# UNIVERSITY OF DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

Design and Management of Complex Technical Processes and Systems by means of Computational Intelligence Methods


# Optimization with Randomized Search Heuristics: The (A)NFL Theorem, Realistic Scenarios, and Difficult Functions


Stefan Droste Thomas Jansen Ingo Wegener



No. CI-91/00

Technical Report ISSN 1433-3325 August 2000

Secretary of the SFB 531 · University of Dortmund · Dept. of Computer Science/XI
44221 Dortmund · Germany

This work is a product of the Collaborative Research Center 531, "Computational Intelligence", at the University of Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

# Optimization with Randomized Search Heuristics – The (A)NFL Theorem, Realistic Scenarios, and Difficult Functions

Stefan Droste*, Thomas Jansen*, and Ingo Wegener*

FB Informatik, LS 2, Univ. Dortmund, 44221 Dortmund, Germany

{droste,jansen,wegener}@ls2.cs.uni-dortmund.de

**Abstract**

The No Free Lunch (NFL) theorem due to Wolpert and Macready (1997) has led to controversial discussions on the usefulness of randomized search heuristics, in particular, evolutionary algorithms. Here a short and simple proof of the NFL theorem is given to show its elementary character. Moreover, the proof method leads to a generalization of the NFL theorem. Afterwards, realistic complexity theoretical based scenarios for black box optimization are presented and it is argued why NFL theorems are not possible in such situations. However, an Almost No Free Lunch (ANFL) theorem shows that for each function which can be optimized efficiently by a search heuristic there can be constructed many related functions where the same heuristic is bad. As a consequence, search heuristics use some idea how to look for good points and can be successful only for functions "giving the right hints". The consequences of these theoretical considerations for some well-known classes of functions are discussed.

## 1 Introduction

Randomized search heuristics like evolutionary algorithms (Fogel (1995), Goldberg (1989), Schwefel (1995)), simulated annealing (van Laarhoven and Aarts (1987)), and tabu search (Glover and Laguna (1993)) have found many applications. There are fine-tuned variants of these algorithms for problems with a known structure where certain modules are tuned to work for the given situation and there are general variants for so-called black box applications. In this scenario the function $f$ to be maximized (or minimized) is not known and the only way to gain information on $f$ is to ask for the $f$-value of inputs. The $i$th query may depend on the first $i-1$ queries and the corresponding $f$-values. Black box optimization is useful in situations where nobody has the resources and/or capabilities to design a specialized algorithm or where one has to optimize a complex system whose behavior is not well understood and not formally described.

Everybody is aware that specialized algorithms beat general search heuristics for the problems they are designed for. However, many monographs on evolutionary algorithms or other search heuristics (see, e. g. Goldberg (1989)) claim that some randomized search heuristic beats (all) other algorithms on the average "of all problems" (which typically is not formalized).

*This work was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Collaborative Research Center "Computational Intelligence" (531).

Wolpert and Macready (1997) have considered the set $F_{A,B}$ of all functions $f : A \to B$, $A$ and $B$ finite sets, $B$ totally ordered, as a formalization of "all problems" and have proved that in this scenario all search heuristics have the same average behavior. This result has led to many controversial discussions (see, e. g., Culberson (1998), Radcliffe and Surry (1995) or WWW discussions, e. g., `http://www.aic.nrl.navy.mil/~spears/yin-yang.html`). We review the NFL theorem in Section 2 and present a new simple proof not only of the NFL theorem but also a generalized NFL theorem. The simplicity of the proof also implies that "there is no secret or myth" behind the NFL theorem.

Section 3 is devoted to more realistic black box scenarios. Even these very general scenarios allow some advantage for search heuristics. However, one has to admit that most people working on heuristics for black box optimization have much more restricted classes of functions in mind. The considered function should be "simple" and "natural", notions which cannot be formalized. In Section 4, a result called Almost No Free Lunch (ANFL) theorem is proved. It proves that each search heuristic $H$ which is able to optimize some functions efficiently follows some idea about the structure of the considered functions. It is possible to describe other simple functions which are closely related to functions easy for $H$ and which nevertheless are hard for $H$. It is also shown that this approach shows the difficulty of well-known classes of functions. One may argue that the functions discussed in Section 4 are simple to describe but not natural. Therefore, we present in Section 5 a natural and simple function which is difficult for all typically used search heuristics. We finish with some conclusions.

