Continuous lunches are free plus the design of optimal optimization algorithms

Anne Auger, Olivier Teytaud

Continuous lunches are free plus the design of optimal optimization algorithms

Anne Auger and Olivier Teytaud
TAO Team - INRIA Saclay,
LRI - Paris-Sud University
91405 Orsay Cedex, France
{firstname.lastname}@inria.fr
Fax: +33 1.69.15.42.40, Tel: +33 1.69.15.34.71

March 21, 2009

Abstract

This paper analyses extensions of No-Free-Lunch (NFL) theorems to countably infinite and uncountable infinite domains and investigates the design of optimal optimization algorithms.

The original NFL theorem due to Wolpert and Macready states that, for finite search domains, all search heuristics have the same performance when averaged over the uniform distribution over all possible functions. For infinite domains the extension of the concept of distribution over all possible functions involves measurability issues and stochastic process theory. For countably infinite domains, we prove that the natural extension of NFL theorems, for the current formalization of probability, does not hold, but that a weaker form of NFL does hold, by stating the existence of non-trivial distributions of fitness leading to equal performances for all search heuristics. Our main result is that for continuous domains, NFL does not hold. This free-lunch theorem is based on the formalization of the concept of random fitness functions by means of random fields.

We also consider the design of optimal optimization algorithms for a given random field, in a black-box setting, namely, a complexity measure based solely on the number of requests to the fitness function. We derive an optimal algorithm based on Bellman's decomposition principle, for a given number of iterates and a given distribution of fitness functions. We also approximate this algorithm thanks to a Monte-Carlo planning algorithm close to the UCT (Upper Confidence Trees) algorithm, and provide experimental results.

Key words: No-Free-Lunch, Kolmogorov's extension Theorem, Expensive Optimization, Dynamic Programming, Complexity, Bandit-Based Monte-Carlo Planning.

1 Introduction

Search heuristics like evolutionary algorithms, tabu search, grid search are general heuristics that can be applied to any objective function. Much research is devoted to develop heuristics that are superior to others when the target functions belong to a certain class of problems.

The No-Free-Lunch (NFL) theorem for optimization [42], introduced for fitness functions defined on a finite domain, rules out statements that some search heuristics have some advantages when performances are averaged uniformly among all possible functions. Controversial discussions on the usefulness of search heuristics ensued from this theorem [8, 36]. Droste et al. argue that the NFL scenario is not a realistic one and they show that for realistic black box scenarios, NFL will not hold [13]. Igel and Toussaint [25] show that classes of functions relevant in practice are not likely to satisfy the (sharpened) NFL scenario [38]. Corne and Knowles [7] show that NFL does not hold in the multi-objective case.

However, a basic assumption of NFL theorems is that the search space is finite. In that case "averaging uniformly among all possible functions" has a clear meaning: the average is made with respect to the cardinal and uniform random fitness can be easily defined as a mapping from a probability space to the finite set of all possible functions. It is more tricky to give a meaning to uniform average among functions when the domain is countably infinite or continuous (uncountable infinite). More generally, it is difficult to define a random mapping from a probability space to the (infinite) space of all possible functions. Doing so in a proper way, for the current formalization of probability, involves measurability and stochastic process theory [9, 31, 4].

A related question is the existence of optimal optimization algorithms assuming a priori information on the distribution of objective functions and given a finite number of iterations. Such optimal optimization algorithms require representation of all the past information. This has already been investigated in the field of global optimization; for example, [26] proposes the EGO algorithm (Efficient Global Optimization) based on a modelization by Gaussian processes, and the criterion is improved in [40]. In these two cases, the expensive optimization framework is the opportunity for deriving complex algorithms with large computation times; as the fitness function is assumed very expensive, the goal is solely the optimization for a given number of iterations, and not the computational cost of choosing the iterates. However, in these papers, the suboptimality is due to the principle itself; even though all priors would be true and the algorithms could be run without any numerical trouble, the algorithm would not be mathematically optimal.

In this paper, we investigate NFL results for countably infinite and continuous domains and study the optimal framework, for a given distribution of problems, a given criterion, and a given number of iterates, at least in the limit of a large computational power for choosing each iterate. In Section 2 we start by reviewing NFL theorems for finite domains. We recall important definitions from measure theory required when dealing with uncountable infinite domains.

We also introduce definitions required for the extensions of (N)FL theorems to countably infinite and uncountable domains. In Section 3, we show that the natural (for the current formalization of probability) extension of NFL results, for countably infinite domains does not hold but show that a weaker form of NFL does hold. In Section 4, we show that this weaker form of NFL does not hold anymore for continuous domains. In Section 5, we derive an optimal optimization algorithm assuming a prior distribution on the distribution of functions and a finite number of iterates. We also approximate this algorithm using a Monte-Carlo planning algorithm and show its tractability.

2 Preliminaries on the NFL theorems

In this section, we present (i) the finite case (ii) the definitions required for the rest of the paper.

2.1 Finite lunches

We present in this section NFL theorems for objective functions mapping a finite domain $\mathcal{X}$, with cardinal $|\mathcal{X}|$, into a finite codomain $\mathcal{Y} \subset \mathbb{R}$ [42, 38, 25, 13, 7]. The search heuristics considered are randomized or deterministic and it is assumed that they are non-repeating. In practice, this can be ensured by archiving the different inputs. For any integer m in $\{1, \ldots, |\mathcal{X}|\}$, the vector $(x_1, \ldots, x_m)$ represents the m first iterates of a search algorithm and the vector $(f(x_1), \ldots, f(x_m))$ their associated objective values for a given function f mapping $\mathcal{X}$ into $\mathcal{Y}$. The performance of an algorithm a after m iterations is measured using the vector of cost values

$$Y(f, m, a) = \langle f(x_1), \ldots, f(x_m) \rangle \tag{1}$$

where we stick to the notations used in [25]. Let c denote a performance measure mapping the vector of cost values to the real numbers. The function c can be for instance the minimum value of the vector $Y(f, m, a)$ or the number of iterations before reaching a given value.

We denote by $\Pi(\mathcal{X})$ the set of permutations on $\mathcal{X}$. A set of functions $\mathcal{F}$ is closed under permutation (c.u.p.) if for any $f \in \mathcal{F}$ and any permutation $\pi \in \Pi(\mathcal{X})$, $f \circ \pi \in \mathcal{F}$.

The original NFL for optimization was stated for the set of all possible functions on $\mathcal{X}$, i.e., $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$ [42] and has been generalized for c.u.p. subsets [38, 13]:

Theorem 2.1 (NFL for c.u.p. subsets)**.** *Let $\mathcal{F}$ be a subset of $\mathcal{Y}^{\mathcal{X}}$. Then, for any two algorithms a and b, any m in $\{1, \ldots, |\mathcal{X}|\}$, any performance measure c, any $k \in \mathbb{R}$*

$$\sum_{f \in \mathcal{F}} \delta(k, c(Y(f, m, a))) = \sum_{f \in \mathcal{F}} \delta(k, c(Y(f, m, b))) \tag{2}$$

iff $\mathcal{F}$ is c.u.p.[1]

In the previous theorem, averaging is done by uniform summation over $\mathcal{F}$ which implicitly means that all functions in $\mathcal{F}$ are equally likely. An equivalent point of view is to define f as a random variable taking values in the set of functions $\mathcal{F}$, with the same probability for each function in $\mathcal{F}$. The vector $Y(f, m, a)$ defined in Eq. 1 is then a random vector and an equivalent formulation to "for all k in $\mathbb{R}$, Eq. 2 holds" is

$$\forall k \in \mathbb{R}, \;\; \mathbb{P}\left(c(Y(f,m,a)) = k\right) = \mathbb{P}\left(c(Y(f,m,b)) = k\right) \;\; , \tag{3}$$

where the random variable f is uniformly distributed on $\mathcal{F}$, i.e., for any $f_0 \in \mathcal{F}$, $\mathbb{P}(f = f_0)$ is constant and equals to $1/|\mathcal{F}|$. Eq. 3 is an equivalent way to state that the distributions of $c(Y(f, m, a))$ and $c(Y(f, m, b))$ are the same and an equivalent formulation for Theorem 2.1 is

Theorem 2.2 (NFL for c.u.p. subsets)**.** *Let $\mathcal{F}$ be a subset of $\mathcal{Y}^{\mathcal{X}}$ and f a random variable uniformly distributed on $\mathcal{F}$. Then, for any two algorithms a and b, any m in $\{1, \ldots, |\mathcal{X}|\}$ and any performance measure c, $c(Y(f, m, a))$ and $c(Y(f, m, b))$ follow the same distribution iff $\mathcal{F}$ is c.u.p.*

A generalization of this NFL theorem to non uniform distributions of fitness has been presented in [25]. They consider the histogram h_{f_0} of a function $f_0 \in \mathcal{Y}^{\mathcal{X}}$ defined for each $y \in \mathcal{Y}$ as the cardinal of the pre-image $f_0^{-1}(y)$, i.e.,

$$h_{f_0} : y \in \mathcal{Y} \to |f_0^{-1}(y)| \;\; .$$

Then, Theorem 2.2 holds for random variables f having distributions constant for functions having the same histogram [25], i.e.,

$$\text{if } h_{f_0} = h_{f_1}, \mathbb{P}(f = f_0) = \mathbb{P}(f = f_1). \tag{4}$$

One simple random variable satisfying the condition in Eq. 4 is defined as follows: consider the c.u.p. subset $\mathcal{F}_{f_0}^{\Pi}$ defined for a function $f_0 \in \mathcal{Y}^{\mathcal{X}}$ as

$$\mathcal{F}_{f_0}^{\Pi} = \{f_0 \circ \pi \text{ for } \pi \in \Pi(\mathcal{X})\} \tag{5}$$

and define the corresponding random variable $f_{\mathcal{F}_{f_0}^{\Pi}}$ as

$$f_{\mathcal{F}_{f_0}^{\Pi}} = f_0 \circ \pi \;\; , \tag{6}$$

where π is a random variable uniformly distributed on $\Pi(\mathcal{X})$. Then $f_{\mathcal{F}_{f_0}^{\Pi}}$ satisfies the histogram condition (Eq. 4) and also Theorem 2.2. Averaging with any unequal weights multiple random variables of this form (with different functions f_0) provides a random variable satisfying Eq. 4 without, in general, satisfying the uniformity on a c.u.p. subset condition required in Theorem 2.2.

