Classes of Problems in the Black Box Scenario

Yossi Borenstein
University of Essex
Colchester, U.K
yboren@essex.ac.uk

Riccardo Poli
University of Essex
Colchester, U.K
rpoli@essex.ac.uk

ABSTRACT

The no-free-lunch theorems (NFLTs) do not consider explicitly the *structure* of problems. In [1] we gave a formal definition of structure. We showed that many metaheuristics have identical performance on problems which belong to the same structural class. In this paper, we define a notion of a distance between fitness functions. We argue that an algorithm cannot be efficient on a class of problems if the distance between the fitness function associated with instances of that class is too big. In [2] we corroborate our ideas using several problems.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY

General Terms

Algorithms,Theory

Keywords

No Free Lunch, Genetic algorithms, Heuristics, Theory

1. INTRODUCTION

With their No-free-lunch theorem (NFLT) Wolpert and Macready [3] put an end to the hope of developing a general-purpose optimization algorithm. Following these results the NFLTs are usually interpreted in the following way: even though it is not possible to design a general algorithm it is possible to design an efficient algorithm for a specific subset of problems, real-world problems being such set.

In other words, considering the subset $F = \{f_1, f_2, ..., f_l\}$, the idea is that it should be possible to choose a search algorithm a such that the performance of a over F will be good. Naturally, this is not possible for an arbitrary set. However, it is assumed that instances of real-world problems which belong to the same class (e.g., SAT or MAXSAT) are structurally "related" and that, therefore, there exists a search algorithm which performs well on all of them. This informal definition of "structure" and "relation" makes it impossible to validate (or disprove) this assumption. In this paper we make a first attempt to solve this problem.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

The paper is organized as follows. In section 2 we define, for each function, the class of its structurally identical functions, which we call the *isometry group* of a function. We argue that the many metaheuristics are expected to have the same performance on every function which belong to the same isometry group (see [1] for more details). In section 3 we define a notion of distance between functions. We discuss, in section 4, functions with various distances and assess the likelihood that a search algorithm exists that can efficiently optimise them.

2. STRUCTURE

A problem in the black box scenario is often represented by the triple (X, d, f) where X is the search space, $d : X \to \mathbb{R}$ is a distance function and $f : X \to \mathbb{R}$ is the objective function. In this section we propose to define structure as an isometric isomorphism relation between functions.

Definition 1. Let (X, d) be a metric space. The transformation $\sigma^d : X \to X$ is a distance preserving transformation (or an isometry) if:

$$\forall x, y \in X \;\; d(x, y) = d(\sigma^d(x), \sigma^d(y))$$

The *Isometry group* is the set of all isometries under function composition.

Based on a distance preserving permutation of a metric space, we can define a distance preserving permutation of a function. Such a permutation preserves all the structural properties of the fitness landscape.

Definition 2. Let (X, d) be a metric space, σ^d a distance preserving permutation and Σ the isometry group. For any $f : X \to Y$:

1. The permutation $\sigma^d f$ of f is the function $\sigma^d f : X \to Y$ defined by $\sigma^d f(x) = f(\sigma^{d^{-1}}(x))$.

2. The set $F = \{g | \exists \sigma^d \in \Sigma, g = \sigma^d f\}$ is the *structural class* (or orbit) of f.

The distance preserving permutation of a function preserves the relations between the fitness values and the neighborhood structure. The notion of structural class of a function clarifies the many symmetries which exist in the space of all possible problems.

3. DISTANCE BETWEEN FUNCTIONS

The main objective of this paper is to estimate, given a group of instances (or fitness functions defined over the same metric space), the likelihood that a search algorithm which solves all of them efficiently, exists. Intuitively, if the functions are completely different (i.e., they do not share any structural property), it is not likely for such an algorithm to exist. In this section we formally define a notion of distance or similarity between functions.

Before we define the distance formally we represent a fitness function in terms of the relative fitness values: each fitness function, $f : X \to Y$, can be represented by a $|X| \times |X|$ matrix M with entries $m_{i,j} = t(x_i, x_j)$, where $t : X \times X \to \{0, 1\}$ is given by

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

For every pair (x_i, x_j) of elements in X, $m_{i,j}$ indicates whether they have the same fitness and if not which is preferable. This matrix was denoted in [4] as an information landscape. Since, in this paper we consider the matrix in a different context we will simply refer to it as a *relative fitness matrix* or *relative fitness function*.