## 2 A generalized NFL Theorem

We formally introduce the original NFL scenario:

**Scenario 1 (No Free Lunch Scenario).** *The sets $A$ and $B$ are finite and $B$ is completely ordered. The class $F_{A,B}$ contains all functions $f : A \to B$ and the aim is maximization. The function $f$ is drawn uniformly from $F_{A,B}$.*

Usually, we expect from optimization algorithms that they stop after having found an optimal point $x$ which implicitly implies that the algorithm has proved $x$ to be optimal. This is not the right aim in black box optimization. In order to know that $x$ is optimal in the NFL scenario, it is sufficient to know that $f(x)$ is the maximal value of $B$. If the $f$-value of optimal points is smaller than the maximum of $B$, it is necessary to see all $f$-values in order to prove that some point is optimal. Since search heuristics use some stopping criterion (implying that they sometimes fail to find an optimum), we measure the resources spent by an algorithm $H$ for the function $f$ by the (expected) number of *different* inputs $a \in A$ such that $f(a)$ has been evaluated until $f$ is evaluated for an optimal input $a^*$. Many popular search heuristics evaluate certain points more than once but this can be avoided by using a dictionary of the inputs and their $f$-values evaluated so far.

We have announced to prove a generalization of the NFL theorem, which not only holds in the NFL scenario. A set $F \subseteq F_{A,B}$ is called *closed under permutations* if it contains with $f$ also all $f_\pi$ where $\pi$ is a permutation on $A$ and

$$f_\pi(a) := f(\pi(a)).$$

Obviously, $F_{A,B}$ is closed under permutations.

**Theorem 1 (Generalized NFL Theorem).** *Let $H$ be an arbitrary (randomized or deterministic) search heuristic for functions $f \in F \subseteq F_{A,B}$ where $F$ is closed under permutations. Let $r(H)$ be the average (under the uniform distribution on $F$) of the expected runtimes of $H$ on $F$. Then $r(H)$ is a value independent of $H$, i. e., $r(H)$ is the same for all $H$.*

**Proof.** Let $a \in A$ and $B'_a = \{b \in B \mid f(a) = b \text{ for some } f \in F\}$. For $a \in A$ and $b \in B'_a$ let $F_{a,b}$ be the set of functions $f' : A - \{a\} \to B$ such that the extension $f : A \to B$ defined by $f(a) = b$ and $f(a') = f'(a')$ for $a' \in A - \{a\}$ belongs to $F$. The essential properties are the following ones:

- $F_{a,b}$ is closed under permutations,
- $F_{a,b}$ and $F_{a',b}$ are isomorphic for $a \neq a'$, i. e., if $f \in F_{a,b}$ then $f'$, defined by $f'(a) := f(a')$ and $f'(a'') := f(a'')$ for all $a'' \in A - \{a, a'\}$, belongs to $F_{a',b}$.

We prove these two claims:

- Let $f \in F_{a,b}$ and $f'$ its extension defined by $f'(a) = b$. Let $\pi$ be a permutation on $A - \{a\}$ and $\pi'$ its extension on $A$ defined by $\pi'(a) = a$. Since $F$ is closed under permutations, $f'_{\pi'} \in F$ and $f'_{\pi'}(a) = b$. Hence, $f_\pi \in F_{a,b}$.
- Let $f \in F_{a,b}$ and $f^*$ its extension defined by $f^*(a) = b$. Then $f^* \in F$. Let $f'$ be defined by $f'(a) = f(a')$ and $f'(a'') = f(a'')$ for all $a'' \in A - \{a, a'\}$. Let $f^{**}$ be the extension of $f'$ defined by $f^{**}(a') = b$. Then $f^{**} = f^*_\pi$ for the transposition $\pi$ interchanging $a$ and $a'$, i. e., $\pi(a) = a'$, $\pi(a') = a$, and $\pi(a'') = a''$, otherwise. Moreover, $f^{**} \in F$, since $F$ is closed under permutations. This implies that $f' \in F_{a',b}$.

