No Free Lunch, Kolmogorov Complexity and the Information Landscape

Abstract- The permutation closure of a single function is the finest level of granularity at which a no-free-lunch result can hold [1]. Using the information landscape framework which was introduced in [2], we are able to identify the unique properties of each closure. In particular, we associate each closure with the amount of information its members contain. This poses a boundary on the expected performance of the algorithm on members of that closure. Moreover, we identify a misconception in the way the Kolmogorov complexity of a landscape is measured. We suggest measuring it in a new way. This allows us to associate each permutation closure with a particular Kolmogorov complexity.

1 Introduction

Wolpert and Macready [3] put an end to the hope of developing a general-purpose optimisation algorithm by proving that such an algorithm does not exist. Over all possible problems, all the algorithms have the same performance.

This led to an ongoing debate about the usefulness of NFLT. In particular, the main argument against the NFLT is it generality: it can be applied only when considering all possible problems [4]. Hence, the problems on which some algorithms may fail will be, on most cases, artificial. On the other hand, a simple example of non-artificial functions where some heuristics are bound to perform badly is given in [5].

One of the attempts to resolve this argument is to find a subset, smaller than the set of all possible problems on which the NFLT holds. It was proven in [1] that the permutation closure of any function is both sufficient and necessary for no-free-lunch results to hold. This was termed in [6],[7] as block uniform functions.

The properties of these sets are of particular interest to the understanding of the practical implication of the NFLT. English [7] argues that the NFLT should be studied in the more general framework of conservation. Schumacher, Vose and Whitley argue [1] that the NFLT results are independent from whether or not the set of functions (within a block) is compressible.

Compressibility or Kolmogorov complexity is associated with a string. It is defined by the length of the shortest program that can generate that string and halts. When the string represents a problem, it is argued that compressible strings are associated with easy problems whereas random strings with difficult ones [8]. English [9],[10] used this framework to establish a connection between the complexity of an algorithm and its potential to solve a complex problem. In addition, observing the high complexity of a typical string (problem) he suggests that a typical search will be, in any case, random.

Despite its frequent use, it is claimed that Kolmogorov complexity cannot be associated with the difficulty of a problem. Consider the needle-in-a-haystack (NIAH). This is a highly compressible landscape and yet difficult to search on. This was the reason that Schumacher et al [1] observed the independency of the NFL results from the compressibility of the closure.

In this paper we use the information landscape framework [2],[11],[12] in order to address some aspects of the NFL. In particular we are able to pose boundaries on the expected performance of the algorithm on functions that are members of the same permutation closure

Moreover, by distinguishing between a landscape and the algorithmic interpretation of the landscape (see section 5) we show that the Kolmogorov complexity of a NIAH is actually high. Hence, the counter example given earlier is not a valid one. This allows us to associate each block with a particular Kolmogorov complexity.

Section 2 introduces the information landscape framework. We use this framework in section 3 to develop a new version of a NFLT. Section 4 gives the contribution of the information landscape framework to the study of the NFL. We illustrate in a simple the different subsets of problems to which the NFLT holds, explain their properties and discuss the possible applications. In section 5 we explain the misconception regarding the way the Kolmogorov complexity of a landscape is measured. We use this result to prove the claims regarding Kolmogorov complexity made in section 4. We conclude with discussion and conclusion.

2 Background

In [2],[11],[12] 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.* In [2] we used it to predict the performance of a GA over landscapes of a very small size (all 3-bit problems). In [11] it was used in order to infer general properties of a search algorithm. Finally in [12] we used the information landscape as a measure of problem hardness in more realistic scenarios.

Section 2.1 introduces a concept which is fundamental to the understanding of this framework. Section 2.2 and 2.3 define the information landscape and the performance landscape respectively. Section 2.4 introduces a theorem which will be the basis of the proof (section 3) of the NFL for information landscapes.

For the sake of simplicity in this paper we consider a landscape with only one global optimum. The objective of the algorithm is to find it.

