Reinterpreting No Free Lunch

Jon E. Rowe J.E.Rowe@cs.bham.ac.uk
School of Computer Science, The University of Birmingham, Edgbaston, Birmingham, B15 2TT, United Kingdom

M. D. Vose vose@eecs.utk.edu
Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, 37996-3450, USA

Alden H. Wright aldenwright@fastmail.us
Department of Computer Science, The University of Montana-Missoula Missoula, MT 59812, USA

Abstract

Since it's inception, the "No Free Lunch theorem" has concerned the application of symmetry results rather than the symmetries themselves. In our view, the conflation of result and application obscures the simplicity, generality, and power of the symmetries involved. This paper separates result from application, focusing on and clarifying the nature of underlying symmetries. The result is a general set-theoretic version of NFL which speaks to symmetries when arbitrary domains and co-domains are involved. Although our framework is deterministic, we note situations where our deterministic set-theoretic results speak nevertheless to stochastic algorithms.

Keywords

Black Box Search, No Free Lunch

1 Introduction

We reinterpret the "No Free Lunch" theorem (NFL) to be a statement which is most naturally expressed in set-theoretic terms and which concerns symmetries inherent in Black Box search without regard to any purpose to which those symmetries may be put. This interpretation conflicts with the historical fact that NFL was first expressed using the language of probability by Wolpert and Macready (1995) and was very much concerned with exploiting symmetry. Although probability may provide a a means by which underlying symmetries shed light on inherent limitations of Black Box Search, probabilistic language nevertheless complicates both the statement and proof of NFL results – as is clear upon comparing Wolpert and Macready (1997), Köppen (2000), Köppen et al. (2001) with the set-theoretic treatment of Schumacher (2000).

If the goal is to understand underlying symmetries – which has not historically been the case – then continued use of probabilistic language drags probability into a situation where it does not belong. Rather than clarifying the nature of symmetries, probability provides a straw-man, as Droste et al. (1999) point out: "...taking randomly a function...we have with large probability not enough time to evaluate...at only one sample point". Probability leads one to conclude, as Auger and Teytaud (2007) do, that NFL fails for continuous domains. Whereas that is a valid conclusion regarding the classical probabilistic NFL, it can be argued that the conclusion speaks more to a failure of the probabilistic framework than to the absence of NFL symmetries in the general case.

Because our goal is limited to NFL symmetries, we do not have much to say about classical probabilistic NFL results; they are directed at applications, which is orthogonal to our purpose of clarifying the nature of symmetries involved. We do demonstrate in a concluding section, however, that our abstract results have sufficient power to imply the classical probabilistic NFL theorem (involving finite domains and co-domains).

Because our goal is also ambitious – to make sense of NFL for arbitrary domains and co-domains – the treatment is necessarily technical. In particular, it uses concepts like cardinality, ordinality, and transfinite induction. We assume the reader knows about such things, but mention Hewitt and Stromberg (1965), and Devlin (1994) as references.

Given that we consider algorithms which search uncountable domains, it might be natural to wonder: what is the algorithmic content of such algorithms? For the purposes of this paper, suffice it to say that an algorithm is a *mathematical abstraction*; the value $\mathcal{A}_f^\alpha(\emptyset)$ of applying search algorithm $\mathcal{A}$ to function f for α steps beginning from the empty sequence $\emptyset$ of points sampled is well defined (with respect to Zermelo-Fraenkel set theory) for every ordinal α. Moreover, the knowable properties of the result are those properties which can be proved concerning the result, and proofs do not necessarily require the result to be Turing computable. It should be appreciated that a general NFL theorem which holds even for search algorithms that need not be Turing computable will necessarily specialize to a NFL theorem which holds for search algorithms that *are* Turing computable.

The next section presents definitions and notation and defines Deterministic Non-Repeating Black Box Search Algorithms in set-theoretic terms. We mention here that a search algorithm's definition involves a search operator which is described as being defined at points in its domain that it will never be called upon to evaluate (the search operator could be arbitrarily defined at such points). While that may seem odd, it is of no consequence (and it is not acknowledged where it occurs). Search operators are so defined as a simple matter of convenience (just as, for instance, it may be convenient to define a crossover operator to act on any pair of chromosomes, even though some particular pair of chromosomes might not actually occur during an optimization run for a given initial population and random number seed). Although general stochasticity is not investigated (the issues involved are beyond the scope of this paper), we do mention in the concluding section situations where our deterministic results speak to non-deterministic algorithms.

Section 3 presents preliminary results which generalize the approach to NFL taken by Schumacher (2000) from finite to arbitrary ordinals. It should be mentioned that Schumacher's account was inspired by and is an extension, not a revision, of results contained in Radcliffe and Surry (1995). The Uniqueness, Completeness, and Duality Theorems proved in section 3 are the cornerstones of NFL. Interestingly, it is only the Completeness Theorem that takes on a different character when the search space $\mathcal{X}$ is infinite. The NFL theorem must also take on a different character, since the smallest ordinal α such that $\mathcal{A}_f^\alpha(\emptyset)$ exhaustively explores the space can depend both on $\mathcal{A}$ and on f, which is not the case when $\mathcal{X}$ is finite.