First, we prove the theorem for deterministic search heuristics. This is done by induction on $|A|$. The claim is obvious for $|A| = 1$. For $|A| > 1$, let $H$ and $H'$ be deterministic search strategies whose first search points are $a$ and $a'$, respectively. Since $F$ is closed under permutations, the number of functions $f \in F$ where $a$ is optimal is equal to the number of functions $f \in F$ where $a'$ is optimal. If $f(a) = b$, strategy $H$ is faced with $F_{a,b}$. If $f(a') = b$, strategy $H'$ is faced with $F_{a',b}$. The second claim shows that these are isomorphic problems and the first claim shows that these problems are handled by induction hypothesis. Hence, the strategies $H$ and $H'$ have the same average cost for all possible results $b$ of the evaluation of the first search point. This implies the result for deterministic search heuristics.

Now it is easy to generalize the result to randomized search strategies. The number of different deterministic search strategies is finite. Let $m$ be its number. A randomized search strategy is a probability distribution $p = (p_1, \ldots, p_m)$ and chooses the $i$-th deterministic strategy with probability $p_i$. We are considering a two persons game where our opponent chooses $f \in F$ and we choose the search strategy. Our aim is to minimize the expected cost of the strategy. However, the strategy of the opponent is fixed to the uniform distribution of all $f \in F$. Then it is well-known (see, e. g., Motwani and Raghavan (1995)) that the expected cost of a randomized search heuristic is the weighted average of the cost of the deterministic search heuristics. Since all deterministic search heuristics have the same cost, this also holds for all randomized search heuristics. □

The generalized NFL theorem is by no means surprising. If a class of functions does not change by any permutation on the input space, there is no structure which can be used for search. Hence, all search strategies show the same behavior.

# 3 More Realistic Scenarios for Black Box Optimization

The classical NFL theorem holds in the NFL scenario and this is a scenario which seems to be a reasonable formalization of the scenario containing "all" functions. However, this scenario is not realistic at all. The class of functions $F_{A,B}$ contains $|B|^{|A|}$ functions. The situation of $A = \{0,1\}^{100}$ is not unusual. Assuming, e. g., that $B$ contains the set of 20-bit integers, the number of functions equals $2^{20 \cdot 2^{100}}$ and, for each coding, only a tiny fraction of all functions can be described or evaluated with available resources. Search heuristics are meaningless if it is too expensive to evaluate the given function.

Before we discuss more realistic scenarios for black box optimization, we mention two scenarios which should not be mixed up with black box optimization. We still assume that $A$ and $B$ are finite and $B$ is completely ordered.

**Scenario 2 (One Shot Scenario).** *One function $f : A \to B$ has to be maximized.*

This is the real life situation. However, in black box optimization we do not know $f$. Theory in the one shot scenario is not possible, since there is an algorithm with runtime 1 (the difficulty is to find this algorithm). Moreover, we are never designing algorithms for the one shot scenario but for functions of a given type.

**Scenario 3 (Fixed Function Type Scenario).** *It is known that $f$ is chosen from some class of functions sharing some properties.*

This is the typical situation for the design of efficient algorithms if each function is some instance of the same general problem like the traveling salesperson problem or maximal matching. If the functions share some structural property like separability, unimodality, or being a polynomial of small degree, there are special optimization techniques and also special search heuristics (see, e. g., Papadimitriou and Steiglitz (1998) or Hochbaum (1996)).

However, this is more knowledge than we can expect to have in black box optimization.

**Scenario 4 (Restricted Black Box Optimization).** *The scenario is the same as the NFL scenario with the only exception that $F_{A,B}$ is replaced by some subset $F'_{A,B}$ of functions whose complexity (in a sense to be specified) is restricted.*

In order to discuss the NFL theorem we start the discussion with restrictions which necessarily are fulfilled in real life situations.