2.1 Seperation of the global optimum from the cost function

Considering optimisation problems, the performance of an algorithm can be defined as the number of fitness evaluations it takes to find the optimum[1].

The optimum is defined as the solution which has the highest fitness value. This value does not affect the performance of the algorithm[2].

We are interested only in the *relevant* information that a problem contains (i.e. information which affects the search, either negatively or positively). Since the performance does not depend on the fitness of the optimum, we consider the cost function and the target solution separately. Figure 1 gives an example of two arbitrary fitness functions. Usually, the optimum for function (a) is solution B whereas that of function (b) is D. However, using our notation we define separately that solution D is the target solution for both. Since the actual fitness of D does not affect the performance, an algorithm running on the two functions should give *exactly* the same results (i.e. the expected number of fitness evaluation it takes to find solution D will be the same).

This separation might look artificial. However it allows us to study the connection of the global optimum to the topology of the space. We can study, given a neighbourhood structure, for example, the topological properties of function *a* (figure 1) irrespectively of the identity of the global optimum. Then, for *any* possible target solution, we can approximate the difficulty of the problem based on how well it connects to the topological properties induced by the function.

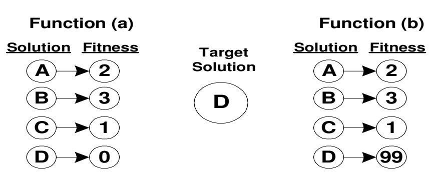


Figure 1. The decomposition of a fitness function. Usually, a fitness function defines the optimum. However, in our framework, we define the target solution (optimum) explicitly, irrespectively of its fitness value.

2.2 Information landscape

An *information landscape* is a triple (*X*, χ, *t*) including: 1) a set of configurations *X*, 2) a notion χ of neighbourhood, 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:

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

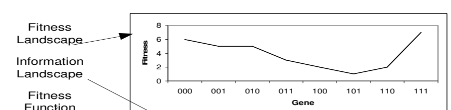
If the fitness function is noisy, *t* can take values other than 0, 0.5 and 1.

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

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

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 neighbourhood 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 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 (information landscape).

Since $t(x_i, x_j) = 1 - t(x_j, x_i)$ the matrix (figure 2) presents symmetries with respect to the diagonal; the grey 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

[1] This can be extended to the more general case by defining performance as the number of fitness evaluations it takes to sample a solution with fitness higher than a threshold

[2] Naturally, it affects the convergence properties of the algorithm however, this is insignificant to this measure of performance.

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:

$$|V| \equiv n = (|X|-1)(|X|-2)/2 \quad (3)$$

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:

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

The degree of information identifies when a problem is difficult due to a lack of information (e.g. the degree of information for the NIAH equals 0) and when it is difficult because it contains deceptive information (for which the degree of information, can, in principal be equal to 1).

2.3 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.[3] As an approximation for *P,* in [2] we adopted an *n*-variate linear function of the form

$$P(V) \cong c_0 + \sum c_i (v_i - 0.5) \quad (5)$$

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

In [2] 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:

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

Note that the smaller the degree of information is the smaller the performance over $V_{\max}$ will be. Also, similarly to $V_{\max}$, $V_{\min}$ can be defined as well. The performance of an algorithm over $V_{\max}$ and $V_{\min}$ is expected to be symmetric w.r.t random search [2].

2.4 The negation of an information landscape

A landscape with all entries equal to 0.5 contains no information. Its equivalent fitness landscape is the needle-in-a-haystack. On this problem any algorithm is expected to perform like random search.

In [2] we defined the negation of a landscape as:

$$\bar{V} \equiv \underline{1} - V \quad (7)$$

where $\underline{1}$ is an information landscape with all *n* entries equal to 1. This simply defines a landscape which, for every tournament between two solutions, gives the opposite outcome of the original landscape. The negation of a flat landscape will still remain a flat landscape.