[1] The Kronecker delta function, δ, is defined as $\delta(a, b) = 1$ if $a = b$ and $\delta(a, b) = 0$ otherwise.

2.2 Generalization

The generalization of NFL theorems for countably infinite and uncountable infinite domains calls upon measurability theory. We summarize here some basics that we will need in the rest of the paper. Consider a probability space $(\Omega, \mathfrak{A}, \mathbb{P})$, where $\mathfrak{A}$ is a σ-algebra on Ω and $\mathbb{P}$ a probability measure defined on $\mathfrak{A}$. When $\mathcal{X}$ is finite, a mapping $X : \Omega \to \mathcal{X}$ is a random variable[2] if for any x_0, $\{\omega \in \Omega, X(\omega) = x_0\}$, denoted in short $\{X = x_0\}$, is in $\mathfrak{A}$. $\mathbb{P}_X : x_0 \mapsto \mathbb{P}_X(x_0) = \mathbb{P}(\{X = x_0\})$ is then well defined and called the distribution function of X. This definition is extended to $\mathcal{X} = \mathbb{R}^n$ using measurability. A mapping $X : \Omega \to \mathbb{R}^n$ is a random variable if it is a measurable function, i.e., a function such that $X^{-1}(E) = \{\omega \in \Omega, X(\omega) \in E\}$ is in $\mathfrak{A}$ for any measurable subset E of $\mathbb{R}^n$. This implies that $\mathbb{P}_X(E) = \mathbb{P}(\{\omega \in \Omega; X(\omega) \in E\})$ is defined for any measurable $E \subset \mathbb{R}^n$. When $\mathcal{X}$ has no natural measure, defining something similar to a random variable is more difficult. For defining a random variable with values in $\mathcal{Y}^{\mathcal{X}}$ with $\mathcal{Y} \subset \mathbb{R}$ and $\mathcal{X}$ finite, one can simply use the natural isomorphism from $\mathcal{Y}^{\mathcal{X}}$ to $\mathbb{R}^n$ where n is the cardinal of $\mathcal{X}$. Doing so induces a measure on $\mathcal{Y}^{\mathcal{X}}$ that is the usual measure in this case. When $\mathcal{X}$ is infinite but countable, then Kolmogorov's extension theorem provides a natural extension. But when $\mathcal{X}$ is uncountable, typically $\mathcal{X} = \mathbb{R}^n$, we need stochastic processes or random fields [39][3]. This will be detailed in Section 4.

Performance measurement

In Theorem 2.2, performance is measured using the distribution of $c(Y(f, m, a))$: two algorithms a and b are equivalent if the distributions of $c(Y(f, m, a))$ and $c(Y(f, m, b))$ are the same. Lemma 2.3 below shows that it is equivalent to requiring that the distributions of $Y(f, m, a)$ and $Y(f, m, b)$ are the same.

Lemma 2.3. *Let $\mathcal{M}$ be the set of measurable functions from $\mathbb{R}^n$ to $\mathbb{R}$ and $\mathcal{M}'$ the set of characteristic functions[4] of measurable sets of the form $]-\infty, t_1] \times]-\infty, t_2] \times \cdots \times]-\infty, t_n] \subset \mathbb{R}^n$. Then, for any family $\mathcal{A}$ of random variables in $\mathbb{R}^n$, the following statements are equivalent:*

$$\forall (a_1, \ldots, a_n) \in \mathcal{A}, \quad \forall c \in \mathcal{M}, c(a_1, \ldots, a_n) \text{ have the same distribution} \tag{7}$$

$$\forall (a_1, \ldots, a_n) \in \mathcal{A}, \quad \forall c \in \mathcal{M}', c(a_1, \ldots, a_n) \text{ have the same distribution} \tag{8}$$

$$\forall (a_1, \ldots, a_n) \in \mathcal{A}, \quad \forall c \in \mathcal{M}', \mathbb{E}\left(c(a_1, \ldots, a_n)\right) \text{ are equal} \tag{9}$$

$$\forall (a_1, \ldots, a_n) \in \mathcal{A}, \quad (a_1, \ldots, a_n) \text{ have the same distribution} \tag{10}$$

Proof. Eq. 10 implies Eq. 7 which in turn implies Eq. 8 which in turn implies Eq. 9. We just have to show that Eq. 9 implies Eq. 10. Eq. 9 states that all the

[2] We implicitly use here the standard σ-algebra on finite sets. With other σ-algebra, more restrictive definitions of a random variable could be stated.

[3] Stochastic processes and random fields are formally very similar; however, the term "random field" is usually prefered when the dimension of $\mathcal{X}$ is greater than 1.

[4] A characteristic function of a set A is defined to be identically one on A and is zero elsewhere.

$(a_1, \ldots, a_n)$ have the same cumulative distribution function, and therefore the same distribution. This implies Eq. 10. □

For the sake of simplicity we will prefer the last statement in our NFL definitions: two algorithms will be equivalent if $Y(f, m, a)$ and $Y(f, m, b)$ follow the same distribution.

No-Free-Lunch definitions

We introduce here different notions of NFL. We generalize first the definition of No-Free-Lunch in order to extend results stated in Theorem 2.2. To do so, we consider the simple c.u.p. subset $\mathcal{F}_{f_0}^{\Pi}$ introduced in Eq. 5 and satisfying Theorem 2.2. A permutation π on $\mathcal{X}$ is a bijective (or one-to-one) mapping from $\mathcal{X}$ to $\mathcal{X}$. This definition also holds when $\mathcal{X}$ is uncountable. In some cases, we will have to consider measure-preserving permutations:

Definition 2.4 (Measure-preserving)**.** *Let $(\mathcal{X}, \mathfrak{B}, \mu)$ be a measure space, and $T : \mathcal{X} \to \mathcal{X}$ be a measurable mapping. We call T measure-preserving if for all $A \in \mathfrak{B}$, we have that $\mu(T^{-1}(A)) = \mu(A)$.*

When $\mathcal{X}$ is a discrete space and μ the cardinal measure ($\mu(A) = |A|$), any permutation is measure-preserving. In all the paper, we will restrict our attention to the cardinal measure when $\mathcal{X}$ is a discrete (finite or countable) domain. In $[0, 1]$, we will consider the Lebesgue measure.

For our first definition of NFL we consider distributions of functions $f_0 \circ \pi$ where π is a random permutation (see below for a formal definition). We will see other, more general (weaker) forms of NFL in the rest of the paper, and show that even weaker forms do not hold in the continuous case.

The intuitive idea of a random permutation π is a random variable with values in $\mathcal{X}^{\mathcal{X}}$. However, the definition of a random variable involves σ-algebras: a mapping with values in $\mathcal{X}^{\mathcal{X}}$ is a random variable if it is measurable. Therefore, we need a σ-algebra on $\mathcal{X}^{\mathcal{X}}$ for using this terminology. Such σ-algebras exist, but they have several weaknesses [4]. Instead, we will simply require that π is a measurable map as a function $(\omega, x) \in \Omega \times \mathcal{X} \mapsto \pi(x)$ (where, as usual, π implicitly depends on $\omega \in \Omega$). We extend Definition 2.4 to a random permutation $T : \Omega \times \mathcal{X} \to \mathcal{X}$ as follows: T is measure-preserving if $T(\omega)$ is measure-preserving in the sense of Definition 2.4 almost surely in ω. Our first No-Free-Lunch definition reads:

Definition 2.5. *Let $f_0 : \mathcal{X} \to \mathbb{R}$ be a measurable function and π a measure-preserving random permutation on $\mathcal{X}$. $\mathcal{NFL}(\mathcal{X}, \pi, f_0)$ holds iff for any integer m (smaller than $|\mathcal{X}|$ when $\mathcal{X}$ is finite) and any two optimization algorithms a and b, $Y(f_0 \circ \pi, m, a)$ and $Y(f_0 \circ \pi, m, b)$ follow the same distribution.*

A proper median of a deterministic mapping f_0 is a value M_{f_0} such that the measure of $\{x \in \mathcal{X}; f(x) > M_{f_0}\}$ is equal to the measure of $\{x \in \mathcal{X}; f(x) < M_{f_0}\}$ and $\{x \in \mathcal{X}; f(x) = M_{f_0}\}$ has measure zero.

