Information Landscapes and Problem Hardness

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

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

ABSTRACT

In [20] we introduced a new concept of a landscape: the *information landscape*. We showed that for problems of very small size (e.g. a 3-bit problem), it can be used to generally and accurately predict the performance of a GA. Based on this framework, in this paper we develop a method to predict GA hardness on realistic landscapes. We give empirical results which support our approach.

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

For over a decade GA researchers have attempted to predict the behavior of a GA in different domains. The goal is to be able to classify problems as hard or easy according to the performance a GA would be expected to have on such problems, accurately and *without actually running the GA*.

The Building Block (BB) hypothesis [1] states that a GA tries to combine low, highly fit schemata. Following the BB hypothesis the notion of deception [1], [2] isolation [3] and multimodality [4] have been defined. These were able to explain a variety of phenomena. Unfortunately, they didn't succeed in giving a reliable measure of GA-hardness [5], [6].

Given the connection between GAs and theoretical genetics, some attempts to explain the behavior of GAs were inspired by biology. For example, epistasis variance [7] and epistasis correlation [8] have been defined in order to estimate the hardness of a given real world problem. NK landscapes [9], [10] use the same idea (epistasis) in order to create an artificial, arbitrary, landscape with a tunable degree of difficulty. These attempts, too, didn't succeed in giving a full explanation of the behavior of a GA [6], [11], [12].

Finally, fitness distance correlation [13] tries to measure the intrinsic hardness of a landscape, independently of the search algorithm. Despite good success, fitness distance correlation is not able to predict performance in some scenarios [14].

The partial success of these approaches is not surprising. Several difficulties present themselves when developing a general theory that explains the behavior of a GA and is able to predict how it will perform on different problems.

A GA is actually a *family* of different algorithms. Given a problem, the GA designer first decides which representation (e.g. binary, multiary, permutation or real numbers) to use, then how to map the solution space into the search space, and finally which operator(s) (mutation, crossover) to use. Moreover, there are limited concrete guidelines on how to choose a representation and a genotype-phenotype mapping. Indeed this is a very difficult task. Different genotype-phenotype representations can completely change the difficulty of a problem [15]. There have been attempts to evolve the right representation [16] and there are some general design guidelines [15], [17], [18]. However, the reality is that the responsibility of coming up with good ingredients for a GA is still entirely on the GA designer.

In the absence of a good, predictive theory of GA performance, unavoidably we are only left with an experimental approach. The idea is to divide the space of real-world problems into GA hard and GA easy by experimentation. This, however, first requires finding a good GA (with its representation, mapping, and operators) for every specific instance (or class of instances) of a problem.

The information landscape framework was introduced in [20] as an alternative interpretation to the concept landscape. In particular this allows defining a simple measure of distance between landscapes. In this paper we suggest to use this notion as a predictive measure to problem difficulty. The bigger the distance between a landscape and an "optimal" landscape the harder to search the landscape will be.

The paper has the following structure. We start by giving an overview of the information landscape perspective. This is followed by a theoretical motivation of a new measure of problem difficulty. We conclude with empirical results and a discussion.

2. Background

In [20] we proposed a redefinition of the concept of landscape that makes the quantity and quality of the information available to guide a search algorithm explicit. This is why the new landscape was called an *information landscape.*

The performance of any search algorithm on any particular information landscape can be approximated. In order to do so, we introduced the notion of *performance landscape*, which was then used to predict the performance of a GA over landscapes of a very small size (all 3-bit problems).

Since the work in [20] is the starting point for this paper, in the next sub-sections we define the notions of information and performance landscape and discuss interpretations of the two concepts.

2.1 Information Landscapes

An *information landscape* is a triple (X, χ, t) including: 1) a set of configurations X, 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 $|X| \times |X|$ 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 landscape* to denote M without ambiguity.

