any algorithm. This in turn enabled us to accurately predict the performance of a GA over all possible (small) landscapes using a very simple model – without running the GA.

Clearly, the performance landscape cannot be calculated for any realistic problem. However, in [21] we showed that even a performance landscape of a small size is sufficient in order to infer *general* properties of a search algorithm. In particular by clustering the coefficients in the performance landscape we can derive the effective neighborhood structure of an algorithm.

Moving toward realistic landscapes in [20] we developed a GA-hardness measure based on the information landscape alone. That is, we predicted the GA hardness of a problem without having to go through the computationally expensive process of calculating the performance landscape. In particular, we estimated the optimal (easiest) problem for a 12 bit landscape. The distance from this landscape was used as an indicator to the hardness of a problem.

most initial starting points. So, from the fitness evaluations point of view, averaged over multiple runs, an algorithm's performance can be worse than the performance of random search.

In future work we intend to extent this framework to deal with landscapes with more than one global optimum. Furthermore, we intend to explore how the quality of the prediction will change for different target solutions.

The growing number of new metaheuristics makes it more and more difficult to choose between them. We believe the robust and general definition of problem that the information landscape supplies will allow us to use this framework to compare different algorithms in a consistent way. This will also be the subject of future work.

6. Acknowledgment

This work was supported by the Anglo-Jewish-Association and the Harold Hyam Wingate Foundation.

7. REFERENCES

[1] C. Blum and A.Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv. 35(3): 268-308 2003.

[2] D. E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Morgan Kaufmann, 1989

[3] S.Forrest and M.Mitchell. Relative Building Block Fitness and the Building Block Hypothesis. Foundations of Genetic Algorithms 2. Morgan Kaufmann, San Mateo, CA, 1992.

[4] Y.Davidor. Epistasis variance: A viewpoint on GA-hardness. Foundations of Genetic Algorithms, Morgan Kaufmann, 1991.

[5] B.Naudts..Measuring GA-hardness. Ph.d thesis. UV. Of Antwerp. Belgium, 1998.

[6] B.Naudts and L. Kallel, Comparison of summary statistics of fitness landscapes,IEEE Trans. Evol. Comp. V.4.1:1-15, 2000

[7] J.J.Grefenstette. Deception considered harmful. Foundations of Genetic Algorithms 2,Morgan Kaufmann: 75-91 1993

[8] T. Jansen. On Classifications of Fitness Functions. Reihe CI 76/99, SFB 531, Universität Dortmund, 1999

[9] H. Guo and W.H.Hsu. GA-hardness Revisited. GECCO 2003, Lecture Notes in Computer Science 2724, 1584-1585 Springer 2003.

[10] Christian M. Reidys Peter F. Stadler. Combinatorial Landscapes. SIAM Review 44: 3-54 2002.

[11] D. E. Goldberg. Making genetic algorithm fly: a lesson from the Wright brothers. Advanced Technology For Developers, 2 pp.1-8, February 1993.

[12] S. Rana. Examining the Role of Local Optima and Schema Processing in Genetic Search. PhD thesis, Colorado State University, 1998.

[13] L. Kallel, B. Naudts, and C. R. Reeves. *Properties of fitness functions and search landscapes*. In L. Kallel, B. Naudts, and A. Rogers, editors, Theoretical Aspects of Evolutionary Computing, 175--206. Springer, Berlin, 2001.

[14] T.Jones and S.Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In Larry Eshelman, editor, Proceedings of the Sixth International Conference on Genetic Algorithms, pages 184-192, San Francisco, CA, 1995.

[15] L.Altenberg. Fitness Distance Correlation analysis: An instructive counter-example. ICGA, pp 57-64, 1997.

[16] S.A. Kauffman. The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, Oxford, 1993.

[17] L.Altenberg. NK fitness landscapes. Handbook of Evolutionary Computation Section B2.7.2 . 1997.

[18] C.R.Reeves. Landscapes, operators and heuristic search. Annals of Operational Research, 86 473-490, 1999.

[19] P.F.Stadler and C.R.Stephens. Landscapes and Effective Fitness. Comments Theori. Biol. 8: 389-431, 2002.

[20] Y.Borenstein and R.Poli. Information Landscapes and Problem Hardness. In Proceedings of the Genetic and Evolutionary Computation Conference, ACM, 2005.

[21] Y.Borenstein and R.Poli. Fitness distribution and GA hardness. PPSN VIII, Lecture Notes in Computer Science 3242 Springer 2004

[22] Y.Borenstein and R.Poli. Information Landscapes and the Analysis of Search Algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference, ACM, 2005.