In [2] we showed that:

Theorem 1: Let V_0 be an *information landscape* with all entries equal to 0.5. Then:

$$P(\bar{V}) - P(V_0) \cong P(V_0) - P(V) \quad (8)$$

Theorem 1 states that the performance of an algorithm on any arbitrary landscape and the negation of this landscape is symmetric w.r.t to random search.

Figure 3 illustrates the possible effect of negating the landscape. Clearly, the hardness of the problem changed from an easy unimodal landscape (left) to a deceptive one (right). Recall that the target solution is not part of the landscape, hence it is not negated. This might look as an extreme example. However, theorem 1 suggests that this is the case for more general scenarios as well (see [12] for more details).

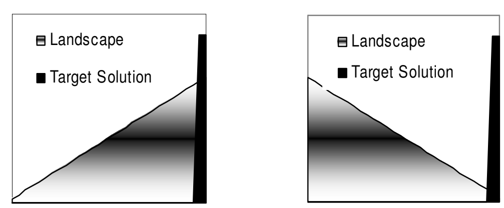


Figure 3. The effect of negating the landscape. In the left the target solution combines with the landscape to create a unimodal function. In the right, the negation of the landscape creates a deceptive problem.

This observation (theorem 1) will be used in order to prove a NFLT using the information landscape framework.

3 No-Free-Lunch for information landscapes

The information landscape theory is based on a different notation than the classical NFL [3]. In particular it is less general and unlike the NFLT it is based on an *approximation* of the algorithm. However, it makes one important further step: it explicitly formulates, in a general way, how the search algorithm processes the information in the landscape. In this section we explain how the different notations connect.

[3] 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.

3.1 The algorithm

An optimisation algorithm a is represented as a mapping from previously visited sets of points to a single new (i.e., previously unvisited) point in X. Let $m<|X|$ be a non-negative integer. Define $d_m = \left\{ (d_m^x(i), d_m^y(i) = f(d_m^x(i))) \right\}$, $i=1,\ldots,m$ where $d_m^x(i) \in X \forall i$ and $\forall_{i,j}\, d_m^x(i) \neq d_m^x(j)$. A search algorithm a therefore is the mapping:

$$d_{m+1}^x(m+1) = a[d_m] \notin \{d_m^x\}.$$

In the information landscape framework the algorithm is not defined explicitly. However, similarly to [5], we assume that it has no preferences of search regions, searched in a stochastic way and prefers to look at nearer neighbours of fit solutions. In other words, we expect the algorithm to search in a "reasonable" way.

3.2 The cost function

The NFL considers the set of all possible cost functions $f : X \rightarrow Y$ where X and Y are finite. The objective, without loss of generality is to seek the global optima of f.

The information landscape considers problems with only one global optimum. The set of all possible problems is divided accordingly to |X| subsets. Each is defined w.r.t a particular global optimum (target solution). Since the algorithm does not have a preference to search regions (section 3.1), it is sufficient to prove a NFLT on one subset.

In addition, rather than considering the fitness of each solution the information landscape considers its *relative* fitness. A problem, in general, is defined as: $f : V \mapsto \{0, 0.5, 1\}$ where V represents the set of all possible tournaments of size 2 in the original search space X.

3.3 The performance measure

Under the oracle-based model of computation any measure of the performance of an algorithm after m iterations is a function of the sample d_m^y. In [3] the measure of performance for a search algorithm is $P(d_m^y \mid f, m, a)$ which is the conditional probability of obtaining a particular sample d_m^y under the stated conditions.

A problem in the information landscape framework is defined w.r.t one particular global optimum. We therefore use the average number of distinct fitness evaluations required to find the optimum as a measure of performance.

Note that the information landscape framework uses the function P (performance landscape) in order to *approximate* the performance of an algorithm (equation 3). [2],[11] and [12] indicates that the approximation is good, however, it is still a first order approximation. Nevertheless, this should not affect the general results (see section 5).