Section 4 presents set-theoretic NFL in the form of three theorems concerning the behavior of Deterministic Non-Repeating Black Box Search Algorithms. The first is local in the sense that it focuses on two given search algorithms, and, roughly speaking, says that for any function f there exists a function f' such that the behavior of the first algorithm applied to f matches the behavior of the second algorithm applied to f'. The second theorem corresponds to what is most commonly thought of as NFL, and, roughly speaking, says that all algorithms perform equally well on a set F of functions if and only if F is closed (with respect to permutation). The third NFL theorem speaks to performance measures that evaluate the behavior of algorithms for some number of steps having cardinality less than the cardinality of $\mathcal{X}$. The reader is cautioned that the description of NFL given in this introduction is very rough indeed. As mentioned in the previous paragraph, NFL results have a different character when $\mathcal{X}$ is infinite, and that character is reflected in technical conditions which qualify the oversimplified description given in this paragraph.

As already mentioned, the final section briefly mentions applications and touches upon the issue of stochasticity. It also suggests potential directions for extending this work.

2 Basic Definitions and Notation

Let $D(r)$ denote the domain of r (for arbitrary relation r), and let $I(\alpha)$ denote the set of ordinal numbers less than α (for arbitrary ordinal α). A *sequence* S is a function whose domain is $I(\alpha)$. Let S^* denote the range of S, and let $\pi_\beta(S)$ denote the restriction of sequence S to domain $I(\beta)$. Let $|S|$ denote the cardinality of S (for arbitrary set S), and define the cardinality $\overline{\alpha}$ of ordinal α to be $|I(\alpha)|$. Let $\lfloor S \rfloor$ denote the smallest ordinal α such that $\overline{\alpha} = |S|$.

Let $f : \mathcal{X} \to \mathcal{Y}$ be a function between arbitrary sets, and let y_i denote $f(x_i)$. The domain $\mathcal{X}$ and co-domain $\mathcal{Y}$ are fixed for the following discussion, but f may vary.

Definition: A *trace T corresponding to f* is a sequence $\langle (x_0, y_0), \ldots \rangle$ of pairs from $\mathcal{X} \times \mathcal{Y}$ where the x components are unique (in particular, a trace is an injective function); T is a *trace* if it is a trace corresponding to some f. The following notation will be used,

$$
\begin{array}{lcll}
T^* & = & \{(x_0, y_0), \ldots\} & \text{set of components} \\
T_x & = & \langle x_0, \ldots \rangle & \text{sequence of } x \text{ components} \\
T_y & = & \langle y_0, \ldots \rangle & \text{sequence of } y \text{ components}
\end{array}
$$

In particular, $T^* \subset f$. A *performance sequence* is a sequence of values from $\mathcal{Y}$. The *performance sequence associated with trace T* is T_y.

Definition: Trace T corresponding to f is *total* if $T^* = f$. A *partial* trace is one which is not total. The set of all partial traces corresponding to function f is denoted by $\mathcal{T}(f)$, and $\mathcal{T}$ is defined by

$$
\mathcal{T} = \bigcup_f \mathcal{T}(f)
$$

Definition: A *search operator* is a function $g : \mathcal{T} \to \mathcal{X}$ which maps a partial trace T to some element not occurring in T_x.

Definition: A *deterministic non-repeating Black Box search algorithm $\mathcal{A}$* corresponds to a search operator g, and will be referred to simply as a *search algorithm*. Algorithm $\mathcal{A}$ applied to function f is denoted by $\mathcal{A}_f$, and maps traces to traces

$$
\mathcal{A}_f(T) = \begin{cases} T \,\|\, (g(T), f \circ g(T)) & \text{if } T \in \mathcal{T}(f) \\ T & \text{otherwise} \end{cases}
$$

where $\|$ is the concatenation operator ($D(T) = I(\alpha) \Longrightarrow T\|z = T \cup \{(\alpha, z)\}$). For any ordinal n,

$$\mathcal{A}_f^n(\emptyset) \;=\; \begin{cases} \emptyset & \text{if } n = 0 \\ \bigcup_{m<n} \mathcal{A}_f^m(\emptyset) & \text{if } n \text{ is a limit ordinal} \\ \mathcal{A}_f(\bigcup_{m<n} \mathcal{A}_f^m(\emptyset)) & \text{otherwise} \end{cases}$$

The trace *generated by* search algorithm $\mathcal{A}$ applied to function f is

$$\mathcal{A}(f) \;=\; \bigcup_{0<n} \mathcal{A}_f^n(\emptyset)$$

where $\emptyset$ is the empty trace. Search algorithms $\mathcal{A}$ and $\mathcal{A}'$ are considered identical if and only if they generate the same trace for all f. A *performance table* is a matrix whose rows are labeled by the search algorithms and whose columns are labeled by the functions; the element in row $\mathcal{A}$ and column f is $\mathcal{A}(f)_y$.

3 Preliminary Results

We begin with a theorem providing technical results which, among other things, give legitimacy to definitions in the previous section.