**Scenario 4.1 (Time Restricted Black Box Optimization).** *The restriction is given by a time bound $t$ on the number of steps to evaluate $f$.*

**Scenario 4.2 (Size Restricted Black Box Optimization).** *The restriction is given by a bound $s$ on the size of a representation of $f$, e. g., by a circuit.*

**Scenario 4.3 (Kolmogoroff Complexity Restricted Black Box Optimization).** *The restriction is given by a bound $b$ on the Kolmogoroff complexity of $f$.*

Search heuristics can be successful only if the considered function is easy to evaluate. The function has to be evaluated at many points. Hence, the evaluation time has to be small. Remember that most of the famous NP-equivalent optimization problems have objective functions which can be evaluated in linear time. The other way of thinking is a hardware oriented one, a hardware description should be small and the Kolmogoroff complexity oriented point of view (Li and Vitányi (1993)) is an even more robust one.

Our conjecture is that no such scenario where the restriction is not a trivial one allows an NFL theorem. The problem with a proof is that nobody is able to discuss the class of functions with bounds like $t(n) = n$, $s(n) = n$, $c(n) = n$, or even $c(n) = O(\log(n))$ (Kolmogoroff complexity) for functions on $n$ Boolean variables. There are no average case results known for such classes of functions. The only possibility is to consider very small classes of functions where a complete case inspection is possible. Droste, Jansen, and Wegener (1999) have performed such considerations for different complexity measures. For all non-trivial situations considered they have proved that there is no NFL theorem. However, the advantages of the better algorithms are small (also because of the small sets $A$ and $B$). This has led to the claim that there is at least a "free appetizer" in restricted black box optimization.

# 4 An ANFL Theorem

Knowing a restriction on the complexity of the considered function breaks the total symmetry of the input space. This allows a free appetizer, i. e., algorithms using the given information in a better way are better than other ones. However, knowing that a function can be evaluated in linear or quadratic time gives not much information to direct the search. There are functions which are very easy to evaluate but hard for black box optimization. We later discuss such functions, among them functions known as "needle in a haystack". Moreover, the fitness functions of many hard optimization problems are easy to evaluate. Hence, we conjecture that restricted black box optimization does not allow much more than a free appetizer. Such conjectures are hard to formalize and general "free lunch" theorems in restricted black box optimization cannot be proved with nowadays available methods. Therefore, we have to be satisfied with less ambitious results.

Let $H$ be a randomized strategy which is efficient for some function $f : \{0,1\}^n \rightarrow \{0, 1, \dots, N-1\}$. The expected runtime is bounded by a (small) polynomial and/or the probability that $H$ finds an optimal input for $f$ within a small number of steps is very close (exponentially close) to 1. We describe a set of functions $f^* : \{0,1\}^n \rightarrow \{0, 1, \dots, N\}$ such that the success probability of $H$ on $f^*$ is exponentially small even for some exponentially increasing bound on the runtime. The number of these functions is increasing double exponentially. Finally, it is proved that the complexity of exponentially many of these functions is only by a small amount larger than the complexity of $f$.

Randomized search heuristics do not recognize whether they have found a solution of the optimization problem. They use some external stopping criterion. Hence, we may consider $H$ without a stopping criterion and, therefore, $H$ is running forever. We investigate the first $2^{n/3}$ steps of $H$ working on $f$. For $a \in \{0,1\}^n$ let $q(a)$ be the probability that $H$ evaluates $f(a)$ during its first $2^{n/3}$ steps. This time restriction implies

$$\sum_{a \in \{0,1\}^n} q(a) \le 2^{n/3}.$$

We partition $\{0,1\}^n$ into the $2^{2n/3}$ disjoint subspaces $S_b$, $b \in \{0,1\}^{2n/3}$, where $S_b$ contains all $a \in \{0,1\}^n$ such that $a_i = b_i$, $1 \le i \le 2n/3$. Let $q^*(b)$ be the probability that $H$ evaluates at least one $f(a)$, $a \in S_b$, during its first $2^{n/3}$ steps. Since

$$q^*(b) \le \sum_{a \in S_b} q(a),$$

the pigeonhole principle implies the existence of some $b^*$ such that

$$q^*(b^*) \le \frac{2^{n/3}}{2^{2n/3}} = 2^{-n/3}.$$

We even may conclude that for a fraction $\varepsilon(n)$ of all $b$ the inequality $q^*(b) \le (1-\varepsilon(n))^{-1} \cdot 2^{-n/3}$ holds (Markoff inequality).