The notion of information landscape does not require the availability of a fitness function. However, when a fitness function f is available, we should normally assume:

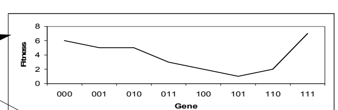
$$t(x_i, x_j) = \begin{cases} 1 & if\ f(x_i) > f(x_j) \\ 0.5 & if\ f(x_i) = f(x_j) \\ 0 & otherwise \end{cases} \tag{1}$$

If the fitness function is noisy, t can take values other than 0, 0.5 and 1. Given the information landscape we can construct the following rank-based fitness function:

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

Note that not all 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 cannot be derived from a corresponding fitness landscape.

Figure 1 gives an example of a fitness function, a landscape defined over a real neighborhood structure and the matrix which represents our *information landscape* for a bit-string configuration space.



(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 1. 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 (information landscape).

Since $t(x_i, x_j) = 1 - t(x_j, x_i)$ the matrix (figure 1) presents symmetries with respect to the diagonal; the gray area marks the independent elements of the information landscape. Diagonal elements (omitted for clarity) are all 0.5. Moreover, we exclude the entries related to the optimum. We assume that we have a way to identify it, hence once it is found, the search is over.

In order to account for all this in a simple way we use a vector to store the relevant entries in the matrix:

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

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

This definition of a landscape allows us to easily define the distance between two landscapes. Let V_a, V_b be two information landscapes, the distance between them is defined as:

$$d(V_a, V_b) = \frac{1}{n} \sum \left| v_{a_i} - v_{b_i} \right| \tag{3}$$

In addition we are in a position to *quantify* the amount of information present in a landscape. The *degree* $d^{0.5}$ of the *information landscape* is the degree to which the information in the matrix available to an algorithm is different from 0.5. Formally, it is the distance between a landscape and the landscape where all matrix elements are 0.5 normalized to the range [0,1]:

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

2.2 Performance Landscapes

Let $P : V \rightarrow \Re$ be a performance measure over the landscape. For example, P could be the number of fitness evaluations required to find the global optimum.

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.[1] As an approximation for P, in [20] we adopted an n-variate linear function of the form

$$P(V) \cong c_0 + \sum c_i (v_i - 0.5) \tag{5}$$

and we used multivariate linear regression to estimate the coefficients. We then defined the array $C = (c_i)$ as the *performance landscape.*

In [20] we indicated how, for a given performance landscape C and a degree of information d_0, we should expect our algorithm to provide best performance on the following information landscape:

$$V_{\max} = \left(\arg\max_{v_i} [c_i (v_i - 0.5)] \middle| d^{0.5}(V_{\max}) = d_0 \right) \tag{6}$$

2.3 Interpretation

It is important to understand how an entry v_i in the information landscape and the corresponding coefficient c_i in the performance landscape are related to the performance of an algorithm. (See Fig. 2.)

[1] The training set includes examples of the form (V,P), V being an information landscape and P being an estimate of $P(V)$ obtained by running an algorithm on V and measuring performance.

The assumption underlying most optimization algorithms is that applying the search operators to solutions with high fitness (as opposed to ones with low fitness) is more likely to yield solutions close to the optimum (we only consider optimizations problems). In [20], we termed 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 a GA, given a particular population, the effective distance of a string from the optimum would be the probability that given that this string is selected into the mating pool, the optimum will be found during the run. The effective distance is only a function of the search operators and the neighborhood structure.

The fitness of a solution is not related to the effective distance of the solution from the optimum. However, the algorithm uses the fitness of the solution as an indicator for such a distance. The performance of the algorithm depends on the correlation between the *relative fitness* (i.e. the *information* given by the fitness function) and the effective distance from the optimum

An entry in the *information landscape* represents therefore the *assumed* relative effective closeness to the optimum (i.e. if $m_{i,j}=1$ solution "i" is closer to the optimum than solution "j"). Each element of the *performance landscape,* on the other hand, represents the degree to which the effective distance of one solution is closer to the optimum than another. In other words, the information landscape states *which* solution should be assumed to be better whereas the performance landscape states *whether indeed and by how much* a solution is better than another for the purpose of eventually solving a problem.