Theorem (Recursion)**.** *For any function $f \in \mathcal{Y}^{\mathcal{X}}$, search algorithm $\mathcal{A}$, and ordinal n,*

$$\begin{aligned} \bigcup_{m\le n} \mathcal{A}_f^m(\emptyset) &\;=\; \mathcal{A}_f^n(\emptyset) \quad \textit{is a trace corresponding to } f \\ \mathcal{A}_f^n(\emptyset) &\;=\; \mathcal{A}_f(\mathcal{A}_f^{n-1}(\emptyset)) \quad \textit{if } n \textit{ is not a limit ordinal} \\ n \le \lfloor \mathcal{X} \rfloor &\;\Longrightarrow\; D(\mathcal{A}_f^n(\emptyset)) = I(n) \\ \mathcal{A}(f)^* &\;=\; f \end{aligned}$$

Proof: First note that $T \subset \mathcal{A}_f(T)$ for every trace T, since if $D(T) = I(\alpha)$, then

$$T \,\|\, (g(T), f \circ g(T)) \;=\; T \cup \{(\alpha, (g(T), f \circ g(T)))\}$$

The first assertion of the theorem is proved by transfinite induction. Note that it is trivially true when $n = 0$. Suppose it is true for all $n < \alpha$.
Case 1: α is a limit ordinal. Then

$$\bigcup_{m\le\alpha} \mathcal{A}_f^m(\emptyset) \;=\; \mathcal{A}_f^\alpha(\emptyset) \cup \bigcup_{m<\alpha} \mathcal{A}_f^m(\emptyset) \;=\; \bigcup_{m<\alpha} \mathcal{A}_f^m(\emptyset) \;=\; \mathcal{A}_f^\alpha(\emptyset)$$

Therefore,

$$\begin{aligned} \mathcal{A}_f^\alpha(\emptyset)^* &\;=\; \bigcup_{m<\alpha} \mathcal{A}_f^m(\emptyset)^* \subset f \\ D(\mathcal{A}_f^\alpha(\emptyset)) &\;=\; \bigcup_{m<\alpha} D(\mathcal{A}_f^m(\emptyset)) \end{aligned}$$

The right hand side above is $I(\beta)$, where β is the smallest ordinal not contained in the right hand side above. To complete case 1 we show, via contradiction, that $\mathcal{A}_f^\alpha(\emptyset)$ is an injective function. If $\{(k, (x, y)), (k, (x', y'))\} \subset \mathcal{A}_f^\alpha(\emptyset)$, then, for some $n < \alpha$,

$\{(k,(x,y)),(k,(x',y'))\} \subset \mathcal{A}_f^n(\emptyset)$, which contradicts that $\mathcal{A}_f^n(\emptyset)$ is a trace. Likewise, if $\mathcal{A}_f^\alpha(\emptyset)(k) = \mathcal{A}_f^\alpha(\emptyset)(k') = (x,y)$ for $k \neq k'$, then $\mathcal{A}_f^n(\emptyset)(k) = \mathcal{A}_f^n(\emptyset)(k') = (x,y)$ for some $n < \alpha$, which contradicts that $\mathcal{A}_f^n(\emptyset)$ is a trace.
Case 2: $\alpha > 0$ is not a limit ordinal. Then, since $T \subset \mathcal{A}_f(T)$ for every trace T,

$$\bigcup_{m \le \alpha} \mathcal{A}_f^m(\emptyset) \;=\; \mathcal{A}_f^\alpha(\emptyset) \cup \bigcup_{m<\alpha} \mathcal{A}_f^m(\emptyset) \;=\; \mathcal{A}_f(T) \cup T \;=\; \mathcal{A}_f(T) \;=\; \mathcal{A}_f^\alpha(\emptyset)$$

where

$$T \;=\; \bigcup_{m \le \alpha-1} \mathcal{A}_f^m(\emptyset) \;=\; \mathcal{A}_f^{\alpha-1}(\emptyset) \;\text{ is a trace corresponding to } f$$

It follows, by the definition of $\mathcal{A}_f$, that $\mathcal{A}_f^\alpha(\emptyset)$ is a trace corresponding to f.
The second assertion of the theorem follows from Case 2 above.
The third assertion of the theorem is proved by transfinite induction. Note that it is trivially true when $n = 0$. Suppose it is true for all $n < \alpha \le \lfloor \mathcal{X} \rfloor$. If α is a limit ordinal, $D(\mathcal{A}_f^\alpha(\emptyset)) = I(\alpha)$ follows by definition (and the inductive hypothesis). If $\alpha > 0$ is not a limit ordinal, then (via the second assertion of the theorem and the definition of $\mathcal{A}_f$),

$$D(\mathcal{A}_f^\alpha(\emptyset)) \;=\; D(\mathcal{A}_f(\mathcal{A}_f^{\alpha-1}(\emptyset))) \;=\; I(\alpha-1) \cup \{\alpha-1\} \;=\; I(\alpha)$$

provided $\mathcal{A}_f^{\alpha-1}(\emptyset)^* \neq f$. That is true because their cardinalities differ; since $\alpha - 1 < \lfloor \mathcal{X} \rfloor$,

$$|\,\mathcal{A}_f^{\alpha-1}(\emptyset)^*\,| \le \;\overline{\alpha-1}\; < |\,\mathcal{X}\,| = |\,f\,|$$

The fourth assertion follows from the existence of β for which $\mathcal{A}_f^\beta(\emptyset) = \mathcal{A}_f^{\beta+1}(\emptyset)$; that yields, via the second assertion of the theorem,

$$\mathcal{A}_f^\beta(\emptyset) \;=\; \mathcal{A}_f^{\beta+1}(\emptyset) \;=\; \mathcal{A}_f(\mathcal{A}_f^\beta(\emptyset))$$