Let $f^*$ be one of the at least $N^{2^{n/3}-1}$ (and, therefore, double exponentially many) functions $f^*$ defined in the following way. Let $f^*(a) := f(a)$ if $a \notin S_{b^*}$ and the only restriction for $f^*$ on $S_{b^*}$ is the existence of some $a^* \in S_{b^*}$ such that $f^*(a^*) = N$. A search heuristic can distinguish two functions $f$ and $f^*$ only by evaluating $f$ and $f^*$ for some $a$ where $f(a) \ne f^*(a)$. Before such a point of time $H$ works on $f$ as on $f^*$. This implies for the search heuristic $H$ the following.

With a probability of at least $1 - 2^{-n/3}$ the search heuristic $H$ faced with $f^*$ does not evaluate some point $a \in S_{b^*}$ during its first $2^{n/3}$ steps. Hence, its success probability is bounded above by $2^{-n/3}$. This also implies that it will not gain much from multistart options.

Most of the functions $f^*$ are hard to evaluate and need large representation size and long descriptions. This already follows by counting arguments, since the number of functions $f^*$ is double exponentially increasing. In order to describe $f^*$ it is sufficient to describe $f$, $b^*$, and the function $f^*$ restricted to $S_{b^*}$. For the function $f^*$ restricted to $S_{b^*}$ we now allow only choices of functions which are easy with respect to evaluation time, representation size, and Kolmogoroff complexity. There are exponentially many functions $f^*$ which only take zeros on $S_{b^*}$ with the exception of $a^* \in S_{b^*}$ where the fitness value equals $N$. There are also exponentially many functions on $S_{b^*}$ where the fitness value is $N$ minus the Hamming distance to the chosen $a^* \in S_{b^*}$. These functions are easy to evaluate: we only need to decide whether $a \in S_{b^*}$ or not and then we evaluate $f$ if $a \notin S_{b^*}$ or $f^*$ if $a \in S_b$. Circuits and other representation types may realize such a case inspection and the Kolmogoroff complexity grows at most by an additive term of $O(n)$ for the description of $b^*$, $a^*$, and some extra information.

**Theorem 2 (Almost No Free Lunch (ANFL) Theorem).** *Let $H$ be a randomized search strategy and $f : \{0,1\}^n \to \{0,1,\dots,N-1\}$. Then there exist at least $N^{2^{n/3}-1}$ functions $f^* : \{0,1\}^n \to \{0,1,\dots,N\}$ which agree with $f$ on all but at most $2^{n/3}$ inputs such that $H$ does find the optimum of $f^*$ within $2^{n/3}$ steps with a probability bounded above by $2^{-n/3}$. Exponentially many of these functions have the additional property that their evaluation time, circuit size representation, and Kolmogoroff compexity is only by an additive term of $O(n)$ larger than the corresponding complexity of $f$.*

It is obvious that we get better bounds on the success probability if we decrease the number of available steps and vice versa. However, in order to ensure that different functions $f$ and $g$ lead to different functions $f^*$ and $g^*$ which are pairwise unequal we have to consider functions $f$ and $g$ which differ for at least $2 \cdot 2^{n/3} + 1$ inputs.

The ANFL theorem implies that a search strategy has to pay for its success for some functions $f$ with its bad behavior on many functions which are not much more complex than $f$. Hence, each search strategy bears some intuition in mind how functions for optimization look like. One such assumption is that inputs $a$ with large $f(a)$ are most likely close to other inputs with large $f$-values.