Definition 2.6. *Let $f_0 : \mathcal{X} \to \mathbb{R}$ be a measurable function with a proper median. Then $\mathcal{NFL}(\mathcal{X}, f_0)$ holds iff there exists a measure-preserving random permutation π on $\mathcal{X}$ such that $\mathcal{NFL}(\mathcal{X}, \pi, f_0)$ holds.*

Definition 2.7. *$\mathcal{NFL}(\mathcal{X})$ holds iff $\mathcal{NFL}(\mathcal{X}, f_0)$ holds for all measurable functions $f_0 : \mathcal{X} \to \mathbb{R}$ with proper median.*

When $\mathcal{X}$ is finite, Theorem 2.2 implies that $\mathcal{NFL}(\mathcal{X})$ holds:

Proposition 2.8. *When $\mathcal{X}$ is finite, $\mathcal{NFL}(\mathcal{X})$ holds.*

Proof. Apply Theorem 2.1 to $f_{\mathcal{F}^\Pi_{f_0}}$ defined in Eq. 6. □

We want now to generalize Definitions 2.5, 2.6 and 2.7. In particular we do not want to restrict the distribution of fitness functions to $f_0 \circ \pi$ where π is a measure-preserving random permutation. We therefore have to define a general framework for "random variables", that works also in the case of $\mathcal{X}$ continuous. This is the issue in stochastic processes also termed random fields when $\mathcal{X}$ is multi-dimensional, for instance $\mathbb{R}^n$. A stochastic process f is a mapping $\Omega \times \mathcal{X} : (\omega, x) \mapsto f(x)$ (where, as usual, $f(x)$ implicitly depends on ω). It is termed measurable when it is measurable as a function of both (ω, x). This condition is not restrictive: it is necessary for making $Y(f, m, a)$ meaningful as a random variable. We implicitly assume in the rest of the paper that all stochastic processes are measurable, otherwise all statements are pointless.

We have no more permutation, but we need the weaker assumption that a median of the fitness values is constant. Our definition of a random fitness distribution is as follows:

Definition 2.9. *A random fitness is a measurable map f from Ω to $\mathbb{R}^\mathcal{X}$ such that there exists a constant $M_f \in \mathbb{R}$ which is a proper median of f for any ω.*

Definition 2.9 requires that a proper median exists for any ω, and that it does not depend on ω. Our generalized definition for No-Free-Lunch reads as follows:

Definition 2.10. *Let f be a random fitness. $\mathcal{GNFL}(\mathcal{X}, f)$ holds iff for any $m \in \mathbb{N}$ (smaller than $|\mathcal{X}|$ when $\mathcal{X}$ if finite) and any two optimization algorithms a and b, $Y(f, m, a)$ and $Y(f, m, b)$ follow the same distribution.*

This No-Free-Lunch statement depends on the random fitness distribution f. We would like to characterize domains $\mathcal{X}$ for which $\mathcal{GNFL}(\mathcal{X}, f)$ holds for at least one f:

Definition 2.11. *$\mathcal{GNFL}(\mathcal{X})$ holds if there is one random fitness f such that $\mathcal{GNFL}(\mathcal{X}, f)$ holds.*

Note that we only require the existence of one non-trivial random fitness f such that $\mathcal{GNFL}(\mathcal{X}, f)$ holds which is less restrictive than $\mathcal{NFL}(\mathcal{X})$ requiring that $\mathcal{NFL}(\mathcal{X}, f_0)$ holds for any measurable f_0 with a proper median.

If there exists f_0 such that $\mathcal{NFL}(\mathcal{X}, f_0)$ holds, the distribution

$$f = \{f_0 \circ \pi, \pi \text{ random permutation s.t. } \mathcal{NFL}(\mathcal{X}, f_0, \pi) \text{ holds}\}$$

is such that $\mathcal{GNFL}(\mathcal{X}, f)$ holds (a median of f is a median of f_0 as π is measure-preserving). Therefore:

Proposition 2.12 (Link between NFLs)**.** *For any measure space $(\mathcal{X}, \mathfrak{B}, \mu)$, for any f_0 with a proper median,*

$$\mathcal{NFL}(\mathcal{X}, f_0) \text{ holds } \Rightarrow \exists\, \pi \text{ s.t. } \mathcal{GNFL}(\mathcal{X}, f_0 \circ \pi) \text{ holds}$$
$$\Rightarrow \mathcal{GNFL}(\mathcal{X}) \text{ holds}$$

3 Countable (No)-Free-Lunch

In this section, $\mathcal{X}$ is countably infinite, without loss of generality $\mathcal{X} = \mathbb{N}$. We recall that the measure considered is then the cardinal measure implying that all permutations are measure-preserving. We start by building a non-trivial measurable function f_0 such that $\mathcal{NFL}(\mathcal{X}, f_0)$ does not hold.

Proposition 3.1. *If $\mathcal{X} = \mathbb{N}$ and $f_0(i) = (-1)^{i+1}\, i$, for all $i \in \mathbb{N}$, then there is no random permutation π such that $\mathcal{NFL}(\mathcal{X}, \pi, f_0)$ holds. Therefore, $\mathcal{NFL}(\mathcal{X}, f_0)$ does not hold.*

Proof. First note that f_0 admits proper medians: every $M \in \mathbb{R}$ is a proper median. Assume now that there exists a random permutation π such that $\mathcal{NFL}(\mathcal{X}, \pi, f_0)$ holds. Then, consider, for any $i \in \mathbb{N}$, the algorithm that always chooses $x_1 = i$ as first iterate. The property $\mathcal{NFL}(\mathcal{X}, \pi, f_0)$, i.e., Definition 2.5, applied to this algorithm for any two different values of i, leads to

$$\mathbb{P}(f_0(\pi(i)) = 1) \text{ is the same for all } i,$$

which, thanks to the definition of f_0, leads to

$$\mathbb{P}(\pi(i) = 1) \text{ is the same for all } i.$$

But as the events $\{\pi(i) = 1\}$ for $i \in \mathbb{N}$ are a partition, we have

$$1 = \sum_{i \geq 0} \mathbb{P}(\pi(i) = 1) \ . \tag{11}$$

This yields the expected contradiction as $\mathbb{P}(\pi(i) = 1) = 0$ and $\mathbb{P}(\pi(i) = 1) > 0$ both lead to a contradiction (resp. $\sum_{i \geq 0} \mathbb{P}(\pi(i) = 1) = 0$ and $\sum_{i \geq 0} \mathbb{P}(\pi(i) = 1) = \infty$). □

Note that any injective f_0, i.e., such that $i \neq j \Rightarrow f_0(i) \neq f_0(j)$ would work, provided that a proper median exists. This proposition shows that contrary to finite domain, $\mathcal{NFL}(\mathcal{X})$ does not hold: there are some "histograms of values" (i.e., some f_0) for which some algorithms are better than others, for any measure-preserving random permutation π. However, one can study the existence of distributions of functions for which all algorithms are equivalent. In particular if the objective function is a random function with "enough independence" (see proof below), one does not expect any optimization algorithm to be better than another one. This is what we formalize in the following proposition.

Proposition 3.2 (No-Free-Countable-Lunch)**.** *When $\mathcal{X} = \mathbb{N}$, there exists a non-trivial random fitness f, i.e., a random fitness with minimum different of the maximum with probability one such that $\mathcal{GNFL}(\mathbb{N}, f)$ holds. Moreover, one can choose f such that there exists f_0 satisfying $f = f_0 \circ \pi$ with probability 1, where π is a (necessarily measure-preserving) random permutation of $\mathbb{N}$. Therefore $\mathcal{NFL}(\mathbb{N}, f_0)$ holds.*

Proof. Let f be the random fitness distribution such that the $f(i)$, for $i \in \mathbb{N}$, are independent and uniformly distributed in $\{0, 1\}$. Technically, such a fitness can be built thanks to Kolmogorov's extension theorem.

Then, for any algorithm a, and any m,

$$Y(f, m, a) \text{ is uniformly distributed on } \{0, 1\}^{\{1,\dots,m\}}$$

and therefore $\mathcal{GNFL}(\mathbb{N}, f)$ holds.

Let us now show the second statement of the proposition, i.e., that for some f_0, $\mathcal{NFL}(\mathbb{N}, f_0)$ holds.

With probability one, the subsets of $\mathbb{N}$, $\{f^{-1}(1)\}$ and $\{f^{-1}(0)\}$ are infinite. Consider the deterministic fitness function f_0 defined as $f_0(i) = 1$ if i is even and $f_0(i) = 0$ otherwise. Let π be the random permutation defined for all $m \in \mathbb{N}$ as follows:

- if $f(m) = 1$, then $\pi(m) = 2 \times k(m)$ with $k(m)$ minimal such that $2k(m)$ is different from $\pi(i)$ for any $i < m$
- if $f(m) = 0$, then $\pi(m) = 2 \times k(m) + 1$ with $k(m)$ minimal such that $2k(m) + 1$ is different from $\pi(i)$ for any $i < m$.

Then $\mathcal{NFL}(\mathbb{N}, f_0, \pi)$ holds. □

From this proposition we deduce the following corollary:

Corollary 3.3. *$\mathcal{NFL}(\mathbb{N}, f_0)$ holds with $f_0(x) = 1$ for x even and $f_0(x) = 0$ for x odd.*

We summarize the results of Proposition 2.8, Proposition 3.1 and Proposition 3.3 in the following theorem:

Theorem 3.4 (NFL in discrete spaces)**.** *The following holds:*

- *If $\mathcal{X}$ is finite, then $\mathcal{NFL}(\mathcal{X})$ holds, and therefore $\mathcal{GNFL}(\mathcal{X})$ holds.*
- *If $\mathcal{X}$ is countably infinite:*
  - *$\mathcal{GNFL}(\mathcal{X}, f)$ holds with $f(x) = 1$ with probability $\frac{1}{2}$, independently for each x.*
  - *Therefore, $\mathcal{GNFL}(\mathcal{X})$ holds.*
  - *$\mathcal{NFL}(\mathcal{X}, f_0)$ holds with $f_0(x) = 1$ for x even and $f_0(x) = 0$ for x odd.*
  - *But $\mathcal{NFL}(\mathcal{X})$ does not hold, and for any f_0 with proper median such that $i \neq j \Rightarrow f_0(i) \neq f_0(j)$, $\mathcal{NFL}(\mathcal{X}, f_0)$ does not hold.*

4 Continuous free-lunch

In this section, we show the main result of this paper, namely that there is no random fitness for which all algorithms are equivalent in the sense of Definition 2.10 when $\mathcal{X}$ is a continuous domain. Without loss of generality we assume that $\mathcal{X} = [0, 1]$ and $\mathcal{Y} = \mathbb{R}$.