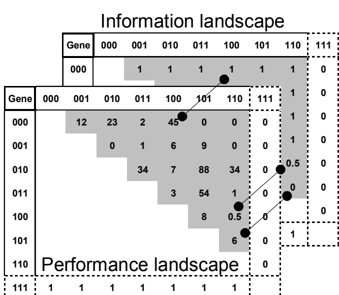


Figure 2. Relation between the information landscape and the performance landscape. High values in the performance landscape indicate that the corresponding entries in the information landscape are important.

3. Doing without the performance landscape

In section 2 we introduced our information landscape framework and reviewed the main results of [20], arguing that the performance of the algorithm can approximately be represented by a *performance landscape* (see [20] for empirical results strongly corroborating this). However, calculating a performance landscape is a computationally expensive process. This prevents the estimation of performance landscapes for realistic problems.

In this paper we want to remove this obstacle and show that the information landscape alone can be used as an estimator to GA hardness. In particular, in this section we will show that the distance between a landscape and a suitably defined "optimal" landscape is a good indicator of problem hardness.

Our argument is simple. The entries of the information landscape represent assumptions about the relative effective closeness of solutions to the optimum. Naturally, we should expect the performance of an algorithm to depend on the number of such assumptions that turn out to be correct.

In this respect the *optimal information landscape* is one where all assumptions are correct. Under the linear approximation in Equation (6), such a landscape, which we will call $V_{\max}$, would be the one given in Equation (7). However, for our purposes $V_{\max}$ could also be determined (or, more likely, approximated) empirically. Once $V_{\max}$ is available, then the distance between $V_{\max}$ and a given landscape V is proportional to the number of wrong assumptions in such a landscape. So, *$d(V,V_{\max})$ can be used as an indicator to problem difficulty*.

This measure of hardness is not exact because different entries in the performance landscape may have different values, i.e. some assumptions are more important than others. However, the error is bounded. In order to show this we rewrite the performance as a function of the distance of a landscape from the optimal landscape and a (bounded) quantity which depends on the elements where the two landscapes differ.

***Lemma 1*:** Let $V_{\max}$ be the optimal information landscape and V' any arbitrary landscape of degree $d(V')=1$. Then:

$$P(V') \cong P(V_{\max}) - n(d_{V'}\bar{c}_{V'}) \qquad (7)$$

where:

$$d_{V'} = d(V_{\max}, V'), \quad n = |V'|,$$

$$\bar{c}_{V'} = \frac{1}{nd_{v'}} \sum_{i:v_{\max_i} \neq v'_i} |c_i|, \quad \min(C_i) \le \bar{c}_{V'} \le \max(C_i)$$

Proof: Using Equation (5) we obtain

$$P(V_{\max}) - P(V') \cong \sum c_i (v_{\max_i} - v'_i)$$

$$\forall_{V',i}, c_i(v_{\max_i} - v'_i) > 0$$
$$\Rightarrow c_i(v_{\max_i} - v'_i) = |c_i(v_{\max_i} - v'_i)| = |c_i||v_{\max_i} - v'_i|$$
$$\Rightarrow \sum c_i(v_{\max_i} - v'_i) = \sum |c_i||v_{\max_i} - v'_i|$$

Let $C_{1_i} \equiv \left\{ |c_i| \,\middle|\, v_{\max_i} \neq v'_i \right\}$ since $|v_{\max_i} - v'_i| \in \{0,1\}$

$\sum |c_i||v_{\max_i} - v'_i| = \left|C_{1_i}\right| \bar{C}_{1_i}$ but $|C_{1_i}| = nd_{V'}$ and $\bar{C}_{1_i} = \bar{c}_{V'}$ therefore,