In view of the definition of $\mathcal{A}_f$, the trace $\mathcal{A}_f^\beta(\emptyset)$ cannot be partial, and therefore

$$f \;=\; \mathcal{A}_f^\beta(\emptyset)^* \;\subset\; \mathcal{A}(f)^* \;\subset\; f$$

Ordinal β can be obtained as follows. Since a trace is an injective function from some $I(\alpha)$ to f,

$$\overline{\alpha} \;\le |\,f\,| \le |\,\mathcal{X}\,|$$

If γ is any ordinal having cardinality greater than $|\,\mathcal{X}\,|$, then the domain of every trace is contained in $I(\gamma)$, and the (cardinal) number of traces is bounded by

$$2^{\overline{\gamma}\,|\mathcal{X}|}$$

Hence the function $\alpha \mapsto \mathcal{A}_f^\alpha(\emptyset)$ cannot be injective over the domain $I(\alpha')$ where α' has cardinality greater than that displayed above. Therefore, let $\beta < \beta'$ be such that

$$\mathcal{A}_f^\beta(\emptyset) = \mathcal{A}_f^{\beta'}(\emptyset)$$

Using the first assertion of the theorem,

$$\mathcal{A}_f^\beta(\emptyset) \;\subset\; \mathcal{A}_f^{\beta+1}(\emptyset) \;\subset\; \mathcal{A}_f^{\beta'}(\emptyset)$$

□

Lemma 1. *If f, $f' \in \mathcal{Y}^{\mathcal{X}}$ are functions and T is a trace such that $T^* \subset f \cap f'$, then*

$$T^* = f \iff T^* = f'$$

Proof: The theorem is symmetric in f and f', so it suffices to show $T^* = f' \implies T^* = f$. If $T^* = f'$ but $T^* \neq f$, then let $(x, y) \in f \setminus T^*$. Since $\mathcal{X} = D(f') = D(T^*)$, there exists $z \neq y$ for which $(x, z) \in T^* \subset f$. Hence $\{(x, y), (x, z)\} \subset f$, contradicting that f is a function. □

Theorem (Uniqueness)**.** *No row of a performance table contains any element more than once.*

Proof by contradiction: show $\mathcal{A}(f)_y = A(f')_y$ implies $\mathcal{A}^n_f(\emptyset) = \mathcal{A}^n_{f'}(\emptyset)$ via transfinite induction (for $n \geq 0$; note it is trivially true when $n = 0$). Therefore $\mathcal{A}(f) = \mathcal{A}(f')$ and $f = \mathcal{A}(f)^* = \mathcal{A}(f')^* = f'$ (via the recursion theorem), which contradicts $f \neq f'$. Assume $\mathcal{A}^n_f(\emptyset) = \mathcal{A}^n_{f'}(\emptyset)$ for all $n < \alpha$. If α is a limit ordinal, $\mathcal{A}^\alpha_f(\emptyset) = \mathcal{A}^\alpha_{f'}(\emptyset)$ follows by definition. If $\alpha > 0$ is not a limit ordinal, then (via the recursion theorem),

$$\begin{aligned}\mathcal{A}^\alpha_f(\emptyset) &= \mathcal{A}_f(T)\\ \mathcal{A}^\alpha_{f'}(\emptyset) &= \mathcal{A}_{f'}(T)\end{aligned}$$

where $T = \mathcal{A}^{\alpha-1}_f(\emptyset) = \mathcal{A}^{\alpha-1}_{f'}(\emptyset)$ and thus $T^* \subset f \cap f'$. It follows (via Lemma 1) that either $T^* = f = f'$ or else $T \subset \mathcal{T}(f) \cap \mathcal{T}(f')$. In the former case, $\mathcal{A}^\alpha_f(\emptyset) = T = \mathcal{A}^\alpha_{f'}(\emptyset)$. In the latter case,

$$\begin{aligned}\mathcal{A}^\alpha_f(\emptyset) &= T \parallel (x, f(x))\\ \mathcal{A}^\alpha_{f'}(\emptyset) &= T \parallel (x, f'(x))\end{aligned}$$

where $x = g(T)$ and $\mathcal{A}$ corresponds to search operator g. Moreover, $f(x) = f'(x)$ since $\mathcal{A}(f)_y = \mathcal{A}(f')_y$ by assumption. Hence $\mathcal{A}^\alpha_f(\emptyset) = \mathcal{A}^\alpha_{f'}(\emptyset)$. □

Theorem (Completeness)**.** *Given any search algorithm $\mathcal{A}$, and any performance sequence S with domain $I(\lfloor \mathcal{X} \rfloor)$, there exists $f \in \mathcal{Y}^{\mathcal{X}}$ such that*

$$\pi_{\lfloor \mathcal{X} \rfloor}(\mathcal{A}(f)_y) = S$$

Proof: Let $\mathcal{A}$ correspond to search operator g, and let f be any function satisfying

$$\bigcup_{n \leq \lfloor \mathcal{X} \rfloor} T^*_n \subset f$$

where

$$T_n = \begin{cases} \emptyset & \text{if } n = 0\\ \bigcup_{m<n} T_m & \text{if } n \text{ is a limit ordinal}\\ T_{n-1} \| \, (g(T_{n-1}), S(n-1)) & \text{otherwise}\end{cases}$$