It is known that the measurability issue in stochastic processes and random fields is non-trivial [11]. Consider $\mathcal{X}$ a continuous domain. One cannot just set a covariance kernel ($\mathcal{X} \times \mathcal{X} \to \mathbb{R}$) on $\mathcal{X}$, marginal laws, and consider "a" random field f with such marginals and covariances. Even with null covariance, one cannot just define marginal laws, and consider "a" random variable with such independent marginals. It is not better with just one distribution of probability on $\mathbb{R}$: one cannot simply define a random field with independent marginals and all marginal distributions equal to the required distribution of probability. It is working with countable domains, as shown by Kolmogorov's extension theorem [9, 31], but it does not work in continuous domains at least for the current formalism of probability theory, based on measurability conditions. The interested reader is referred to [4, 11] for a more detailed discussion, in particular for stochastic processes and random fields.

Theorem 4.1 (Continuous free-lunch)**.** *Let f be a random fitness function with values in $\mathbb{R}^{[0,1]}$. Then $\mathcal{GNFL}([0, 1], f)$ does not hold.*

The proof of this theorem relies on Lemma 4.2 and Lemma 4.5 that are stated and proved below.

Proof of Theorem 4.1: Assume that such an f exists. Then, consider two optimization algorithms for just 1 iterate:

- the algorithm deterministically choosing a given $x \in [0, 1]$;
- the algorithm deterministically choosing a given $x' \in [0, 1]$;

Assuming that $\mathcal{GNFL}([0, 1], f)$ holds leads to the fact that $f(x)$ is distributed as $f(x')$. Therefore, all the $f(x)$ for $x \in [0, 1]$ are identically distributed. Hence, we have shown that in the continuous domain, No-Free-Lunch theorems imply the identical distribution of the fitness-values of each point in the domain.

Define $g(x) = 1$ if $f(x) > M_f$, for some M_f a proper median of f, and $g(x) = 0$ otherwise. g is another random field.

First, $\mathcal{GNFL}([0, 1], f)$ implies that the $g(x)$ are identically distributed for $x \in [0, 1]$.

Now, let us show that the $g(x)$ are independent for $x \in [0, 1]$. Consider some fixed $x_1, \ldots, x_{n-1}, x_n$ in $[0, 1]$ (all different). Then, $(g(x_1), \ldots, g(x_n))$ must be, if $\mathcal{GNFL}([0, 1], f)$ holds, the same as for random search. Therefore, the $g(x_1), \ldots, g(x_n)$ are independent. Besides, for random search, $(g(x_1), \ldots, g(x_n))$ is uniformly distributed on $\{0, 1\}^n$. Therefore, g is a random field, such that $g(x) = 1$ with probability $\frac{1}{2}$ and $g(x) = 0$ otherwise, for any x, and all the $g(x)$ are independent.

We then conclude by Lemma 4.2 that such a g cannot exist. □

Kolmogorov's extension theorem ensures that for any distribution of probability, sequences of independent random variables with the same distribution can be built. Some extensions exist for continuous cases, but without independence. The interested reader is referred to random field theory for this point. We here show that a fully independent and identically distributed family of non-constant random variables indexed by the continuum cannot be defined.

Lemma 4.2 (No continuous family of i.i.d. RV)**.** *Assume that g is a random function with values in $\{0,1\}^{[0,1]}$, such that all the $g(x)$ for $x \in [0,1]$ are identically distributed. Assume that there is $p \in]0,1[$ such that $p = \mathbb{P}(g(x) = 1)$ for all x. Then, the $g(x)$ are not independent.*

We will use in the proof the fact that almost surely g is Lebesgue measurable. This is not a hidden assumption: this is implied by our definition of stochastic processes. Also, this holds as soon as $\int_0^1 g(x)dx$ is well defined almost surely in the realization g.

Proof. Assume that the $g(x)$ are independent. We will now try to get a contradiction. We will split the proof in the following steps:

1. We show (by Lemma 4.5) that almost surely $g^{-1}(1)$ has Lebesgue measure p.
2. We apply Step 1 to show that almost surely, there is an open interval in which $g^{-1}(1)$ has average density $\geq p' > p$.
3. We apply Step 2 to show (by countability) that there is at least one $(a,b) \in \mathbb{Q}^2 \cap [0,1]^2$ with $a < b$ such that $g^{-1}(1)$ has average density $> p'$ in $[a,b]$ with positive probability.
4. We show (by Lemma 4.5) that for any a, b rationals in $[0,1]$, almost surely $g^{-1}(1) \cap [a,b]$ has Lebesgue measure $p \times (b-a)$.
5. The contradiction arises as Step 3 shows that $g^{-1}(1) \cap [a,b]$ has measure $p' \times (b-a)$ and Step 4 shows that $g^{-1}(1) \cap [a,b]$ has measure $p \times (b-a)$.

We now present the detailed steps of the proof.

Step 1. We can apply Lemma 4.5 since all the $g(x)$ are independent, identically distributed with non zero variance (since each $g(x)$ is a Bernoulli random variable with probability $p \in]0,1[$ to take the value 1). Lemma 4.5 ensures that with probability 1, $g^{-1}(1)$ has a Lebesgue measure of p. This concludes the first step.

Step 2. By Lebesgue's density theorem, $E = g^{-1}(1)$ has density 1 at almost every point in E. We recall below the definition of density and Lebesgue's density theorem:

Definition 4.3. *The density of a set $A \subset \mathbb{R}^d$ in an ϵ-neighborhood of a point $x \in \mathbb{R}^d$, with $\epsilon > 0$, is*

$$d_\epsilon(x, A) = \frac{\mu\left(A \cap B(x,\epsilon)\right)}{\mu\left(B(x,\epsilon)\right)}$$

The density of a set A at a point x is

$$d(x, A) = \lim_{\mathcal{E}' \to 0} d_{\epsilon'}(x, A)$$

Theorem 4.4 (Lebesgue's density theorem)**.** *Consider A a measurable set of $\mathbb{R}^d$. Then, almost any $x \in A$ verifies $d(x, A) = 1$.*

Therefore, almost surely there is a rational segment[5] s such that

$$\mu(s \cap E) \geq \left(\frac{1+p}{2}\right) \mu(s) \tag{12}$$

One can extend Eq. 12 for any constant < 1 instead of $(1+p)/2$; $(1+p)/2$ is enough for the rest of this paper, so we here only show Eq. 12.

This concludes step 2.

Step 3. By step 2, and as the set of rational segments is countable, and as almost surely one of them verifies equation 12, there is at least one rational segment which has non-zero probability of verifying equation 12. Consider then $a, b \in \mathbb{Q}^2$ such that $s = [a, b]$ realizes equation 12 with positive probability. This concludes step 3.

Step 4. Consider $g'(x) = g(a+x\times(b-a))$ defined on $[0, 1]$. Apply Lemma 4.5 to g'. Almost surely, by Lemma 4.5

$$g'^{-1}(1) = \{x \in [0, 1]; a + x(b-a) \in E\}$$

has measure p. This implies almost surely

$$g^{-1}(1) \cap [a, b] \text{ has measure } p(b-a). \tag{13}$$

Step 5. Eq. 13 is a contradiction with Eq. 12. □

We now have to deal with the following lemma. This lemma is necessary for our proof, as it is used in order to get a contradiction; as a by-product of this paper, we will see that, in fact, the assumptions in Lemma 4.5 cannot hold.

Lemma 4.5. *Assume that g is a stochastic process with values in $\{0, 1\}^{[0,1]}$. Assume that almost surely g is Lebesgue measurable. Assume that all the $g(x)$ are identically distributed, have non-zero variance and are independent. Define $E = g^{-1}(1)$. Let $p = \mathbb{P}(g(x) = 1)$ that does not depend on x by hypothesis above. Then, with probability 1, $\mu(E) = p$, with μ the Lebesgue-measure.*

Proof. Since g is a stochastic process taking values in $\{0, 1\}^{[0,1]}$, the Lebesgue measure of the pre-image of 1 under g is a random variable that we denote $m = \mu(g^{-1}(1)) = \mu(E)$. We are going to prove that the random variable m is constant almost surely and equal to p (in other words equal to p with probability one).

Let $\mathcal{E}$ be the expected value of $\mu(E)$, i.e., $\mathcal{E} = \mathbb{E}(\mu(E))$ and $\mathcal{V}$ its variance, i.e., $\mathcal{V} = Var(\mu(E))$.

[5] A rational segment is a closed segment included in $[0, 1]$ with rational bounds and non-zero measure.

Step 1. By definition, the three following mappings have the same distribution:

- $x \mapsto g(x)$;
- $x \mapsto g(x/2)$;
- $x \mapsto g((1+x)/2)$.

Step 2: By decomposition of the integral, $\int_0^1 g(x)dx = \int_0^{\frac{1}{2}} g(x)dx + \int_{\frac{1}{2}}^1 g(x)dx$; this shows that $\int_0^1 g(x)dx$ is distributed as $\frac{1}{2}\int_0^1 g(x/2)dx + \frac{1}{2}\int_0^1 g((1+x)/2)dx$. This concludes step 2.