We discuss some simple applications of the ANFL theorem. The constant function $f_n(a) = 0$, $a \in \{0,1\}^n$, is simple for each search heuristic. The functions $\text{HAY}_{n,b} : \{0,1\}^n \to \{0,1\}$ defined by $\text{HAY}_{n,b}(b) = 1$ and $\text{HAY}_{n,b}(a) = 0$ if $a \neq b$ are known as "needle in a haystack" functions and are assumed to be difficult for all search heuristics. The proof of the ANFL theorem implies that each search heuristic fails on most of the haystack functions. A search strategy which does not start searching at quite random points can be efficient for some HAY functions.

Everybody expects that a search heuristic efficiently finds the optimum of $\text{ONEMAX}_n(a) = ||a||_1 := a_1 + \cdots + a_n$. This has been proved for all typically used search heuristics. We expect that $\text{ONEMAX}_n$ gives enough hints to increase the number of ones. The function $\text{TRAP}_n$ differs from $\text{ONEMAX}_n$ only on the all zero string $0^n$ which is optimal for $\text{TRAP}_n$, since $\text{TRAP}_n(0^n) := n + 1$. This function is claimed to be difficult for search heuristics. However, this statement works in a one shot scenario and has to be wrong. A search heuristic may start by evaluating the function $0^n$. Whitley (1997) has discussed for each search heuristic $H$ the variant $H^*$ defined by the following rule. If $H$ evaluates $f$ on $a$, then $H^*$ does the same for $a$ and its bitwise complement $\overline{a}$. If $H$ is efficient on $\text{ONEMAX}_n$ then $H^*$ is efficient (losing at most a factor of 2) on $\text{TRAP}_n$. It is more meaningful to consider the class of trap functions $\text{TRAP}_{n,b}$ where $\text{TRAP}_{n,b}(b) = n + 1$ and $\text{TRAP}_{n,b}(a) = \text{ONEMAX}_n(a)$ if $a \neq b$. The proof of the ANFL theorem implies that each search heuristic fails on most of the trap functions. This proves that Whitley's idea only helps in very special situations.

For certain classes of search heuristics the same conclusions have been proved without the ANFL theorem which nevertheless makes the arguments clearer and which treats the more general situation.

# 5 A Non-Artificial and Simple Function Which Is Hard for Simulated Annealing and Evolutionary Algorithms

Functions like the needle in a haystack or the trap functions have short descriptions and are easy to evaluate. However, we do not expect to be faced with such functions in real life optimization. These functions are artificial. The interesting property of these two classes of functions is that for each search heuristic there are many of them which are hard to maximize. In other words, for each search heuristic many of these functions are giving misleading or no hints. There is no function which is misleading for all search heuristics. Nevertheless, we claim that each needle in the haystack function and each trap function $\textsc{Trap}_{n,b}$ where $||b||_1 \le (1-\varepsilon)n$ for some fixed $\varepsilon > 0$ is misleading for all frequently used search heuristics. Indeed, it is not difficult to prove this for a long list of search heuristics.

However, we are more interested in a non-artificial and simple function with such properties. Later we present a function which is a polynomial of degree 3 with a short description, which is an instance of one of the best-known maximization problems (implying, that it is non artificial), and which is claimed to be difficult for all frequently used search heuristics. We shall prove this conjecture for simulated annealing and evolution strategies.

Before we present the special function we investigate the behavior of these search heuristics on a class of functions which should be easy for all reasonable search heuristics. These considerations can be applied later to show the bad behavior on our example function.

A function $f : \{0,1\}^n \to \mathbb{N}$ is called symmetric if $f(a)$ depends on $a$ only via $||a||_1$, the number of ones in $a$. A symmetric function is called decreasing if $||a||_1 < ||b||_1$ implies $f(a) > f(b)$. We expect that a reasonable search strategy quickly finds the only optimal input $0^n$ for such functions.