Use transfinite induction to show

$$n \leq \lfloor \mathcal{X} \rfloor \implies \mathcal{A}^n_f(\emptyset) = T_n$$

Note that it is trivially true when $n = 0$. Suppose it is true for all $n < \alpha \leq \lfloor \mathcal{X} \rfloor$. If α is a limit ordinal, then $\mathcal{A}^\alpha_f(\emptyset) = T_\alpha$ is true by definition. If $\alpha > 0$ is not a limit ordinal, then (via the recursion theorem and definition of f),

$$\mathcal{A}^\alpha_f(\emptyset) = \mathcal{A}_f(\mathcal{A}^{\alpha-1}_f(\emptyset)) = \mathcal{A}_f(T_{\alpha-1}) = T_{\alpha-1} \| \, (g(T_{\alpha-1}), S(\alpha-1)) = T_\alpha$$

It follows (via the recursion theorem) that,

$$\pi_{\lfloor \mathcal{X} \rfloor}(\mathcal{A}(f)_y) \;=\; \mathcal{A}_f^{\lfloor \mathcal{X} \rfloor}(\emptyset)_y \;=\; (T_{\lfloor \mathcal{X} \rfloor})_y \;=\; S$$

□

Definition: A *permutation* σ is a bijection from $\mathcal{X}$ to $\mathcal{X}$. Corresponding to σ is the *permutation* σf *of* f defined by $\sigma f(x) = f(\sigma^{-1}(x))$. To say f' *is a permutation of* f is to assert $f' = \sigma f$ for some permutation σ. A set $F \subset \mathcal{Y}^{\mathcal{X}}$ is *closed* if for every permutation σ,

$$f \in F \implies \sigma f \in F$$

The *permutation* $\sigma\mathcal{A}$ *of search algorithm* $\mathcal{A}$ is the search algorithm corresponding to search operator σg defined by $\sigma g(T) = \sigma^{-1}(g(\sigma_x(T)))$ where $\mathcal{A}$ corresponds to search operator g, and where σ_x maps traces to traces according to

$$\begin{aligned} \sigma_x(\emptyset) &= \emptyset \\ T(n) = (x, y) &\implies \sigma_x(T)(n) = (\sigma(x), y) \end{aligned}$$

(i.e., σ_x applies σ to the x values of a trace and leaves the y values alone).

Theorem (Duality). *For every search algorithm $\mathcal{A}$, and $f \in \mathcal{Y}^{\mathcal{X}}$,*

$$\sigma_x((\sigma\mathcal{A})(f)) = \mathcal{A}(\sigma f)$$

In particular, $(\sigma\mathcal{A})(f)_y = \mathcal{A}(\sigma f)_y$.

Proof: Use transfinite induction to show $\sigma_x((\sigma\mathcal{A})_f^n(\emptyset)) = \mathcal{A}_{\sigma f}^n(\emptyset)$ (note that it is trivially true when $n = 0$). It follows that $\sigma_x((\sigma\mathcal{A})(f)) = \mathcal{A}(\sigma f)$ which proves the theorem (σ_x does not change the y values in a trace). Suppose it is true for all $n < \alpha$. If α is a limit ordinal, then $\sigma_x((\sigma\mathcal{A})_f^\alpha(\emptyset)) = \mathcal{A}_{\sigma f}^\alpha(\emptyset)$ follows by definition (using the inductive hypothesis). If $\alpha > 0$ is not a limit ordinal, then (via the recursion theorem and the inductive hypothesis),

$$\begin{aligned} \sigma g((\sigma\mathcal{A})_f^{\alpha-1}(\emptyset)) &= \sigma^{-1} \circ g(\sigma_x((\sigma\mathcal{A})_f^{\alpha-1}(\emptyset))) \\ &= \sigma^{-1} \circ g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset)) \\ f \circ \sigma g((\sigma\mathcal{A})_f^{\alpha-1}(\emptyset)) &= f(\sigma^{-1} \circ g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset))) \\ &= \sigma f(g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset))) \end{aligned}$$

Therefore

$$\begin{aligned} (\sigma\mathcal{A})_f^\alpha(\emptyset) &= (\sigma\mathcal{A})_f^{\alpha-1}(\emptyset) \,||\, (\sigma g((\sigma\mathcal{A})_f^{\alpha-1}(\emptyset)), f \circ \sigma g((\sigma\mathcal{A})_f^{\alpha-1}(\emptyset))) \\ &= (\sigma\mathcal{A})_f^{\alpha-1}(\emptyset) \,||\, (\sigma^{-1} \circ g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset)), \sigma f(g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset)))) \\ \sigma_x((\sigma\mathcal{A})_f^\alpha(\emptyset)) &= \sigma_x((\sigma\mathcal{A})_f^{\alpha-1}(\emptyset)) \,||\, (g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset)), \sigma f(g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset)))) \\ &= \mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset) \,||\, (g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset)), \sigma f \circ g(\mathcal{A}_{\sigma f}^{\alpha-1}(\emptyset))) \\ &= \mathcal{A}_{\sigma f}^\alpha(\emptyset) \end{aligned}$$

(using the inductive hypothesis again). □

Definition: Search algorithm $\mathcal{A}$ is *efficient* if $D(\mathcal{A}(f)) = I(\lfloor \mathcal{X} \rfloor)$ for all $f \in \mathcal{Y}^{\mathcal{X}}$.