Step 3: $\mathcal{V} = 0$. Step 1 and step 2 together state that $\mu(E)$ is distributed as $(m+m')/2$, with m' an independent copy of m. An important consequence of that is $\mathcal{V} = Var(\mu(E)) = \mathcal{V}/4 + \mathcal{V}/4 = \mathcal{V}/2$, and therefore $\mathcal{V} = 0$.

Therefore, $\mu(E)$ has variance 0.

Step 4: concluding the proof of Lemma 4.5. Consider x a uniform variable in $[0,1]$, independent of g. Using the Fubini Theorem, we have that $\mathbb{P}(g(x) = 1) = p = \mathbb{E}(\mu(E))$. Therefore, $\mu(E)$ is equal to p with probability 1. □

5 The design of optimal optimization algorithms

In the previous sections, we have seen strong limitations to NFL theorems. Only in some very specific cases of fitness distributions, a NFL result holds. We investigate now the following questions: assuming a given distribution of fitness functions, which algorithm is better than the random search and does an optimal optimization algorithm exist? We here provide a positive answer to the question of optimality among all possible optimization algorithms when a priori information on the distribution of the problems is known and when a finite number of function evaluations (FE) is known in advance. Such a question is of drastic importance for real-world problems where the objective function is so expensive that one can afford to spend minutes or hours to choose the next iterate.

This question can be seen as a problem of sequential decision with uncertainty [33], with a delayed reward:

- at each time step, a "decision", which is in the context of optimization a point of the search domain, is proposed,
- a reward is provided at the N^{th} time step (that can be for instance the mean squared distance to the optimum), where N is the number of FE allowed.

For such a problem, Bellman's optimality principle [2] precisely states that an optimal strategy on average exists and provides an explicit derivation of one optimal algorithm.

5.1 Optimal optimization algorithms

We consider a family of fitness functions defined on a domain D and depending on a random parameter θ:

$$f(.,\theta) : D \to \mathbb{R} \ . \tag{14}$$

The domain D can be either discrete, continuous or mixed. For each realization of θ, we assume that the fitness function $f(.,\theta)$ has only one global minimum that we denote $x^*(\theta)$ (this is not necessary for our results, but simplifies notations).

Example 1. *Let $f_s(.,\theta)$ be the family of sphere functions with optimum located in θ, uniformly distributed in $[0,1]^d$, and mapping the continuous d-dimensional hyper-square $[0,1]^d$ into $\mathbb{R}^+$, i.e.,*

$$f_s(x,\theta) = \|x-\theta\|^2 = \sum_{i=1}^{d} (x^i - \theta^i)^2$$

where $x = (x^1, \ldots, x^d)$ and $\theta = (\theta^1, \ldots, \theta^d)$.

We investigate the question of optimal algorithms for a predefined number of iterations $N \in \mathbb{N} \setminus \{0\}$. Let *Opt* be a general optimization algorithm that defines for a given instance of θ, a sequence $(x_n)_{1 \leq n \leq N}$ of points in D and their associated fitness values $(y_n)_{1 \leq n \leq N}$, i.e.,

$$x_1 = Opt() \text{ and } y_1 = f(x_1, \theta)$$

and for $n \in \{1, \ldots, N\}$

$$x_n = Opt(x_1, x_2, x_3, \ldots, x_{n-1}, y_1, \ldots, y_{n-1}) \text{ and } y_n = f(x_n, \theta).$$

We consider optimality in the sense of minimizing a criterion at iteration N. To define this criterion we measure the so-called *loss* at iteration N by mean of a *loss function* denoted $L(x_N, \theta)$. This loss function can be either

$$\begin{aligned} & L(x_N,\theta) = \|x_N - x^\star(\theta)\|^2, \\ \text{or} \quad & L(x_N,\theta) = \|x_N - x^\star(\theta)\|, \\ \text{or} \quad & L(x_N,\theta) = f(x_N,\theta) - f(x^\star(\theta),\theta), \\ & \ldots \end{aligned}$$

For the example of the sphere functions (Example 1) with $d = 2$ and $N = 4$, there exists an (optimal) algorithm for the loss functions defined above - this algorithm will be optimal for the three loss functions above simultaneously. Indeed, let us choose any point $x_1 \in [0,1]^2$ and evaluate the objective function $f(x_1,\theta) = \|x_1 - \theta\|^2$. The optimum θ is located on the circle $\mathcal{C}_1$ of center x_1 and radius $\sqrt{f(x_1,\theta)}$. Let us choose a second point x_2 (different from x_1), θ is located on the circle $\mathcal{C}_2$ of center x_2 and radius $\sqrt{f(x_2,\theta)}$. The intersection between $\mathcal{C}_1$ and $\mathcal{C}_2$ is at most two points. Let x_3 be one of the two points (or

the only one), if $f(x_3, \theta) = 0$, then $\theta = x_3$ and choose $x_4 = x_3$, otherwise choose the other intersection point for x_4. This leads to $f(x_4, \theta) = 0$. In this case we see that for any random parameter θ, the algorithm sketched finds the optimum in four iterations. In general though, the result after N iterations depends on the instance of the parameter θ. Moreover, some optimization algorithms are stochastic and their outcome for a same θ can be different. In order to take into account stochastic optimization algorithms and the random parameter in the fitness function, one defines optimality on average as follows:

Definition 5.1 (Optimality on average)**.** *An algorithm Opt is optimal on average if it minimizes*

$$E_{rs,\theta}\left(L(x_N, \theta)\right) \quad , \tag{15}$$

where rs is the random variable (independent of θ) which is the random-seed of the optimizer (x_N depends on Opt, and Opt implicitly depends on rs, therefore the loss depends on rs).

For any integer $n \in \{0, \ldots, N-1\}$, we define Opt_n as the restriction of Opt to entries of at least n points. This means that $Opt_n(x_1, \ldots, x_j, y_1, \ldots, y_j)$ is defined if and only if $n \leq j < N$. The only difference between Opt and Opt_n is that for Opt_n the n first points (at least) are already given. This difference is only mathematical, not algorithmic.

Choosing the best possible function Opt is exactly a problem of stochastic optimal sequential decisions with discrete time steps and finite horizons. A classical tool for such a problem is called *Bellman principle of optimality* [2, 3]. For $n \in \{1, \ldots, N\}$, the so-called value function V_n is defined almost surely on $D^n \times \mathbb{R}^n$ as follows:

$$V_n = \begin{cases} D^n \times \mathbb{R}^n & \to \mathbb{R} \cup \{\infty\} \\ (x_1, \ldots, x_n, y_1, \ldots, y_n) & \to \inf_{Opt_n} E_{rs,\theta;\forall k \leq n, y_k = f(x_k,\theta)} L(x_N, \theta) \end{cases} \tag{16}$$

where the expectation with respect to θ is conditionally on the event that {for all $k \leq n, y_k = f(x_k, \theta)$}. This is almost surely well-defined in $(y_1, \ldots, y_i)$ for the random variable $(y_1, \ldots, y_n) = (f(x_1, \theta), f(x_2, \theta), \ldots, f(x_n, \theta))$ - then, the conditional distribution is well-defined, assuming that the loss is non-negative. Bellman's principle of optimality states that:

1. The value function V_n can be computed by backward induction as follows:

$$\begin{aligned} V_N(x_1, \ldots, x_N, y_1, \ldots, y_N) =& E_{\theta;\forall i, f(x_i,\theta)=y_i} L(x_N, \theta) \\ V_{n-1}(x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1}) =& \inf_x E_y V_n(x_1, \ldots, x_{n-1}, x, y_1, \ldots, y_{n-1}, y) \end{aligned} \tag{17}$$

   where y is distributed as $f(x, \theta)$, with θ following its probability distribution conditionally on $\forall i \leq n-1, f(x_i, \theta) = y_i$.

2. Any optimizer Opt such that $\forall n \in \{2, \ldots, N\}$,

$$\begin{aligned} &Opt(x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1}) \in \\ &\quad \arg\min_x E_{\theta;\forall i \leq n-1,\ y_i = f(x_i,\theta)} V_n(x_1, \ldots, x_{n-1}, x, y_1, \ldots, y_{n-1}, f(x, \theta)) \end{aligned}$$

minimizes (15).

The optimality is formalized in the following theorem:

Theorem 5.2 (Bayesian optimization)**.** *Consider $N \in \mathbb{N}$. Consider θ a random variable. Assume that almost surely, $f(.,\theta)$ has one and only one minimum $x^*(\theta)$. If we define Opt as:*

$$Opt(x_1,\dots,x_{n-1},y_1,\dots,y_{n-1}) \in$$
$$\arg\min_x E_{y=f(x,\theta)|\forall i\leq n-1, f(x_i,\theta)=y_i} V_n(x_1,\dots,x_{n-1},x,y_1,\dots,y_{n-1},y) \quad (18)$$

with V_n defined by backwards induction as follows:

$$V_N(x_1,\dots,x_N,y_1,\dots,y_N) = E_{\theta;\forall k=1,\dots,N, f(x_k,\theta)=y_k} L(x_N,\theta) \quad (19)$$
$$\text{For all } n=2,\dots,N, V_{n-1}(x_1,\dots,x_{n-1},y_1,\dots,y_{n-1}) =$$
$$\inf_x E_{y=f(x,\theta)|\forall i\leq n-1, f(x_i,\theta)=y_i} V_n(x_1,\dots,x_{n-1},x,y_1,\dots,y_{n-1},y) \quad (20)$$

then Opt minimizes $E_{rs,\theta}\left(L(x_N,\theta)\right)$ among all possible optimizers.