$P(V') \cong P(V_{\max}) - n(d_{V'}\bar{c}_{V'})$

***Theorem 1*:** For two random information landscapes V_1, V_2 of degree $d^{0.5}(V_1)=d^{0.5}(V_2)=1$ and such that $d_1=d(V_{\max},V_1)<d(V_{\max},V_1)=d_2$ for any two pre-fixed d_1 and d_2 the following holds

$$E[P(V_1) - P(V_2)] \propto (d_2 - d_1)$$

Proof: From Lemma 1 we obtain: $P(V_1) - P(V_2) \cong n(d_{V_2}\bar{c}_{V_2} - d_{V_1}\bar{c}_{V_1})$. Since V_1, V_2 are random landscapes, the averages $\bar{c}_{V_2}, \bar{c}_{V_1}$ are random variables representing sample means of C_i. So, they are unbiased estimators of the true mean $\mu = E[C_i]$. Therefore:

$$E[P(V_1) - P(V_2)] \cong n\mu(d_{V_2} - d_{V_1})$$

An obvious consequence of this is the following

Corollary 1: For any two landscapes V_1, V_2 with degree of information 1

$$P(V_1) > P(V_2) \Leftrightarrow d_{V_2} > d_{V_1}$$

These two results allow us to compare the performance of an optimisation algorithm on any given information landscapes of degree 1 without the need to know the performance landscape.

We can apply a simple heuristic in order to extend these results to landscapes of degree smaller than 1.

Let v_k be an entry in the information landscape with a value of 0.5. Let $v_{\max_k}$ the corresponding entry in the optimal landscape. From the algorithm's perspective, a 0.5 probability means that the algorithm will behave (for one time step) either as if $v_k = 1$ or as if $v_k = 0$ with equal probability. That is, v_k has the same probability to be equal or not equal to $v_{\max_k}$. So, over multiple time steps, v_k will some time drive the search towards the optimum, some times away from the optimum. However, we argue that the performance of the algorithm when v_k=0.5 will be in between the performance obtained when v_k is *constantly* equal to $v_{\max_k}$ and the performance obtained when v_k is constantly equal to 1-$v_{\max_k}$.

In summary we should expect $E[P(V_1) - P(V_2)] \propto (d_2 - d_1)$ even for landscapes of degree smaller than 1.

4. Evaluating hardness

In the previous section we argued that the hardness of a problem can be estimated using the distance of its information landscape from the optimal landscape. However, in general the optimal landscape is not known. Naturally, instead of using the actual optimal landscape, we could use an approximation, but which one? We know from many empirical studies that unimodal problems tend to be GA easy, the onemax problem being a glowing example. Indeed, in [21] we were able to predict an optimal landscape according to our framework, and that landscape was unimodal. So, in this work we decided to use a unimodal landscape as an approximation of the optimal landscape.

The *information landscape* and the performance function are defined for a *fixed* target solution. If we change the target solution the same information landscape can change from being easy to being difficult (e.g. consider the information landscape induced by the onemax function where we change the optimum to being the string 00…0). So, the distance between landscapes must be computed for landscapes with the *same* global optimum. This requires knowing *a priori* the global optimum.

Given the global optimum, we measured the distance between a unimodal landscape and the actual landscape induced by the problem. In all our experiments we calculated the exact distance between the two landscapes. We used a simple GA with uniform crossover used with 100% probability and mutation applied with 10% probability. The size of the landscape was 14 bits. We used a population size of 20. The first generation in which the optimum was found was used as the performance measure. The results are the average of 100 runs.

This section is divided to four subsections. In the first, we consider examples in the literature for which a notion of information was implicitly studied. In section 4.2 empirical results are given for various problems. In section 4.3 we test our approach on three counterexamples for other measures of problem difficulty. Section 4.4 suggests a way in which our measure can be further improved.

4.1 Information in previous literature

There are various examples in the literature that are related, implicitly, to the notion of *information*. In particular, there are examples of problems with no information (NIAH), problems with abundant and reliable information (onemax) and problems with abundant but unreliable information (deceptive problems).

The onemax and fully-deceptive problems were studied in [21] where, using the information landscape perspective, we showed that one problem can be seen as the opposite of the other. In particular, the landscape induced by onemax is the negation[2] of the landscape induced by a fully-deceptive problem of the same size.

One of the aspects studied in the royal-road problems is the size of the lowest order building blocks, which can vary from 1 (onemax) to n (NIAH). So, this parameter is effectively related to our *degree of information*.

