Theoretical Computer Science 600 (2015) 98–106

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

No-Free-Lunch theorems in the continuum

Aureli Alabert [a,*], Alessandro Berti [b], Ricard Caballero [a], Marco Ferrante [b]

[a] Departament de Matemàtiques, Universitat Autònoma de Barcelona, 08913-Bellaterra, Catalonia, Spain
[b] Dipartimento di Matematica, Università di Padova, 35121 Padova, Italy

ARTICLE INFO

Article history:
Received 26 November 2014
Received in revised form 27 June 2015
Accepted 13 July 2015
Available online 17 July 2015
Communicated by P. Widmayer

Keywords:
No-Free-Lunch
Stochastic processes
Black-box optimisation

ABSTRACT

No-Free-Lunch Theorems state, roughly speaking, that the performance of all search algorithms is the same when averaged over all possible objective functions. This fact was precisely formulated for the first time in a now famous paper by Wolpert and Macready, and then subsequently refined and extended by several authors, usually in the context of a set of functions with finite domain and codomain. Recently, Auger and Teytaud have studied the situation for continuum domains. In this paper we provide another approach, which is simpler, requires less assumptions, relates the discrete and continuum cases, and we believe that clarifies the role of the cardinality and structure of the domain.

1. Introduction

In [1], Wolpert and Macready formulated rigorously a principle which was already intuitively known to the operations research practitioners: All search or optimisation algorithms perform equally well when their performance is averaged against all possible objective functions. This principle has been known since then as the *No-Free-Lunch Theorem* (NFL for short).

The precise formulation of the Wolpert–Macready NFL Theorem will be stated in Section 2 (Theorem 2.1), but the basic assumptions are that we are dealing with the set of all functions $f: \mathcal{X} \to \mathcal{Y}$ between two finite sets $\mathcal{X}$ and $\mathcal{Y}$, and that the "averaging" is uniform over all these functions. The measure of performance can be any function of the images $f(x_1), \dots, f(x_m)$ of the points $x_1, \dots, x_m$ sampled by the algorithm.

In [2], Schumacher, Vose and Whitley extended the result to some subsets of all functions (those called "closed under permutation"), whereas Igel and Toussaint [3] stated it for some non-uniform measures. The language of probability theory allows to formulate these statements in a unified and easier way and it is in our opinion very convenient to switch from the finite setting to the continuum. In [4] and [5], Auger and Teytaud considered for the first time this case, and their result is essentially negative: No NFL theorems exist in the continuum.

Our goal in this paper is to improve and clarify the results of Auger and Teytaud, particularly Theorem 4.1, [5]. First of all, we show situations where NFL theorems do exist. This apparent paradox is resolved by noticing that the hypotheses imposed in [5] invalidate our examples. In fact, the authors seem to specifically look for conditions under which no NFL theorem can hold true. The theorem is indeed correct, although there is a gap in the proof, as explained in Section 3. We must also point out, however, that their paper contains much more material of interest on this and other matters.

* Corresponding author.
E-mail addresses: Aureli.Alabert@uab.cat (A. Alabert), berti@math.unipd.it (A. Berti), rcaballero@mat.uab.cat (R. Caballero), ferrante@math.unipd.it (M. Ferrante).

http://dx.doi.org/10.1016/j.tcs.2015.07.029
0304-3975/

The point of view adopted here is different: We establish a simple and natural definition of the NFL property and look for the necessary conditions implied by this definition. In this sense, our main result is Theorem 3.9. The conclusion we reach is that there are no No-Free-Lunch theorems for the set of functions having second-order moments, except for a few cases (illustrated in Examples 3.4).