Proof. This result is an immediate application of Bellman's optimality principle [2]. By backward induction, Eq. 17 constructs the functions V_n of Eq. 16. Eq. 18 and Eq. 16 precisely state the optimality of *Opt*. □

Consider Example 1 with $d = 2$, and $N = 4$ and investigate the behavior of the algorithm defined in Theorem 5.2 in that framework. Denote by $S(x_1,x_2,x_3,y_1,y_2,y_3)$ a value of θ such that $\forall i \in \{1,2,3\}, f(x_i,\theta) = y_i$, when such a value exists. By definition of V_4 and of $L(x_N,\theta) = \|x_N - x^*(\theta)\|^2$, $V_4(x_1,x_2,x_3,x_4,y_1,y_2,y_3,y_4) = y_4$. This implies that x_4, chosen by Eq. 18, is x minimizing

$$\begin{aligned} & E_y V_4(x_1,x_2,x_3,x,y_1,y_2,y_3,y) \\ = \;& y \\ = \;& \|x - S(x_1,x_2,x_3,y_1,y_2,y_3)\|^2 \end{aligned} \quad (21)$$

i.e., $x = S(x_1,x_2,x_3,y_1,y_2,y_3)$ (at least almost surely in $(x_1,x_2,x_3,y_1,y_2,y_3)$ - we drop in the sequel detailed references to such almost-sure well-definiteness notions). We can also compute $V_3(x_1,x_2,x_3,y_1,y_2,y_3)$ by Eq. 17:

$$\begin{aligned} & V_3(x_1,x_2,x_3,y_1,y_2,y_3) = \inf_x E_y V_4(x_1,x_2,x_3,x,y_1,y_2,y_3,y) \\ \leq \;& V_4\left(x_1,x_2,x_3,S(x_1,x_2,x_3,y_1,y_2,y_3),y_1,y_2,y_3,\underbrace{f\left(S(x_1,x_2,x_3,y_1,y_2,y_3),\theta\right)}_{=0}\right) \\ = \;& 0\ . \end{aligned}$$

The previous equation implies that, at least when $x_1 \neq x_2$, x_3 can be chosen anywhere in the domain, except in a set of null measure such that the x_i do not uniquely determine θ.

Then, we can compute $V_2(x_1, x_2, y_1, y_2)$ by Eq. 17 again, at least if $x_1 \neq x_2$:

$$\begin{aligned} V_2(x_1, x_2, y_1, y_2) &= \inf_x E_y V_3(x_1, x_2, x, y_1, y_2, y) \\ &\leq V_3(x_1, x_2, x_3, y_1, y_2, y_3) \\ &\leq 0 \end{aligned} \tag{22}$$

with x_3 such that there is only one solution θ to $\forall i \in \{1, 2, 3\}, f(x_i, \theta) = y_i$. Eq. 22 and Eq. 18 show that x_2 can be chosen anywhere in the domain except at x_1 without losing optimality. Also, $V_1(x_1, y_1) = 0$ and x_1 can be chosen anywhere in the domain.

Therefore, the algorithm suggested by Theorem 5.2 in the specific case of the sphere function works as follows with $N = 4$ points in dimension 2:

- choose x_1 in the domain;
- choose $x_2 \neq x_1$;
- choose x_3 such that $S(x_1, x_2, x_3, y_1, y_2, y_3)$ is uniquely determined;
- choose $x_4 = S(x_1, x_2, x_3, y_1, y_2, y_3)$.

This is exactly the analytic algorithm solving the problem.

5.2 Applications

In this section, we discuss a theoretical application to surrogate models; a theoretical application to the possible optimality of estimation-of-distribution principles and experiments about the algorithm suggested by Theorem 5.2, to see if this optimality is purely theoretical or if it can be applied in practice within reasonable computation time.

5.2.1 Application to surrogate models

Meta-models [10, 5, 14] are used in evolutionary algorithms to make them faster, in particular when the objective function is expensive. Under some very mild assumptions on the objective functions, surrogate algorithms converge globally with a 3/2 convergence rate [1]. Moreover surrogate models built without derivatives are superior to local surrogate models built with derivatives for some surprisingly simple and smooth frameworks [34, 6]. Some experiments with very different forms of meta-models can be found in [18, 28, 37], and many applications can be found in [35, 32, 22, 23, 27, 20, 29, 24].

Theorem 5.2 can also be applied to optimization with surrogate models. Assume that a prior distribution on θ is available, as previously assumed. Then, we have proposed above optimal algorithms $Opt(.)$ that provide x_{n+1} as a function of $(x_1, \ldots, x_n, y_1, \ldots, y_n)$. These algorithms only depend on V_n, where V_n is a function mapping sequences of the form $(x_1, \ldots, x_k, y_1, \ldots, y_k)$ with $k \leq N$ to real values.

Let us consider $\mathcal{P} = P(\theta|x_1, \ldots, x_n, y_1, \ldots, y_n)$, the distribution of θ conditionally on $x_1, \ldots, x_n, y_1, \ldots, y_n$. Interestingly, one can see by induction on V_n (see Eq. 19 and Eq. 20) that $V_n(x_1, \ldots, x_n, y_1, \ldots, y_n)$ only depends on $\mathcal{P}$ if $n < N$, and on $(\mathcal{P}, x_N)$ if $n = N$.

This implies that *Opt* can be reformulated as a function of $\mathcal{P}$ only:

$$x_{n+1} = Opt(\mathcal{P}) = Opt(\theta|x_1, \ldots, x_n, y_1, \ldots, y_n) \tag{23}$$

Interestingly, $\mathcal{P}$ is a distribution of fitness functions. This shows that a *distribution* of meta-models is sufficient, and not only *one* meta-model. Interestingly, some implementations, e.g. in [28], use populations of meta-models and not only one meta-model; also, methods like stepwise-uncertainty-reduction models use distributions of models instead of one model [40, 19]. A distribution of models is richer than pointwise uncertainty as used in many surrogate models.

5.2.2 Application to Estimation of Distribution Algorithms (EDAs)

EDAs are intuitively satisfactory for expensive optimization as they iteratively improve a given distribution of probability for the optima [18], in a way very similar to sequential design of experiments. We have studied conditions under which optimal algorithms do exist. We have seen optimal algorithms, that take decisions depending on the distribution of fitness functions conditionally on previously visited points. We here derive a particular case. We define an EDA as an algorithm depending only on the conditional distribution of probability of the optimum:

$$x_{n+1} = Opt(P(x^*(\theta)|x_1, \ldots, x_n, y_1, \ldots, y_n)) \tag{24}$$

and we define the HEDA hypothesis as

$$\text{HEDA: } \forall x, \text{there exists only one } \theta \text{ such that } x^*(\theta) = x.$$

We show that this implies the optimality of EDAs for any prior distribution of probability on fitness functions.

Theorem 5.3 (A sufficient condition for the optimality of EDAs)**.** *Consider a space of fitness-functions of the form $F = \{f(., \theta); \theta \in \Omega\}$, and algorithms as*

$$x_{n+1} = Opt(x_1, \ldots, x_n, y_1, \ldots, y_n), y_i = f(x_i, \theta)^6 \tag{25}$$

Assume that for any θ, $f(., \theta)$ has one and only one minimum at $x^*(\theta)$. Assume that θ follows some distribution of probability. Consider the criterion

$$L(x_N, \theta) = E_\theta \|x_N - x^*(\theta)\|^2. \tag{26}$$

If HEDA holds, then there is an optimal algorithm for criterion 26 of the form given in Eq. 24.

[6] Note that *all* algorithms can be described as in this equation.

Interpretation: If for each x in the domain there is one and only one possible fitness function such that the optimum is at x, then there is an optimal algorithm which is an EDA.

Proof. We consider a space of fitnesses of the following form:

$$F = \{f(.,\theta); \theta \in \Omega\}$$

Assume that for each $\theta \in \Omega$, $f(.,\theta)$ has one and only one minimum $x^*(\theta)$. Thanks to Eq. 23, an optimal strategy for criterion 26 can be designed of the following form:

$$x_{n+1} = Opt\left(P\left(\theta | x_1, \ldots, x_n, y_1, \ldots, y_n\right)\right)$$

Therefore, if there exists a function $k(.)$ such that $\theta = k(x^*(\theta))$ for any θ (i.e., for any x^* if there is only one function in F with minimum at x^*) then the optimal $Opt()$ can be rewritten as follows

$$x_{n+1} = Opt\left(x_1, \ldots, x_n, P\left(k(x^*(\theta)) | x_1, \ldots, x_n, y_1, \ldots, y_n\right)\right)$$

which can be rewritten (within a change of variable in Opt):

$$x_{n+1} = Opt\left(x_1, \ldots, x_n, P\left(x^*(\theta) | x_1, \ldots, x_n, y_1, \ldots, y_n\right)\right)$$

which is an EDA in the sense of Eq. 24. □

5.2.3 UNLEO, a heuristic approximation of an optimal optimization algorithm using Upper Confidence Trees (UCT)

The principle of our approach (Theorem 5.2) is similar to Bayesian inference in machine learning. We here applied bayesian priors in optimization; this approach is inspired by the known optimality (without taking into account any computational cost but only the number of examples) of Bayesian inference for supervised learning, when a prior distribution is available. A main drawback is that Bayesian inference is often very expensive, and involves billiard algorithms, or Monte Carlo Markov Chains, for high dimensional integration. This drawback also holds for our approach: we can only use it when the computational cost of the fitness function is such that we can spend a lot of time in the choice of each iterate. The approach that we propose here is computationally very expensive, but can be implemented thanks to UCT as explained below, and the approach is realistic for expensive optimization problems (problems in which evaluating the objective function requires a long time). Its efficient approximation for non-expensive optimization problems is under work.