3.4 The no-free-lunch

The NFLT proves that for any two algorithms a_1 and a_2:

$$\sum_f P(d_m^y \mid f, m, a_1) = \sum_f P(d_m^y \mid f, m, a_2)$$

The same results can easily be obtained from the information landscape framework using theorem 1 as follows. Let V_0 be an *information landscape* with all entries equal to 0.5 (i.e. flat landscape). Let a be an arbitrary algorithm $|V|=n$ and $|X|=k$. Notice that the performance of the algorithm on a flat landscape equals to the performance of a random search.

Since each element of V can be 0, 0.5 or 1 (equation 1), the number of possible information landscapes is $N=3^{|V|}=3^n$ therefore:

$$\sum_v P_a(V) = P_a(V_0) + P_a(V_1) + P_a(\overline{V}_1) + \ldots + P_a(V_{\frac{N-1}{2}}) + P_a(\overline{V}_{\frac{N-1}{2}})$$

< note: N is odd> so for Thr 1 $\Rightarrow$

$$= P_a(V_0) + \underbrace{2P_a(V_0) + \ldots + 2P_a(V_0)}_{(N-1)/2} = NP_a(V_0) = \frac{Nk}{2}$$

Since the result is independent of the algorithm a, all algorithms, averaged over all possible information landscapes, have the same performance.

3.5 The sharpen no-free-lunch

Let $f : X \rightarrow Y$ be a function and $\sigma : X \rightarrow X$ be a permutation (i.e. σ is one-to-one and onto). The permutation σf of f is the function $\sigma f : X \rightarrow Y$ defined by $\sigma f(x) = f(\sigma^{-1}(x))$.

Define a set F of functions to be closed under permutation if for every $f \in F$, every permutation of f is also in F.

Schumacher, Vose and Whitley proved [1] that the permutation closure of a single function is the finest level of granularity at which a NFL result can hold.

English [10] named each set F, a block. A distribution which is uniform within each block is called block uniform. The space of all possible problems can be divided into blocks. A block uniform distribution is necessary and sufficient for NFLT in search.

In order to show that the same holds in the information landscape framework, it is sufficient to prove that, for any landscape (function), the negation of that landscape is contained in the permutation closure of the original problem.

Let $f : X \rightarrow Y$ be a fitness function and V be the corresponding information landscape.

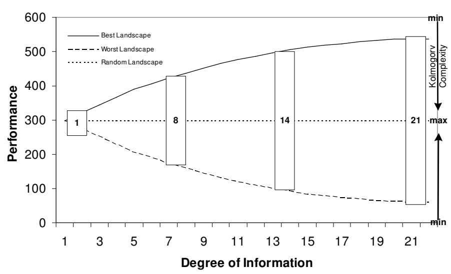
Without loss of generality, let the elements of the set X be ordered in a non descending order (i.e. $\forall_{i>j} x_i \geq x_j$). Let σ be the permutation over X which swaps $x_i \leftrightarrow x_{n-i}$ for all *i*. The function $\sigma f(x)$ induces the information landscape $\overline{V}$. Since the set $\{V, \overline{V}\}$ is closed under the permutation σ, the NFL holds.

4 New insights about the No-Free-Lunch

The no free lunch, put simply, states that if you cannot make any assumption about a problem you can not expect to solve it in an efficient way. No algorithm is expected, averaged over all possible problems, to outperform random search.

It is known that the space of all possible problems can be divided into subsets for which the NFLT holds. It is not clear, however, what are the unique properties, if any, of the different subsets.

The information landscape framework explains the NFL results in a clear way. It shows how the space of all problems is divided into subsets on which the NFLT holds, but more importantly, it explains the properties of these subsets. This is illustrated in the following figure.



8

Figure 4. A unified view of the no-free-lunch. Each rectangle represents a block uniform on which the NFLT holds. The NFL is represented as symmetry around the line of random search. The space of all possible problems is the unification of all the blocks. Each block is associated with a degree of information. This is reflected by a boundary on the performance that all the functions belonging to a block can have. Furthermore, for blocks with lower degree of information the average Kolmogorov complexity is higher.