The relevance of this theoretical discussion for the field of global *black-box* optimisation comes from the so-called *probabilistic models* (see, for instance [6, chap. 4]): In many practical optimisation problems there is little information about the objective function, with no access to derivatives or to any explicit formula; we are only allowed to ask the function for its value at a point of our choice and, after observing the value returned, we may decide on the next point to sample the function; and so on. Moreover, function evaluations can be expensive, and we are constrained to make only a small number of them. In these cases, it may be useful to think that the function has been drawn at random from some set of functions, according to some probability law (perhaps with some unknown parameters) that one specifies using prior information. Technically, we are then in the presence of a *stochastic process*, from which our function is a particular path. Different algorithms will choose different points for the successive evaluations, and some may perform better than others by exploiting better the model, *unless* there is a No-Free-Lunch theorem for that model. If this is the case, all algorithms perform the same in average and, in particular, pure blind search is as good as any other proposal. In the present paper we will see that the presence of the No-Free-Lunch property reduces to a few probabilistic models, which are not really important in practice.

A different approach to No-Free-Lunch can be found in Rowe, Vose and Wright [7], where the concept is discussed in purely set-theoretic terms as a symmetry property of a set of functions rather than in relation with any particular application. The authors conclude that an NFL property holds whenever the set of functions is closed under permutations (as in [2] for the finite case), no matter the cardinality of domain and codomain. Our approach here is totally different, to the point that the respective definitions of No-Free-Lunch are not equivalent in the non-finite context.

The paper is organised as follows: In Section 2 we state the definitions and preliminaries both from algorithmics and from probability theory that are strictly needed in the rest of the paper. In Section 3 we state the main results: We show that No-Free-Lunch cases do exist in the continuum; we impose then a hypothesis of measurability of the stochastic process involved, and we see that NFL can only appear if we are dealing with functions whose randomness is concentrated in a set of null Lebesgue measure (Theorem 3.5), or the model consists of a trivial constant process (Theorem 3.9). In Section 4 we justify the investigation of the existence (or not) of NFL properties in the continuum and propose some open questions.

2. Preliminaries

We follow approximately the notations of [3] and [5], with some convenient modifications.

2.1. Algorithmic concepts

Let $\mathcal{X}$ and $\mathcal{Y}$ be any two sets. The set of all functions $f:\mathcal{X}\to\mathcal{Y}$ can be identified with the Cartesian product $\mathcal{Y}^{\mathcal{X}}$. Denote

$$\mathcal{E}_0 := \{\emptyset\},\ \mathcal{E}_1 := \mathcal{X}\times\mathcal{Y},\ \dots,\ \mathcal{E}_m = \mathcal{X}^m\times\mathcal{Y}^m$$

and $\mathcal{E} := \bigcup_{m\geq 0}\mathcal{E}_m$.

A *(random) algorithm* A is a mapping $A:\mathcal{Y}^{\mathcal{X}}\times\mathcal{E}\times\Theta\to\mathcal{E}$, where $(\Theta,\mathcal{G},Q)$ is a probability space and, if $e = ((x_1,y_1),\dots,(x_m,y_m))\in\mathcal{E}_m$, then $A(f,e,\theta)\in\mathcal{E}_{m+1}$ and

$$A(f,e,\theta) = ((x_1,y_1),\dots,(x_m,y_m),(x_{m+1},y_{m+1}))$$

with $y_{m+1} = f(x_{m+1})$. Therefore, we can think of

$$A(f,e) := ((X_1,Y_1),\dots,(X_m,Y_m))$$

as a random vector $\Theta\to\mathcal{E}$. This definition formalises the fact that the algorithm chooses the next point based on the previous points and an (optional) random mechanism represented by the probability space $(\Theta,\mathcal{G},Q)$. One may assume that f is never evaluated more than once at the same point.

A *measure of performance* of the algorithm is any function C of the values obtained by evaluating f during the algorithm. Formally,

$$C: \bigcup_{m\geq 1}\mathcal{Y}^m\to\mathbb{R}\,.$$

A typical measure of performance for optimisation problems is the function $C(y_1,\dots,y_m) = \min\{y_1,\dots,y_m\}$, the best observed value after m evaluations. (Notice that the measures of performance we are talking about are not related to algorithmic complexity, e.g. to the number of evaluations needed to reach the end of a procedure.)