Note that efficient search algorithms clearly exist, since the cardinality of $\lfloor \mathcal{X} \rfloor$ matches the cardinality of $\mathcal{X}$. For any fixed bijection $b : I(\lfloor \mathcal{X} \rfloor) \to \mathcal{X}$, a trivial example corresponds to enumeration; let $g(T) = b(n)$ where $D(T) = I(n)$. Moreover, efficient search algorithms are not limited to enumeration.

Lemma 2. *If $\mathcal{A}$ is efficient, then $\sigma\mathcal{A}$ is efficient for every permutation σ. If traces T, T' are total, then $T_y = T'_y \implies T^*$ is a permutation of T'^*.*

Proof: The first assertion of the theorem is a consequence of the Duality Theorem. Since $\mathcal{A}$ is efficient, it follows that

$$I(\mathcal{X}) \;=\; D(\mathcal{A}(\sigma f)) \;=\; D(\sigma_x((\sigma\mathcal{A})(f))) \;=\; D((\sigma\mathcal{A})(f))$$

To establish the second assertion of the theorem, let the image of $k = T_x(i)$ under σ be $j = T'_x(i)$. Then $\sigma^{-1}(j) = k$, and

$$(\sigma T^*)(j) \;=\; T^*(k) \;=\; T_y(i) \;=\; T'_y(i) \;=\; T'^*(j)$$

□

4 No Free Lunch

The No Free Lunch theorem must necessarily take on a different character when the domain $\mathcal{X}$ is infinite, since the smallest ordinal α such that

$$\mathcal{A}^{\alpha}_{f}(\emptyset)^* = f$$

can depend both on $\mathcal{A}$ and on f, which is not the case when $\mathcal{X}$ is finite.[1] The smallest ordinal for which the above could possibly be true is $\alpha = \lfloor \mathcal{X} \rfloor$, since for every smaller ordinal the cardinalities of the left hand side and right hand side above would differ.

Theorem (Weak NFL)**.** *Given search algorithms $\mathcal{A}$, $\mathcal{A}'$ and function $f \in \mathcal{Y}^{\mathcal{X}}$, there exists a function $f' \in \mathcal{Y}^{\mathcal{X}}$ such that $\pi_{\lfloor \mathcal{X} \rfloor}(\mathcal{A}(f)_y) = \pi_{\lfloor \mathcal{X} \rfloor}(\mathcal{A}'(f')_y)$.*

Proof: The proof is a straightforward application of the Completeness Theorem □

Definition: A *performance measure with respect to a set $F \subset \mathcal{Y}^{\mathcal{X}}$* is any function μ_F defined over the collection of all search algorithms such that $\mu_F(\mathcal{A})$ is a function of the multiset $\{\!\{\mathcal{A}(f)_y : f \in F\}\!\}$. Search algorithms *perform equally well on F* if they are evaluated identically by every performance measure with respect to F.

Theorem (NFL)**.** *Every efficient search algorithm performs equally well on F if and only if F is closed.*

[1] Consider the following non-efficient algorithm: let $\mathcal{X}$ be the positive integers and consider $\mathcal{A}$ which explores $1, 3, 5, \ldots$ before moving on to $2, 4, 6, \ldots$ unless $f(1) = 1$ in which case $\mathcal{A}$ enumerates $1, 2, 3, \ldots$.

Proof: Appealing to the Completeness Theorem, every row in a performance table corresponding to an efficient search algorithm contains every performance sequence with domain $I(\lfloor \mathcal{X} \rfloor)$. Appealing to the Uniqueness Theorem, each element in such a row is a performance sequence with domain $I(\lfloor \mathcal{X} \rfloor)$ and the elements in a row are unique. Therefore, the row corresponding to an efficient search algorithm is actually a set (as opposed to a multiset).
Let F be closed. If for efficient search algorithms $\mathcal{A}$ and $\mathcal{A}'$ the sets $S = \{\mathcal{A}(f)_y : f \in F\}$ and $S' = \{\mathcal{A}'(h)_y : h \in F\}$ are equal, then the search algorithms must perform equally well on F. By the Weak NFL Theorem, given f there exists h such that $\mathcal{A}(f)_y = \mathcal{A}'(h)_y$. It follows that h is a permutation of f (Lemma 2). Therefore $f \in F \implies h \in F$. Hence $S \subset S'$. The reverse containment follows by symmetry.
Conversely, assume by way of contradiction that all efficient search algorithms perform equally well on F which is not closed; let σ and f be such that $f \in F$ and $\sigma f \notin F$. Fix an efficient search algorithm $\mathcal{A}$, and consider the performance measure

$$\mu_F(A) \;=\; [\mathcal{A}(f)_y \in \{A(h)_y : h \in F\}]$$

where [*expression*] is 1 if *expression* is true, and 0 otherwise. Since $\mu_F(\mathcal{A}) = 1$, it must happen that $\mu_F(A) = 1$ for the particular choice $A = \sigma^{-1}\mathcal{A}$ (by Lemma 2, A is efficient). Therefore,

$$\mathcal{A}(f)_y \in \{(\sigma^{-1}\mathcal{A})(h)_y : h \in F\}$$

which leads to a contradiction as follows. Appealing to the duality theorem,

$$\{(\sigma^{-1}\mathcal{A})(h)_y : h \in F\} = \{\mathcal{A}(\sigma^{-1}h)_y : h \in F\} = \{\mathcal{A}(h)_y : \sigma h \in F\}$$