Figure 4 shows qualitivity plot[4] of the performance of an algorithm over all possible problems of size 8. The information landscape induced from such a landscape is of size 21 - hence there are 21 different degrees of information[5]. For each degree of information, the plot gives the minimum and maximum expected performance of problems belonging to the same groups. For the sake of clarity, other landscapes, apart from the two that gives the best and worst performance are not shown.

Four arbitrary blocks (on which the NFLT holds) are represented as rectangles. Each is associated with a particular degree of information (section 4.1 will prove that this is indeed the case). Note that for blocks associated with lower degree of information: (1) the variance of the performance is smaller and (2) the average Kolmogorov complexity is higher (see section 5).

4.1 A block uniform is associated with a particual degree of infromation

Define a set F of functions to be closed under permutation if for every $f \in F$, every permutation of *f* is also in *F* (see section 3.5). Each block (i.e. a set *F*) can be associated with a particular degree of information. Let $f : X \rightarrow Y, f \in F$ be a function. The degree of information of *f* is:

$$d^{0.5}(f) = \sum_i \binom{|y=i|}{2} \Big/ \binom{|Y|}{2} \qquad (9)$$

Equation 9 counts all possible pairs of solutions which have the same fitness. It follows that the degree of information is defined according to the *fitness distribution* of *f*. Since the permutation operator does not affect the fitness distribution of a function, all the functions which belong to *F* have the same fitness distribution. Therefore, the degree of information for one function can be associated with the whole block.

The properties of a block can be associated to the properties of different degrees of information. This in turn corresponds to amount of information available to an algorithm. The more information an algorithm has, the better the maximum performance (given positive information) can be. On the other hand, if the algorithm is not aligned with the information embedded in the problem the more chances it has to be deceived, hence to have a bad performance (equation 6).

The less information an algorithm has, the more randomised it will be. In the case of the NIAH all the algorithms are expected to have similar performance to random search.

Since the different closures (blocks) corresponds to different degrees of information they are expected to differ in the variance of the expected performance of their members.

5 Kolmogorov complexity

A search algorithm tries to infer the position of new fit solutions based on the position of known fit solutions. In general, given two solutions *a, b*, if the fitness of solution

[4] It is based on results obtained in [] for a simple GA with one point crossover and no mutation. However, as shown in the previous section, these results are general (up to the exact shape of the variance).

[5] In the figure, the value 21 in the x axis corresponds to 1, i.e full degree of information.

a is higher than the fitness of solution *b* the algorithm will try to search in the neighbourhood of *b.*

A flat landscape corresponds to many such *a* and *b* on which the algorithm cannot follow the same heuristic. Since both have the same fitness, it has to make a *random* choice. The information landscape represents the algorithm perspective of a landscape. It shows explicitly what the algorithm can infer from the landscape and what it should guess.

If the algorithm does not use resampling, an information landscape which contains entries equal to 0.5 can be represented by an equivalent landscape on which every such entry is replaced by either 1 or 0 with the equal probability.

Indeed an algorithm under this scenario, given *a,b* with equal fitness, chooses randomly which solution will win in a tournament. Since we assume that it samples each solution only once. It makes a random choice once for each pair.

Note that the information landscape generated by these random assignments is not necessarily valid (see equation 2). Hence it can not be represented by an equivalent fitness landscape.

Information Landscape

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

Algorithm →

Translated Landscape: The Algorithmic Perspective

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

Figure 5. An example for a landscape (left) and a possible interpretation of this landscape by an algorithm. The algorithm "translates" each undecidable tournament to a decidable one by assigning either 1 or 0 in a random way.

In algorithmic information theory each string s = {0,1}* is associated with an algorithm complexity denoted *h(s)*. It is defined by the length of the shortest program that output *s* and halts.