Definition 3. Let f', f'' be two fitness functions and M', M'' be the two corresponding relative fitness matrices. Let $s(X) = (|X|^2 - |X|)/2$ be the number of distinct elements in a matrix M. The distance between f' and f'' is defined as:

$$d_f(f', f'') = d_f(M', M'') = \frac{1}{s(X)} \sum_{i>j} |m'_{i,j} - m''_{i,j}|. \quad (2)$$

That is, the distance between two functions is defined as the absolute difference between the two matrices, normalised to stay in the interval $[0, 1]$. This distance was used in [4] as a predictive measure of problem difficulty. It was shown, using several case-studies, that the distance of a particular problem from an optimal (easy) landscape correlates well with the efficiency of a simple GA on this problem.

However, this distance does not correspond to the structure of the search space, as defined in the previous section. The following distance measure resolves this problem:

Definition 4. Let f', f'' be two fitness functions. Let F'' be the structural class of f''. The structural distance, d^s_f, between f', f'' is defined as:

$$d^s_f(f', f'') = \min_{g \in F''} d_f(f', g) \quad (3)$$

Equation 3 measures the distance of aligned functions. Keeping one of the functions (f') fixed, it picks the function (g) from the structural class (F'') of (f'') which minimises the distance as defined by equation 2.

4. A CLASS OF TWO PROBLEMS

In the previous section we defined a notion of distance between functions, we argued that a search algorithm is able to search efficiently a group of problems only if the structural distance (equation 3) between them is small. In [2] we exemplify this notion using *onemax*, needle-in-a-haystack, longpath (isotonic) and deceptive functions. We show, *using this distance*, that while it is likely for an efficient algorithm to solve both deceptive and onemax problems, it is not possible to solve efficiently both onemax and NIAH problems. We argue that depending on the specific instances of the longpath problem such an algorithm may or may not exists.

Moving from artificial problems to more realistic ones, we consider in [2] the SAT and MAXSAT problems. We argue that it is not possible to consider, from *the black box perspective* the structure of the SAT problem. All instances of SAT are variants of the NIAH (with possible more than one needle). The search strategy of randomized search heuristics depends on the ability to *select* solutions (from the ones already sampled) around which to focus the search – if no such solutions exists, the algorithm behave like a random search. This is not the case for MAXSAT, the optimisation variant of SAT. However, we gave example of two instances which belong to MAXSAT and have a big structural distance.

5. CONCLUSIONS

Understanding the connection between real world classes of problems to the black box scenario is perhaps one of the most important objectives for a theoretical framework for metaheuristics. In this paper we made some initial steps in this direction.

The starting point of our results was our formal definition of *structure* as an isometric isomorphism relation between functions. We then proceeded by defining a notion of a structural distance – that is a measure of distance between functions which respects structural similarities and differences. We argued that the same algorithm is not likely to solve efficiently instances of problems unless they are structurally related, i.e., their mutual structural distances are small.

Section 4 summarizes briefly the results obtained in [2]. In particular, we showed that instances of MAXSAT may not necessarily belong to the same *black box class* of problems. We believe that these surprising preliminary results call for further investigation. The distribution of problem instances, for example, might play a major role in determining whether or not an algorithm capable of solving a problem efficiently may exist. E.g., if most of the instances are structurally related while a minority are not, then the situation might be better than what we can infer at the moment for MAXSAT and SAT.

6. ACKNOWLEDGMENT

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

7. REFERENCES

[1] Yossi Borenstein and Riccardo Poli. Structure and metaheuristics. GECCO 2006.

[2] Yossi Borenstein and Riccardo Poli. Classes of problems in the black box scenario. Technical report, Department of Computer Science, University of Essex, 2006.

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

[4] Yossi Borenstein and Riccardo Poli. Information landscapes and problem hardness. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1425–1431, New York, NY, USA, 2005. ACM Press.