Appealing to the uniqueness theorem, $\mathcal{A}(f)_y \in \{\mathcal{A}(h)_y : \sigma h \in F\} \implies \sigma f \in F$. □

Definition: Performance measure μ_F is *dominated (by ordinal β)* if there exists $\beta < \lfloor \mathcal{X} \rfloor$ and function μ such that for all search algorithms $\mathcal{A}$,

$$\mu_F(\mathcal{A}) \;=\; \mu(\{\!\{\, \pi_\beta(\mathcal{A}(f))_y : f \in F \}\!\})$$

Theorem (Dominated NFL)**.** *If a performance measure with respect to a closed set F is dominated, then it evaluates every search algorithm the same.*

Proof: Given any search algorithm $\mathcal{A}$ and ordinal $\beta < \lfloor \mathcal{X} \rfloor$, it suffices to show an efficient search algorithm $\mathcal{B}$ exists such that $\pi_\beta(\mathcal{A}(f)) = \pi_\beta(\mathcal{B}(f))$ for all f. In that case search algorithms can be regarded as efficient without loss of generality ($\mathcal{B}$ is a surrogate for $\mathcal{A}$), and the NFL Theorem may be applied.
Let $\mathcal{A}$ correspond to search operator g, and let function $b : I(\lfloor \mathcal{X} \rfloor) \to \mathcal{X}$ be bijective. Let $\mathcal{B}$ correspond to search operator h defined by

$$h(T) \;=\; \begin{cases} g(T) & \text{if } D(T) \subset I(\beta) \\ b(\min\{\, \gamma : b(\gamma) \notin D(T^*)\}) & \text{otherwise} \end{cases}$$

Note that $\pi_\beta(\mathcal{A}(f)) = \pi_\beta(\mathcal{B}(f))$ since $\mathcal{A}$ and $\mathcal{B}$ have search operators which agree on all traces whose domains are contained in $I(\beta)$. Moreover, $\mathcal{B}$ is efficient if $\mathcal{X}$ is finite. If $\mathcal{X}$ is infinite, use transfinite induction to show that for all $0 \leq n \leq \lfloor \mathcal{X} \rfloor$,

$$b(I(n)) \;\subset\; D(\mathcal{B}_f^{\beta+n}(\emptyset)^*)$$

Note that it is trivially true when $n = 0$. Suppose it is true for all $n < \alpha$. If α is a limit ordinal, then so too is $\beta + \alpha$ and therefore $b(I(\alpha)) \subset D(\mathcal{B}_f^{\beta+\alpha}(\emptyset)^*)$ follows by definition. If $\alpha > 0$ is not a limit ordinal, then (via the Recursion Theorem and the definition of $\mathcal{B}_f$),

$$D(\mathcal{B}_f^{\beta+\alpha}(\emptyset)^*) \ = \ D(\mathcal{B}_f^{\beta+\alpha-1}(\emptyset)^*) \cup \{b(\min\{\,\gamma : b(\gamma) \notin D(\mathcal{B}_f^{\beta+\alpha-1}(\emptyset)^*)\})\}$$

By hypothesis, $b(I(\alpha-1)) \subset D(\mathcal{B}_f^{\beta+\alpha-1}(\emptyset)^*)$ is contained in the right hand side above. If $b(\alpha-1) \in D(\mathcal{B}_f^{\beta+\alpha-1}(\emptyset)^*)$, then the inductive argument is complete. Otherwise, the ordinal $\gamma = \alpha - 1$ is smallest such that $b(\gamma) \notin D(\mathcal{B}_f^{\beta+\alpha-1}(\emptyset)^*)$, in which case $b(\alpha-1)$ is contained in the right hand side displayed above (which completes the inductive argument).

Finally (keeping in mind that $\mathcal{X}$ is infinite),

$$\mathcal{X} \ \subset \ b(I(\lfloor\mathcal{X}\rfloor)) \ \subset \ D(\mathcal{B}_f^{\beta+\lfloor\mathcal{X}\rfloor}(\emptyset)^*)$$

and $\beta + \lfloor\mathcal{X}\rfloor = \lfloor\mathcal{X}\rfloor$ follows from $n < \lfloor\mathcal{X}\rfloor \iff \overline{n} < |\mathcal{X}|$ (for every ordinal n). □

5 Discussion

We have presented a general set-theoretic version of NFL which speaks to underlying symmetries of Black Box search without regard to any particular purpose to which those symmetries may be put. Although applications are not our main concern, we indicate how the classical "Non-Uniform NFL-theorem" of Igel and Toussaint (2004) is implied by our results.

Assume domains and co-domains are finite (therefore all search algorithms are efficient). According to our NFL Theorem, if F is closed, then the left hand side below is independent of $\mathcal{A}$ for every function μ; the following chain of equalities results from making special choices for the arbitrary functions μ, ξ, ϕ

$$\begin{aligned}
\mu(\{\!\{\mathcal{A}(f)_y : f \in F\}\!\}) &= \sum_{f\in F} \xi(\mathcal{A}(f)_y) \\
&= \sum_{f\in F} \phi(\mathcal{A}(f)_y)\,\psi(\mathcal{A}(f)_y) \\
&= \sum_{f\in F} w(\{\!\{\mathcal{A}(f)_y(n) : n < \lfloor\mathcal{X}\rfloor\}\!\})\,\psi(\mathcal{A}(f)_y) \\
&= \sum_{f\in F} w(\{\!\{f(x) : x \in \mathcal{X}\}\!\})\,\psi(\mathcal{A}(f)_y)
\end{aligned}$$