Implementation and experimental setup. We perform experiments with the algorithm presented in Theorem 5.2. The comparison-based nature of some optimization algorithms, in particular most evolutionary algorithms, provides

optimal robustness for some criterion [16]; we here adapt the algorithm in Theorem 5.2 to the comparison-based case. This reduces the assumptions underlying the algorithm: we only need a prior on the fitness functions up to a monotonic mapping. The algorithm is as follows:

$$Opt(x_1,\ldots,x_{n-1},y_1,\ldots,y_{n-1}) \in$$
$$\arg\min_x E_{y=f(x,\theta)|\forall i\leq n-1, f(x_i,\theta)=y_i} V_n(x_1,\ldots,x_{n-1},x,y_1,\ldots,y_{n-1},y)$$

with V defined by backwards induction as follows:

$$V_N(x_1,\ldots,x_N,y_1,\ldots,y_N) = E_{\theta;\sigma(f(x_1,\theta),\ldots,f(x_N,\theta)=\sigma(y_1,\ldots,y_N)} L(x_N,\theta)$$

For all $n = 2,\ldots,N$

$$V_{n-1}(x_1,\ldots,x_{n-1},y_1,\ldots,y_{n-1}) =$$
$$\inf_x E_{y=f(x,\theta)|\sigma(f(x_1,\theta),\ldots,f(x_{n-1},y_{n-1}))=\sigma(y_1,\ldots,y_{n-1})} V_n(x_1,\ldots,x_{n-1},x,y_1,\ldots,y_{n-1},y)$$

where $\sigma(a_1,\ldots,a_k)$ is the ranking of $a_1,\ldots,a_k$, i.e., $\sigma(a_1,\ldots,a_k) = (\texttt{sign}(a_i - a_j))_{(i,j)\in\{1,\ldots,k\}^2}$ and $\texttt{sign}(x) = 1$ if $x > 0$, $\texttt{sign}(x) = -1$ if $x < 0$ and $\texttt{sign}(0) = 0$.

The inputs (including parameters) of the optimizer are therefore:

- a random field on the domain;
- a number of iterations,
- a sequence of points $x_1,\ldots,x_n$ in the input domain;
- the ranking of the $f(x_1,\theta),\ldots,f(x_n,\theta)$ (as defined above).

The algorithm outputs x_{n+1}. The two first inputs are parameters of the algorithm (they do not change during an optimization run).

We used the now famous UCT algorithm, which is a bandit-based Monte-Carlo planning algorithm [30], for approximately solving the dynamic programming problem. The family of functions is $x \mapsto f(x,\theta) = \frac{1}{d}\sum_i |x^i - \theta^i|$ (hence $x^*(\theta) = \theta$), for θ uniformly distributed in $[0,1]^d$. The number of fitness evaluations is fixed to $d+4$. We term the resulting algorithm UNLEO (UCT for non-linear expensive optimization). The domain is $[0,1]^d$.

Results. The UNLEO algorithm that we define, using UCT, is optimal only at the limit of a huge computational cost. The comparisons with random search, quasi-random search and the $(1+1)$ Evolution Strategy ($(1+1)$-ES) are here only so that one can see that, with reasonnable computation times, one can have reasonnable results with UNLEO. A detailed comparisons with other techniques is a further work, beyond the scope of this paper.

We present the results in Table 1; all the computation times are per optimization run and averaged over 400 runs. The quasi-random search is performed

by the Halton-sequence. Of course, this is not a fair comparison: UNLEO is computationally far more expensive than other algorithms. We only want here to point out the feasability of the approach. A further work is the comparison with other approaches, like e.g. EGO [26]; this paper and this section only show the feasibility of the UCT approximation of our proved optimal optimization approach.

We see that the time is far from being prohibitive. Preliminar experiments with standard approximate dynamic programming *were* prohibitive; this confirms the known efficiency of UCT for implementing large scale dynamic-programming problems (on this topic, see [41, 17, 15], which have provided the first win of a computer against a professional human in the difficult game of Go).

6 Conclusion

In this paper we have investigated extensions of NFL results for countably infinite and continuous domains and derived applications to the design of optimal optimization algorithms. We have shown that a consequence of NFL theorems for a finite domain $\mathcal{X}$ is that for any algorithms a and b, for any m, the random vectors $Y(f_0 \circ \pi, m, a) = \langle f_0 \circ \pi(x_1), \ldots, f_0 \circ \pi(x_m) \rangle$ and $Y(f_0 \circ \pi, m, b) = \langle f_0 \circ \pi(x_1), \ldots, f_0 \circ \pi(x_m) \rangle$ follow the same distribution for any objective function f_0 and π random permutation uniformly distributed among all permutations over $\mathcal{X}$. We investigate how this property generalizes to countably infinite and continuous domains. For a non-trivial measurable objective function f_0 and π a measure-preserving random permutation, we define $\mathcal{NFL}(\mathcal{X}, \pi, f_0)$ as the fact that for any integer m and any two optimization algorithms a and b, $Y(f_0 \circ \pi, m, a)$ and $Y(f_0 \circ \pi, m, b)$ follow the same distribution. For $\mathcal{X} = \mathbb{N}$ we give non-trivial objective functions f_0 such that it is not possible to find a random permutation π such that $\mathcal{NFL}(\mathbb{N}, \pi, f_0)$ holds. Those objective functions can be chosen such that they admit finite proper median. We also prove that there exists non-trivial f_0 (also with finite proper median) and a random permutation over $\mathbb{N}$ such that $\mathcal{NFL}(\mathbb{N}, \pi, f_0)$ holds. We define a weaker form of NFL, $\mathcal{GNFL}$ that does not restrict the distribution of fitness to the form $f_0 \circ \pi$. For a non-trivial random fitness distribution f, with constant proper median, we define $\mathcal{GNFL}(\mathcal{X}, f)$ as the fact that for any integer m and any two optimization algorithms a and b, $Y(f, m, a)$ and $Y(f, m, b)$ follow the same distribution. Since there exists f_0 with finite proper median and π such that $\mathcal{NFL}(\mathbb{N}, \pi, f_0)$ holds, the distribution $f = f_0 \circ \pi$ is such that $\mathcal{GNFL}(\mathbb{N}, f)$ holds. When $\mathcal{X} = [0, 1]$, we show that it is not possible to find non-trivial random fitness f such that $\mathcal{GNFL}([0, 1], f)$ holds. Our conclusions for NFL can be summarized in Table 2.

Assuming a prior distribution for the fitness distribution and a fixed number of iterates, Bellman's optimality principle allow us to derive optimal optimization algorithms. We define an EDA as an algorithm based on the conditional distribution of the optima. We show that EDAs are optimal if there exists a

Dimension 2 - 6 points	
Algorithm	mean best fitness value
UNLEO (computation time)	
1.55 ms	13.50 ± 0.003
8.025 ms	10.75 ± 0.003
25.975 ms	9.91 ± 0.003
91.85 ms	8.60 ± 0.002
212.85 ms	7.89 ± 0.002
432.975 ms	7.74 ± 0.002
1 s	7.54 ± 0.002
Random search	14.2 ± 0.06
Quasi-random search	11.8 ± 0.05
$(1+1)$-ES	14.8 ± 0.04

Dimension 3 - 7 points	
Algorithm	mean best fitness value
UNLEO (computation time)	
7.3 ms	15.86 ± 0.003
14.05 ms	14.60 ± 0.003
27.05 ms	14.00 ± 0.003
53 ms	13.84 ± 0.003
115.325 ms	12.88 ± 0.003
202.725 ms	11.91 ± 0.003
5s	10.13 ± 0.002
Random search	16.4 ± 0.06
Quasi-random search	14.9 ± 0.05
$(1+1)$-ES	17.2 ± 0.05

Dimension 4 - 8 points	
Algorithm	mean best fitness value
UNLEO (computation time)	
24.725 ms	17.22 ± 0.002
47.1 ms	16.32 ± 0.002
239.35 ms	15.73 ± 0.002
7.5s	12.44 ± 0.002
Random search	18.0 ± 0.05
Quasi-random search	16.7 ± 0.04
$(1+1)$-ES	18.8 ± 0.04

Table 1: Results of UNLEO versus random-search, quasi-random search and the $(1+1)$-ES with one-fifth rule (isotropic, $\sigma_0 = 1/3$, $\sigma_{n+1} = 2\sigma_n$ for succesful mutations and $\sigma_{n+1} = 2^{-1/4}\sigma_n$ otherwise). The random part in the experiments is due to (i) the random choice of the target function and (ii) the random part in the optimization algorithm. We see that the improvement over random-search is much better than the improvement from quasi-random-search versus random-search, in spite of the very small number of iterates. The $(1+1)$-ES does not perform well in this frugal case (only a few iterations). All results are averaged on 400 runs unless otherwise stated. As UCT is an approximate algorithm for stochastic dynamic programming, the results are presented for various values of the computation time; this computation time is controlled by the number of simulations in the Monte-Carlo exploration of the tree in UCT.

Domain $\mathcal{X}$	Finite	Count. inf.	Continu.
$\exists f_0, \mathcal{NFL}(\mathcal{X}, f_0)$ holds	y	y	n
$\forall f_0, \mathcal{NFL}(\mathcal{X}, f_0)$ holds	y	n	n
$\exists f, \mathcal{GNFL}(\mathcal{X}, f)$ holds	y	y	n

Table 2: Conclusions for the NFL.

one-to-one mapping between the fitness-functions and the optima.

In some sense, NFL theorems are true for extremely hard finite cases (e.g. [12, 13, 25]), but they are false for multi-objective optimization [7] and we show in this paper that they are also moderately true in infinite discrete cases and false in continuous spaces. The deep reason for this fact is that in "bigger" spaces (and continuous spaces are "very" big), random fields (and distributions of fitness functions *are* non-trivial random fields in the continuous case) necessarily have correlations [11].