Since each algorithm and each problem are represented in a computer essentially as binary strings, the complexity of strings which represent them is associated with the complexity of the problem (algorithm).

In principal, compressible problems (landscapes) are associated with easy problems, whereas uncompressible (random) problems are associated with hard ones.

The NIAH is usually given as an example which contradicts this intuition. The NIAH assigns 1 to exactly one point in the search space and 0 to all the others. For a search space of size N, the average description length of a needle is O(ln N) [1]. This function is therefore highly compressible, yet, very difficult to search on.

This seemingly contradiction can be resolved by considering the algorithm perspective of the NIAH landscape. As previously noted, the algorithm does not see a flat landscape. At each time step it is has to make a concrete decision, namely to decide which solution to sample next. It therefore has to "translate" the flat landscape to a landscape which contains concrete information. Figure 5 gives an example to a possible translation.

In order to avoid confusion we will denote the algorithm's interpretation of the landscape, *translated landscape.*

In order to find the true Kolmogorov complexity of a problem, the translated landscape (and not the regular one) should be considered. Note that for a landscape with a degree of information equals 1, the translated landscape coincides with the regular one.

Let us consider the compressibility of the NIAH problem. All the entries of the induced information landscape equal to 0.5. The translated landscape is generated by assigning for each entry either 1 or 0 with equal probability. The compressibility of the translated landscape which is by *definition random* is naturally, minimal, hence the Kolmogorov complexity, maximal.

The NIAH, however, is an extreme example. Generally, moving from degree 1 (full information) to degree 0 (NIAH) we should obtain many intermediate values. In order to illustrate how the average Kolmogorov complexity changes as the degree of information decreases, we conducted a simple experiment.

Considering a landscape of size 50 (i.e. the size of the associate information landscape equals 1200). We generated 1000 random different landscapes for each level (degree) of information. The translated landscape was obtained by assigning randomly either 1 or 0 to each entry equals to 0.5.

We then used the lamepl-ziv compression algorithm to compress the two landscapes. Figure 6 presents, for each degree of information, the average compressibility of the two original landscapes versus the translated landscape. Note that due to the irregular measure of compressibility, the figure gives qualitative information rather than quantitative one.

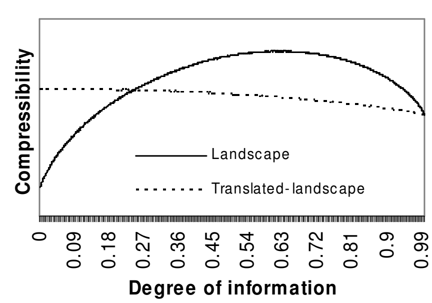


Figure 6. The average compressibility of the two aspects of a landscape over all possible degrees of information.

Figure 6 illustrates how the common understanding of the Kolmogorov complexity of a problem differs from the suggested one. A small degree of information yields for the regular landscape maximum compressibility. As the degree of information increases the average compressibility decreases. The translated landscape, however, exhibits the opposite behavior.

Note that for a degree of information equals to one the compressibility of the two landscapes coincides[6].

5.1 The Kolmogorov complexity of a block

In this section we show that the average Kolmogorov complexity of a block increases as the degree of information associated to a block decreases. The intuition behind the proof is simple, however, the proof itself is beyond the scope of this paper. We begin with a formal definition of the average Kolmogorov complexity. We then give an informal explanation for the reason it should decrease for higher degree of information.

Let $f: X \rightarrow Y$ be a fitness function and $|X| = |Y| = N$. Define F the set of permutation obtained on function *f.* Also, let $d^{0.5}(F) \rightarrow [0,1]$ denote the degree of information for the set F (as defined in section 4.1).

Let V be the set of the associated information landscapes, for each i let V_i be the information landscape associated with F_i.