The last equality above follows from the fact that $\mathcal{A}(f) : \lfloor\mathcal{X}\rfloor \longrightarrow f$ is a bijection, which implies the arguments to w are the same multiset. Assuming F is closed, the fact that the last displayed summation above is independent of $\mathcal{A}$, for every choice of w and ψ, can be phrased as: all algorithms have identical expected performance as measured by arbitrary (but fixed) ψ with respect to an arbitrary (but fixed) probability distribution over any closed set F of functions, provided the probability of f as given by w depends only on $\{\!\{f(x) : x \in \mathcal{X}\}\!\}$.[2]

[2]This direction of the "Non-Uniform NFL-theorem" – i.e., F closed implies the last displayed summation above is independent of $\mathcal{A}$, for every choice of w and ψ – is a consequence of results found in both Radcliffe and Surry (1995) and in Schumacher (2000), but neither point it out and both make comments suggesting they may be unaware of the result. For the converse, see Schumacher (2000) page 54.

Our version of set-theoretic NFL assumes deterministic algorithms. In practice, that can sometimes be an annoyance rather than a limitation. Oftentimes randomness is a fiction – a deterministic pseudo random number generator is used – and sometimes making a random choice from a collection of deterministic algorithms suffices to model stochastic behavior. As noted by Schumacher (2000), if the probability that a given stochastic algorithm is equivalent to a deterministic algorithm $\mathcal{A}$ is described by $d\lambda(\mathcal{A})$, then the expected overall performance of the randomized algorithm is

$$\int \mu(\{\!\{\mathcal{A}(f)_y : f \in F\}\!\})\, d\lambda(\mathcal{A}) \;=\; c \int d\lambda(\mathcal{A}) \;=\; c$$

where performance $c = \mu(\{\!\{\mathcal{A}(f)_y : f \in F\}\!\})$ is algorithm-independent as guaranteed by NFL, assuming algorithms are efficient or the performance measure is dominated, and F is closed (subject to measurability conditions, to make sense of integration). It is certainly possible – as already demonstrated by this and the previous paragraph – for deterministic set-theoretic results to admit probabilistic interpretations. Our point is not that stochasticity should be swept under the rug, but the symmetry results we have presented need not speak only to situations which are devoid of probability.

In closing, we note how the Dominated NFL theorem might potentially obviate concerns regarding the algorithmic content of search algorithms. For example, if the performance measures involved are serendipitously dominated by some ordinal β that makes algorithmic concerns regarding $\mathcal{A}_f^\beta(\emptyset)$ irrelevant, then whether or not one feels good about $\mathcal{A}(f)$ is a non-issue. In particular, if β were a *finite ordinal*, then search algorithms need only operate for a finite number of steps, and their search operators therefore need only be defined on traces of finite length. If $\mathcal{X}$ and $\mathcal{Y}$ are countable, then every finite trace is finitely representable. Moreover, it could be natural to assume f is an oracle. This is obviously not sufficient to guarantee Turing computability of $\mathcal{A}_f^\beta(\emptyset)$ for *every* $\mathcal{A}$ relative to *every* f, but it may provide initial context in which to begin the investigation of computability questions.

6 Acknowledgements

Some of the ideas for this paper were formed while M. D. Vose and Alden H. Wright were visiting The University of Birmingham; we are grateful for the gracious support provided by the School of Computer Science. We would also like to acknowledge the critical comments made by anonymous revievers which have significantly improved the exposition of this paper.

References

Auger, A. and Teytaud, O. (2007). Continuous lunches are free! In *Proceedings of the 9th annual conference on Genetic and evolutionary computation, GECCO-2007*, pages 916–922, New York. ACM Press.

Devlin, K. (1994). *The Joy of Sets: Fundamentals of Contemporary Set Theory*. Springer.

Droste, S., Jansen, T., and Wegener, I. (1999). Perhaps not a free lunch but at least a free appetizer. In *Proceedings of the First Genetic and Evolutionary Computation Conference(GECCO '99)*, pages 833–839, San Francisco CA. Morgan Kaufmann Publishers, Inc.

Hewitt, E. and Stromberg, K. (1965). *Real and Abstract Analysis*. Springer.

Igel, C. and Toussaint, M. (2004). A no-free-lunch theorem for non-uniform distributions of target functions. *Journal of Mathematical Modelling and Algorithms*, 3(4):313–322.

Köppen, M. (2000). Some technical remarks on the proof of the no free lunch theorem. In *Proceedings of the Joint Conference on Information Sciences (JCIS 2000), Atlantic City, NJ*, pages 1020–1024.

Köppen, M., Wolpert, D., and Macready, M. (2001). Remarks on a recent paper on the "no free lunch" theorems. *IEEE Transactions on Evolutionary Computation*, 5(3):295–296.

Radcliffe, N. and Surry, P. (1995). Fundamental limitations on search algorithms: Evolutionary computing in perspective. In *Computer Science Today*, pages 275–291.

Schumacher, C. (2000). *Black Box Search - Framework and Methods*. PhD thesis, The University of Tennessee, Knoxville.

Wolpert, D. and Macready, M. (1995). No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Wolpert, D. and Macready, M. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.