For our analysis, we have kept a finite horizon perspective and investigated non-asymptotic properties by looking at the distribution of $Y(f, m, a)$ for finite values of m. We could reasonably wonder what would be true when considering asymptotic behaviors. We know that many randomized search heuristics–including the random search–asymptotically find the essential minimum of any fitness function, almost surely whereas some non-repeating algorithms (e.g. *any* non-repeating deterministic algorithms) fail for this asymptotic property. Therefore, some algorithms are better than others from the point of view of the asymptotic behavior.

Technically speaking, an interesting fact is that in our proofs, measurability plays a positive role and is not only a technical detail that should be used for the mathematical soundness: the continuous case directly relies on measurability. A possible future direction is to take into account the study of random spaces of fitnesses with some separability conditions [11], where separability could be used to characterize "possible" random-fitness-functions. Hierarchies of optimization algorithms might be defined thanks to a proper formalization of continuous optimization problems as separable random fields.

An interesting point is that the idea of large memory costs, i.e., keeping all the information, is underlying all this paper. Optimal algorithms derived in this paper use all the archive of visited points to decide each new iterate. Optimal meta-model-based algorithms use a distribution of meta-models, and not only one meta-model. It is in accordance with the efficiency of population-based methods, BFGS with large memory, or EDAs with complex-representations of the past information like bayesian-networks, or also the covariance matrix-adaptation memorizing past information [21].

As a consequence, the application to the design of optimal optimization algorithms involves Monte-Carlo planning algorithms, and in particular the famous UCT algorithm, which is based on developing incrementally a large tree representing the possible futures, has a large memory cost.

This paper originates in (i) a theory (the NFL family of results), (ii) an innovative research area (the Monte-Carlo planning algorithms), in the direction of the design of new innovative algorithms, with application to the expensive optimization framework. UNLEO is quite different from all existing algorithms and its main advantages are

- The optimality for a given prior (distribution of problems), a given number of iterates and a criterion quantifying the quality of an optimization run, at least asymptotically in the computation time.
- Very good practical results on artificial benchmarks. These benchmarks are rather preliminary, and further experimental works are required in order to quantify the efficiency of UNLEO, but the experiments in this paper show the feasability of UNLEO, without untractable computational costs.
- Possibility of taking into account particular cases in the design of the algorithm, e.g. we could use specific criteria. For example, we can consider the expected log-fitness, or the expected fitness for increasing the robustness or we can consider a fitness function with noise decreasing with the computational cost (i.e., the fitness value is noisy and the noise depends on the computational effort for reducing the noise as in Monte-Carlo integration) and develop an optimal algorithm for the point of view of the best fitness within some constraint on the overall computational cost instead of the number of iterates.

However, this algorithm is quite complex, is relatively expensive with a cost increasing quickly as the number of iterates increases and has only been tested yet on artificial problems.

Acknowledgments

We thank the anonymous reviewers for their valuable comments and for pointing out the defect in the Proposition 3.1. We also thank Christian Igel and Marc Toussaint for their comments and encouragements.

References

[1] A. Auger, M. Schoenauer, and O. Teytaud. Local and global order 3/2 convergence of a surrogate evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005)*, pages 857–864, New York, 2005. ACM Press.

[2] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[3] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[4] P. Billingsley. *Probability and Measure*. John Wiley and Sons, 1986.

[5] A. Booker, J.E. Jr. Dennis, P. Frank, D. Serafini, V. Torczon, and M. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13, 1999.

[6] A. Conn, K. Scheinberg, and L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming*, 79:397–414, 1997.

[7] D. W. Corne and J. D. Knowles. Some multiobjective optimizers are better than others. In *Proceedings of the 2003 IEEE Congress on Evolutionary Computation (CEC 2003)*, pages 2506–2512. IEEE Press, 2003.

[8] J. C. Culberson. On the futility of blind search: an algorithmic view of "No Free Lunch". *Evolutionary Computation*, 6(2):109–127, 1998.

[9] P. J. Daniell. Integrals in an infinite number of dimensions. *Annals of Mathematics*, 20:281–88, 1919.

[10] J. Dennis and V. Torczon. Managing approximation models in optimization. In N. Alexandrov and M.-Y. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, pages 330–347. SIAM, 1997.

[11] J. Doob. Stochastic process measurability conditions. *Ann. Inst. Fourier, Grenoble*, 25(3 and 4):163–176, 1975.

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

[13] S. Droste, T. Jansen, and I. Wegener. Optimization with randomized search heuristics - The (A)NFL theorem, realistic scenarios, and difficult functions. *Theoretical Computer Science*, 287(1):131–144, 2002.

[14] M. Emmerich, A. Giotis, M. Özdenir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In *Parallel Problem Solving from Nature (PPSN 2002)*, number 2439 in Lecture Notes in Computer Science, pages 371–380. Springer, 2002.

[15] S. Gelly, J. B. Hoock, A. Rimmel, O. Teytaud, and Y. Kalemkarian. The parallelization of monte-carlo planning. In *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO 2008)*, pages 198–203, 2008. To appear.

[16] S. Gelly, S. Ruette, and O. Teytaud. Comparison-based algorithms are robust and randomized algorithms are anytime. *Evolutionary Computation Journal*, 15(4):411–434, 2007.

[17] S. Gelly and D. Silver. Combining online and offline knowledge in UCT. In *Proceedings of the 24th international conference on Machine learning (ICML 2007)*, pages 273–280, New York, NY, USA, 2007. ACM Press.

[18] S. Gelly, O. Teytaud, and C. Gagné. Resource-aware parameterizations of EDA. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC 2006)*, pages 2506–2512. IEEE press, July 2006.

[19] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, 1996.

[20] M. V. Grieken. *Optimisation pour l'apprentissage et apprentissage pour l'optimisation*. PhD thesis, Université Paul Sabatier, Toulouse, 2004.

[21] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[22] D. Hopkins, T. Lavelle, and S. Patnaik. Neural network and regression methods demonstrated in the design optimization of a subsonic aircraft. Technical report, NASA Glen Research Center, Research & Technology, 2002.

[23] M. Husken, Y. Jin, and B. Sendhoff. Structure optimization of neural networks for evolutionary design optimization. *Soft Computing*, 9(1):21–28, 2005.

[24] S. Ibric, M. Jovanovic, Z. Djuric, J. Parojcic, S. Petrovic, L. Solomun, and B. Stupar. Artificial neural networks in the modeling and optimization of aspirin extended release tablets with Eudragit L100 as matrix substance. *AAPS PharmSciTech*, 4(1):62–70, 2003.

[25] C. Igel and M. Toussaint. A No-Free-Lunch Theorem for Non-Uniform Distributions of Target Functions. *Journal of Mathematical Modelling and Algorithms*, 3(4):313–322, 2004.

[26] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[27] A. Keane and P. Nair. *Computational Approaches for Aerospace Design : The Pursuit of Excellence*. John-Wiley and Sons, 2005.

[28] S. Kern, N. Hansen, and P. Koumoutsakos. Local meta-models for optimization using evolution strategies. In *Proceedings of the Parallel Problems Solving from Nature conference (PPSN 2006)*, pages 939–948, 2006.

[29] J. Kleijnen. Sensitivity analysis of simulation experiments: regression analysis and statistical design. *Mathematics and Computers in Simulation*, 34(3-4):297–315, 1992.

[30] L. Kocsis and C. Szepesvari. Bandit-based monte-carlo planning. In *European Conference on Machine Learning (ECML 2006)*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer, 2006.

[31] A. Kolmogorov. *Foundations of the theory of Probability (original: Grundbegriffe der Wahrscheinlichkeitsrechnung)*. Chelsea publishing company, New-York, (1933 original), 1956.

[32] S. J. Leary, A. Bhaskar, and A. J. Keane. A derivative based surrogate model for approximating and optimizing the output of an expensive computer simulation. *Journal of Global Optimization*, 30(1):39–58, 2004.

[33] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML 2006)*, pages 697–704, New York, NY, USA, 2006. ACM.

[34] M. J. D. Powell. Unconstrained minimization algorithms without computation of derivatives. *Bollettino della Unione Matematica Italiana*, 9:60–69, 1974.

[35] Ames Research Center. Aerodynamic design using neural networks, the amount of computation needed to optimize a design is reduced. Technical report, NASA Tech Briefs Online, 2003.

[36] N. Radcliffe and P. Surry. Fundamental limitations on search algorithms: Evolutionary computing in perspective. In J. van Leeuwen, editor, *Computer Science Today*, Lecture Notes in Computer Science, pages 275–291, Berlin, Germany, 1995. Springer Verlag.

[37] T.-P. Runarsson. Ordinal regression in evolutionary computation. In *Proceedings of the Parallel Problems Solving from Nature conference (PPSN 2006)*, pages 1048–1057, 2006.

[38] C. Schumacher, M. D. Vose, and L. D. Whitley. The No Free Lunch and Problem Description Length. In *Genetic and Evolutionary Computation Conference (GECCO 2001)*, pages 565–570, 2001.

[39] E. VanMarcke. *Random Fields: Analysis and Synthesis*. MIT Press, Cambridge MA, 1998.

[40] J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, To appear.

[41] Y. Wang and S. Gelly. Modifications of UCT and sequence-like simulations for Monte-Carlo Go. In *IEEE Symposium on Computational Intelligence and Games, Honolulu, Hawaii*, pages 175–182, 2007.

[42] D. Wolpert and W. Macready. No Free Lunch Theorems for optimization. *IEEE Transactions in Evolutionary Computation*, 1(1):67–82, 1997.