Finally, our framework suggests that the difficulty of a problem depends on the amount of reliable (non-deceptive) information in the landscape. The information landscape resulting for a mixture of problems, from this point of view, will contain a mixture of the (reliable and unreliable) information originally in the two problems. So, a mixture of an easy problem with a difficult one should produce an intermediate difficulty. The same observation was made in [11],[22]. Naudts and Kallel [11] studied a deceptive mixture of onemax and zeromax and a mixture of onemax and nearest-neighbor interaction functions. In [22] Clergue and Collard constructed hard functions for GA by combining two types of misleading functions.

These examples show that the study of different aspects of information is not new. The main contribution of our framework is the ability to consider information explicitly.

4.2 Results

In this subsection we estimate the hardness of problems with no information (NIAH), random information (random problems), maximally reliable information (unimodal) and maximally unreliable information (deception). Furthermore, we study problems with a variable level of difficulty: the NK landscapes [10] with k=1…10, multimodal landscapes with a varying number of local maxima (1-20) and trap functions. Finally, to test our measure of difficulty on landscapes which were not induced by artificial problems, we also considered the performance of 12 random MAXSAT problems. For each problem, we consider only one global optimum. If a problem has more than one global optimum, we choose one at random to be the target solution.

Figure 3 plots the actual performance for our test problems. For easier visualization landscapes were ordered on the basis of their predicted difficulty (from easiest to hardest). The correlation coefficient between observed and predicted difficulty is 0.82.

[2] The negation of a landscape V is the landscape $\overline{V} = \overline{1} - V$. I.e., the looser of a tournament in a landscape is the winner in the other.

Table 1 gives our (predicted) measure of difficulty by problem, showing that our measure fully matches results obtained in previous research. In particular, multimodality is not a good indicator to problem difficulty. For example, a landscape with 3 local maxima has the same expected difficulty as a landscape with 7 local maxima. Moreover, a landscape with 15 local maxima is more difficult than a landscape with 18.

The table confirms that NK model is not appropriate for the study of problem difficulty because problems with a $k>2$ are already too difficult [6]. Indeed, our measure suggests that the difficulty of such landscapes is close to random.

Different instances of the same problem might have different degrees of difficulty in the black-box scenario [6]. The predicted difficulty for different instances of the MAXSAT problems varies from 0.2 (easy) to 0.45 (difficult).

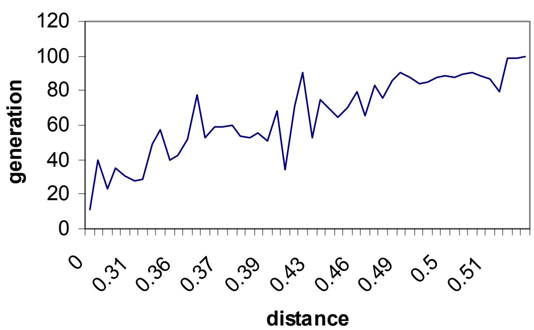


Figure 3 The performance of a GA vs. the predicted difficulty as measured by the distance from a unimodal landscape.

Table 1 Estimated hardness of our test problems.

Distance	**Problem**
0	**MM1**
0.20-0.30	**MM2,MAXSAT**
0.30-0.35	**MAXSAT,NK1**
0.35-0.40	**MM3, MM7,MM5,MAXSAT,NK2**
0.40-0.45	**MM14,MM18,NK3, NK4,Needle,MAXSAT**
0.45-0.50	**MM17,MM13,NK5-9,RAND**
0.50-0.60	**MM15,RAND**
0.60-0.70	**TRAP4**
0.70-0.80	**TRAP3**
0.80-1	**TRAP2,TRAP1**

4.3 Hardness of known counterexamples for other predictive measures

In the previous subsection we showed that for a broad family of problems, our method predicts accurately the problem difficulty. In this subsection we test our framework on three problems where other measures of difficulty have been shown to fail.