Define $t: V_i \rightarrow V_i^*, d^{0.5}(V_i^*) = 1$ denote the *translated* landscape. It receives an information landscape and output another information landscape for which the degree of information equals one (i.e. it assigns randomly either 1 or 0 to each entry equals to 0.5. see figure 5). *t* is stochastic. Since we are interested in the algorithmic complexity of the generated landscape, let $V^* = (t(V_i))$ denote the landscape with the *expected* algorithmic complexity. For each V_i^* let $h(V_i^*)$ denote the algorithm-complexity of V_i^*.

Define $h(V^*) = \frac{1}{|V^*|} \sum h(V_i^*)$ the average algorithmic complexity of V^*.

Let F_1, F_2 be closed under permutation:

Hypothesis:

$$d^{0.5}(F_1) > d^{0.5}(F_2) \Rightarrow h(V_1^*) < h(V_2^*)$$

Consider the space of all possible information landscapes versus the space of all possible fitness landscapes. Let $f: X \rightarrow Y$ where $|X| = |Y| = N$, and let V be an information landscape. It is clear that for any reasonable N $|V_i| > |f_i|$:

$$\begin{aligned} |V_i| > |f_i| &\Leftrightarrow 3^{(N^2 - N)/2} > N^N \\ &\Rightarrow (N^2 - N)/2 > N \log_3(N) \\ &\Rightarrow N - 1 > 2\log_3(N) \\ &\Rightarrow N > 3 \end{aligned}$$

Since the Kolmogorov complexity, as argued in the previous section, should be associated with the *translated landscape*, on the extreme case (i.e. NIAH), the number of possible landscapes (from which the actual one is chosen) is the whole possible set. Since almost all strings are random, we should expect the random generated string to be random as well. The larger the number of possible landscape is, the smaller the variance from the expected (i.e. random, maximum Kolmogorov complexity) landscape will be.

On the other extreme, consider the case of full information. In that case, the translated landscape equals the landscape itself.

Hence, averaging over all possible landscapes in the set – for a set with small degree of information, *all* of the landscape will be random. On the other hand, for the set with full degree of information, some landscapes will be compressible (small fraction), and some random (big fraction). This implies that the average complexity of sets with higher degree of information is expected to be lower.

6 Discussion

In [13] Koppen, Wolpert and Macready objected that researchers tend to consider the NFL theorems from almost a nihilistic point of view. They pointed that the NFLT should be considered as a research topic by itself.

This paper contributes to the ongoing research in this area. It is based on a novel framework (information landscape) which was introduced in [2],[11],[12]. As a starting point we used results obtained in [1],[9],[10] indicating that the permutation closure of a function is both necessarily and sufficient for the NFLT to hold.

Using the information landscape framework we were able to connect a permutation closure of a function with the amount of information its members have, the expected performance of the algorithm and the average Kolmogorov complexity of its members.

Generally, the information of a function can be calculated and associated with what we denote as the degree of information. The results of this paper show that a permutation closure associated with a low degree of information (1) is expected to have high Kolmogorov complexity and (2) the performance of an algorithm over its members is expected to be close to random search.

A second contribution of this paper was to identify a misconception regarding the way to calculate the Kolmogorov complexity of a fitness landscapes. This

[6] Since the y axis reflects just an indicator for the compressibility of the problems, we omitted the actual values. Furthermore, note that the translated landscape contains only two symbols (1 and 0) whereas the regular landscape may contains 3 (1, 0.5 and 0). This explains the increase in compressibility of the landscape as the degree of information approaches 1.

confusion led to the belief that the NIAH is an example to a difficult problem which has a very low Kolmogorov complexity. Arguing, that the algorithm does not interpret the NIAH as a flat landscape (section 5) we showed that its actual Kolmogorov complexity is high.

According to this interpretation the Kolmogorov complexity of a problem can be associated with the extent to which an algorithm can perform either better or worse than random search (see [14] for further details).

The way the information landscape framework is defined might give rise to some objections about its applicability for the general case. One can object that unlike the different NFLT, the information landscape framework and hence the NFL results deriving from it are based on *approximation*. This is not the case.