As we mentioned above Naudts and Kallel [11] constructed an *easy* problem consisting of a deceptive mixture of one-max and zero-max, where both the fitness distance correlation measure and the sitewise optimization measure (a generalization for the FDC and epistasis suggested in the same paper) failed to correctly predict performance. Moreover, the higher the mixture coefficient the harder the problem, yet none of the predicting measures predicted this. We repeated the experiment[3] with the control parameter n varying from 1 (easy) to 9 (hard). The correlation between the predicted difficulty and the actual performance was 0.75. So, we were largely able to capture the difference in performance as the parameter n varied. (The correlation coefficient of the results obtained in this and the previous subsection was 0.72.)

Jansen [6] showed that the fitness distance correlation of the ridge function is very small. Yet, this is quite an easy problem for a hill climber. The distance of the ridge function to the optimal landscape is 0.84, which indicates a very difficult problem. Indeed the GA was not able to find the solution in 100 generations. A problem that is easy for a hill climber is not necessarily easy for a recombinative GA. Our measure of difficulty is *specific* to the algorithm being used. However, we suspect that for a local search, such as a hill climber, a first order approximation of the performance will not necessarily be sufficient in order to capture the problem difficulty. (See discussion section for further details.)

Jansen [6] gave two counter examples to the bit-wise epistasis measure of difficulty. The first one was the NIAH which we already discussed extensively before. The second was the leading one function. The distance of the leading one function from the optimum is 0.36, which predicts well its performance.

4.4 Improving the prediction

In the previous subsection the framework was tested explicitly on counterexamples given for other measures of difficulty. In this subsection we use results obtained other work to enhance the accuracy of our measure.

In [21] we showed that the performance landscape can be used in order to analyze properties of a search algorithm. In particular we showed that for a GA with a uniform crossover, the values of the entries in the performance landscape were proportional to the relative distance of each solution from the global.

Based on this knowledge we assigned a weight to each entry in the information landscape. The weight was equal to the relative distance of the two solutions from the optimum. Then we reassessed problem difficulty using the *weighted distance* between landscapes.

The correlation for the first set of problems (Section 5.1) was improved from 0.82 to 0.86, while that of the *deceptive mixture* was improved from 0.75 to 0.78. Also, the combined overall correlation improved form 0.72 to 0.78.

5. Discussion

The most important feature of the information landscape framework is to view problem difficulty from the perspective of information. In the first section 5.1 we discuss this.

[3] The GA used in [11] is different from the one used here.

The experience of the previous unsuccessful attempts to construct an accurate predictive measure of difficulty suggests that we must state explicitly what exactly we are trying to achieve with our own. We do this in Section 5.2.

Then, in Section 5.3, we turn to a discussion on the empirical results reported in this paper, and on what accuracy we should expect to observe in other problems not tested here.

In Section 5.4, we conclude with a discussion on the possible implication of this work in the study of other search algorithms.

5.1 The information landscape perspective

In this paper a new predictive measure to GA hardness was introduced. It is based on the concept of *information.* A problem can be assessed according to the *quantity* and *quality* of the information it has. The quality of the information is defined only with respect to a specific algorithm.

The easiest problem is one that has *maximum* amount of *reliable* information. In this paper, we used the onemax as a reference which approximates such a problem.

The difficulty of other problems was predicted according to the *amount* of *reliable* information they had. This amount was estimated as the distance from the onemax problem.

The quantity of absolute (reliable and unreliable) information was not measured explicitly. However, this is implicitly considered when computing the distance from the optimal landscape. The distance of a landscape which contains no information (NIAH) is exactly 0.5[4]. The extent to which the performance of an algorithm is good or bad depends on by how much the distance differs from 0.5 (e.g. a distance of 0.3 indicates a landscape less random than one at distance 0.6 but more random than one a distance 0.9 from the optimum landscape).

In section 4 we gave empirical evidence that support our notion of problem hardness. Easy problem were closer to the optimal landscape (i.e, their distance was smaller than 0.5). The distance of hard problem was bigger than 0.5.

5.2 The objective

The experience with past attempts teaches us that it is not feasible to give a *general* predictive measure of problem difficulty.

From theoretical perspective, such a measure should assess hardness irrespectively of the search algorithm. However, different search operators induce different landscapes [23] and, even when considering only GAs, the variety of operators and parameters available makes it impossible to encompass all of them with a single measure of difficulty. This is why our measure of difficulty is defined over a specific search algorithm (i.e. specific operators, parameters and neighborhood structure).

From a practical perspective, the main motivation for developing methods to assess problem difficulty is to be able to classify problems as hard or easy according to the performance a GA would be expected to have on such problems, accurately and *without actually running the GA*. In practice, none of the existing measures can accurately predict performance. Also, most of them require full knowledge of the search space and those that don't give inaccurate results. Indeed, given that a predictive measure for a simple hill climber is by itself PSPACE-complete [6], finding a single, good one for GAs (which are more complicated) would seem unlikely. This is why some researchers have suggested focusing on empirical results rather than theoretical analyses.

Even though we agree with this argument, we do not agree with the conclusion. Firstly, we suspect it might be easier to assess problem difficulty for stochastic algorithms, like GAs, than for deterministic ones, like a simple hill-climber. This is because the latter is much more sensitive to noise, and, so, predicting its performance based on sampling may be very difficult.

Secondly, we believe that some of the existing metaheuristics might be overly complicated. The lack of a strong theoretical background makes it difficult to understand which operators are essential and which are not. We think it is important to attempt to construct predictive measures of problem difficulty to contribute towards this understanding. These may provide useful knowledge on the actual underlying logic of an algorithm (e.g. epistasis variance).

If an accurate predictive measure is found, it will be possible to construct a more efficient search algorithm. An objective of this line of research, therefore, is to refine a search algorithm itself, making it more efficient, rather than just assessing the difficulty of a problem with respect to the search algorithm.

From this point of view, the dependency of most of the predictive measures (including ours) on full knowledge of the search space is not a draw back.

We believe that the information landscape perspective is an important step towards this direction. It explicitly defines the information embedded in a landscape and the relationship between this and the performance of an algorithm, giving the opportunity to study problems and search algorithms from a unified and radically new perspective.

5.3 The empirical results

The results obtained in this paper strongly support our notion of difficulty. In particular our measure was able to predict GA performance for problems where other measured had failed (section 4.2).

The notion of GA easy or hard problems is certainly ill-defined [14]. It is not clear how to compare the performance of different algorithms.

Our approach to overcome this problem was to define difficulty *with respect to a reference point*: the optimum landscape. The performance over a random landscape (with distance 0.5 from the optimum landscape) is our threshold to separate easy and difficult problems. The upper and lower limits for the possible performances are represented by the performance of the GA on the optimal and worst landscapes.

We define difficulty *with respect* a specific algorithm. The performance values obtained for optimal, random and worst landscapes differ from algorithm to algorithm. However, we believe that in the future we might be able to normalize these reference points so that different algorithm can be compared.

In spite of strong empirical support, we can not claim, at this stage, that our measure of difficulty is precise in all cases.

---

[4] In this paper we used the onemax as an approximation for the optimal landscape. Therefore the actual distance for NIAH was smaller than 0.5.

Experiments in more scenarios will be needed to assess generality and robustness of this measure.

5.4 Implications for other search algorithms

It might appear that, similarly to the fitness distance correlation, our predictive method attempts to predict the performance of a GA without considering its properties. This is not the case.

Our method is based on the comparison between a landscape and a reference optimal one. That is we compare a landscape with a landscape which is known to be good *for the specific algorithm* under a *specific representation*.

We think this is the right thing to do. The GA is a family of different search algorithms. Each operator or even the same operators with different parameters (i.e. size of population, mutation rate) might have different performance on the same landscape. Hence, we do not believe all GAs can be assessed accurately with the same predictive measure. Naturally, in some cases, if the algorithms are sufficiently close, the same predictive measure could be used as a good approximation.

Our predictive measure is tailored to each algorithm. But to which extent is it general? In order to answer this question we have to consider the underlying assumption of our model.