The main results of this paper are independent of the approximation of the performance done via the performance landscape: both, the degree of information associated with blocks and the new notion of Kolmogorov complexity depends on the information landscape alone. Moreover, even if, for the sake of argument, the approximation of the performance is not accurate, still the interpretation of the NFL results is correctly interpreted *qualitvitely*: the NFLT, in general, predict the same symmetry w.r.t to random search as illustrated in figure 4. Since that in this paper, we did not use the information landscape framework to actually *predict* the performance of an algorithm, the hypothetical accuracy of the prediction is insignificant.

Additional possible objections may relate to the assumptions done by the information landscape. In particular, it considers only one global optimum. This can be generalized easily. Instead of considering a fixed X_{trgt}, consider a vector of targets. The expected performance will be linearly dependant on this vector. The symmetry w.r.t to random search will not be affected.

The information landscape makes additional assumption, rather than considering absolute fitness values, it considers only relative fitness. However, this is the way modern search algorithms interpret *in practice* the landscape (e.g. GA with tournament selection, PSO, local search heuristics). In any case, it turns out that determining performance order generally requires much less computational effort than determining performance value [15]. Moreover, this implies a reduction in the number of possible problem instances which suggests a possible increase in performance [15]. It is suggested therefore, even for search algorithms that use the absolute fitness value, to use the relative order instead.

7 Conclusion

It was proven in [1] that the permutation closure of a single function is the finest level of granularity at which a NFL result can hold. However, properties of the different closures were not identified.

In this paper, we proved that each closure can be associated with a particular degree of information, which affects the possible performance of an algorithm, over its members.

In addition, we identified a misconception regarding the calculation of the Kolmogorov complexity for fitness landscapes. We showed that the actual Kolmogorov complexity of the NIAH, for example, is very high.

Finally, we associated each permutation closure with a particular Kolmogorov complexity. This is illustrated in Figure 4.

Bibliography

[1] C. Schumacher, M. D. Vose, and L. D. Whitley. The No Free Lunch and description length. In L. Spector, E. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors, Genetic and Evolutionary Computation Conference (GECCO 2001), pages 565-570, San Francisco, CA, USA, 2001. Morgan Kaufmann

[2] ****

[3] David Wolpert, William G. Macready: No free lunch theorems for optimization. IEEE Trans. Evolutionary Computation 1(1): 67-82 (1997)

[4] Joseph C. Culberson: On the Futility of Blind Search: An Algorithmic View of ``No Free Lunch''. Evolutionary Computation 6(2): 109-127 (1998)

[5] Stefan Droste, Thomas Jansen, Ingo Wegener: Optimization with randomized search heuristics - the (A)NFL theorem, realistic scenarios, and difficult functions. Theor. Comput. Sci. 287(1): 131-144 (2002)

[6] Thomas M. English: On the Structure of Sequential Search: Beyond "No Free Lunch". EvoCOP 2004: 95-103

[7] English, T. "No More Lunch: Analysis of Sequential Search," to appear

[8] P.D. Grunwald, P.M.B. Vitanyi, Shannon Information and Kolmogorov complexity, IEEE Trans. Information Theory, Submitted

[9] T. M. English. Optimization is easy and learning is hard in the typical function. In A. Zalzala, C. Fonseca, J.-H. Kim, and A. Smith, editors, Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000), pages 924-931, LA Jolla, CA, USA, 2000. IEEE Press

[10] Thomas M. English: Practical Implications of New Results in Conservation of Optimizer Performance. PPSN 2000: 69-78

[11] ****

[12] ****

[13] M. Koppen, D. H. Wolpert, and W. G. Macready. Remarks on a recent paper on the "No Free Lunch" theorems. IEEE Transactions on Evolutionary Computation, 5(3):295-296, 1995

[14] ****

[15] Y. C. Ho and D. L. Pepyne, "Simple explanation of the No Free Lunch Theorem and its implications," *J. Optim. Theory Applicat.*, vol. 115, no.3, Dec. 2002