Firstly, we assume that all the information that the algorithm uses is captured in the *information landscape*. As stated in section 2 this is not the case for search algorithms that use the absolute values of the fitness. However, by and large it is true for many modern algorithms and, therefore, we hope this assumption will not limit the applicability of our theory.

Secondly, we assume that a first order, linear approximation is sufficient to describe the properties of the information landscape. As mentioned on section 4.3, this might not be the case for deterministic search algorithms such as a classical hill climber. This is because in such an algorithm, the interdependencies between the entries of the information landscape might have a tremendous impact on the probability to find the solution. The ridge function is a good example of that. However, motivated by the result presented in this paper and in [21], we think that a linear approximation works quite well for realistic stochastic population based algorithms. Naturally, the generality of this assumption needs to be validated by extending our work for different metaheuristics.

6. Conclusion

In section 3 we have shown that the *information landscape* framework predicts that the distance of a problem from the optimal landscape is a good indicator of its difficulty.

This is perhaps the most intuitive and simple way of assessing hardness. The further a landscape is from the optimal landscape the worse it becomes. It might seem overly simplistic, but the results obtained in section 4 suggest otherwise.

In section 5 we presented the possible contribution of the new approach to the research of search algorithms in general and GAs specifically. We considered both its potential and its limitations.

Bridging the gap between theory and practice is becoming one of the main goals of current research. From this perspective many researchers (e.g. see [12][6]) suggest to focus more on empirical results for specific classes of problems rather than a general predictive measure.

Our predictive measure can be viewed as being half way between theory and practice. We base our prediction on a good *reference* landscape. In this paper, we used a unimodal landscape, but any known (empirically "proven") good landscape can be used. Furthermore, the local properties of a particular problem (not necessarily an easy one) can be studied simply by taking it as a reference point (see [20] for further details). In the future this may allow incorporating knowledge gained from empirical experience into a general theoretical framework.

7. Acknowledgment

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

8. REFERENCES

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

[2] S.Forrest and M.Mitchell. Relative Building Block Fitness and the Building Block Hypothesis. In FOGA 2. Morgan Kaufmann, 1992.

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

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

[5] J.J.Grefenstette. Deception considered harmful. In Foundations of Genetic Algorithms 2, D. Whitley (Ed.), San Mateo, CA: Morgan Kaufmann, 1993.

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

[7] Y.Davidor. Epistasis variance: A viewpoint on GA-hardness. In G.J.E Rawlines, editor FOGA-1,pp. 23-55. Morgan Kaufman 1991.

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

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

[10] L.Altenberg. NK fitness landscapes. Handbook of Evolutionary Computation 1997b.

[11] B.Naudts and L.Kallel, Comparison of summery statistics of fitness landscapes, IEEE Trans.Evol.Comp. 2000.

[12] H. Guo and W.H.Hsu. GA-hardness Revisited. GECCO 2003 Poster paper.

[13] 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 CEC, 1995.

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

[15] F.Rothlauf. Representations for Genetic and Evolutionary Algorithms. Studies in Fuzziness and Soft Computing, Volume 104. Heidelberg: Springer, 1st edition 2002.

[16] L.Altenberg. Evolving better representations through selective genome growth. Proceedings of the 1st IEEE Conference on Evolutionary Computation. Part 1, 1994.

[17] A.Moraglio and R.Poli. Topological interpretation of crossover. In Proceedings of GECCO, Springer, 2004.

[18] N.J.Radcliffe. Equivalence class analysis of genetic algorithms. In Foundations of Genetic Algorithms 3. Morgan Kaufmann, 1994.

[19] H. Huttel. On Rice's Theorem. Aalborg University. 2001.

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

[21] Y.Borenstein and R.Poli. Information Landscapes and the Analysis of Search Algorithms. In Proceedings of GECCO, ACM, 2005.

[22] M.Clergue, P.Collard. GA-hard functions built by combination of trap functions. In Proceedings CEC2002, IEEE Press. 2002.

[23] C.R.Reeves. Landscapes, operators and heuristic search. Annals of Operational Research, 86 473-490, 199