**Hypothesis 1.** *For each reasonable search heuristic $H$, each $\varepsilon > 0$, and each sequence $f = (f_n)$ of symmetric decreasing functions there are some $\alpha, \beta > 0$ such that the probability that $H$ tests among the first $\exp(o(n^\alpha))$ search points one point $a$ where $||a||_1 \ge (1/2+\varepsilon)n$ is bounded above by $\exp(-\Omega(n^\beta))$.*

This rather technical hypothesis has a simple informal description. A symmetric decreasing function gives only hints to look for individuals with a small number of ones. By Chernoff's inequality (Hagerup and Rüb (1989)) the fraction of search points $a$ where $||a||_1 \ge (1/2+\varepsilon)n$ is bounded above by $\exp(-\varepsilon^2 n/3)$. Hence, random search running for $\exp(\varepsilon^2 n/6)$ steps has a probability of at most $\exp(-\varepsilon^2 n/6)$ to test such a search point. "Reasonable" search heuristics should have even a smaller chance of looking for such points, since all hints lead into the other direction. We do not define the term *reasonable*. We think of all search heuristics which have no a priori preference of search regions, which prefer to base their search more on evaluated search points with a high $f$-value (fitness based selection), and which prefer to look at nearer (Hamming) neighbors. Since in this general setting, the claim can only be falsified, we have not called it conjecture but hypothesis. However, the hypothesis can be proved for specific search heuristics. This is very easy for simulated annealing and more difficult for evolution strategies.

**Theorem 3.** *The hypothesis for search heuristics on symmetric decreasing functions holds for simulated annealing for the parameters $\alpha = 1$ and $\beta = 1$.*

**Proof:** Simulated annealing starts with a randomly chosen input $a$. The probability that $||a||_1 \geq (1/2 + \varepsilon/2)n$ is bounded above by $\exp(-\varepsilon^2 n/12)$. Then simulated annealing chooses a random Hamming neighbor $a'$ of $a$ and accepts it with probability 1 if $||a'||_1 < ||a||_1$ and accepts it with some probability $p_f(a, a')$ if $||a'||_1 > ||a||_1$. The chance of reaching a search point with at least $(1/2 + \varepsilon)n$ ones is maximized for $p_f(a, a') = 1$. Since we start with at most $||a||_1 \leq (1/2 + \varepsilon/2)n$ ones in order to reach some $b$ where $||b||_1 \geq (1/2 + \varepsilon)n$, there has to be a time period $t$ where we start with a point $a'$ with $(1/2+\varepsilon/2)n$ ones, end with a point $b'$ with $(1/2 + \varepsilon)n$ ones and find in between only points with more than $(1/2 + \varepsilon/2)n$ ones. For such points the probability that a random neighbour has a one more is at most $1/2 - \varepsilon/2$. Hence, we overestimate the probability to reach the level of points with $(1/2 + \varepsilon)n$ ones if we assume that we start at level $(1/2 + \varepsilon/2)n$ and that the probability of increasing the number of ones in one step equals $1/2 - \varepsilon/2$.

We consider a time interval of length $t$ and Bernoulli trials with success probability $1/2-\varepsilon/2$. We are interested in the probability of at least $t/2+\varepsilon n/4$ successes (implying at most $t/2-\varepsilon n/4$ missuccesses). This event is necessary and sufficient to reach the level with $(1/2+\varepsilon)n$ ones from the level with $(1/2+\varepsilon/2)n$ ones. The expected number of successes equals $t/2 - \varepsilon t/2$. If $t \leq \varepsilon n/4$, it is impossible to have at least $t/2 + \varepsilon n/4$ successes. If $t > \varepsilon n/4$, the probability of at least $t/2$ successes can be bounded by Chernoff's bound by $\exp(-\Omega(t)) \leq \exp(-\Omega(n))$.

For a time bound $T = \exp(o(n))$, we have at most $T^2 = \exp(o(n))$ time intervals and the success probability for each interval is $\exp(-\Omega(n))$. Hence, the total success probability is still $\exp(-\Omega(n))$. $\square$

In order to investigate evolution strategies, we have to describe this class of search strategies. An evolution strategy works with populations of some fixed polynomial size whose members are initialized randomly and independently. New individuals are created by mutation from old individuals. Mutation is driven by a probability $p$. If the individual $x$ is chosen for mutation, each bit is flipped independently with probability $p$. If $y$ differs from $x$ at $d$ positions, the probability that $y$ is obtained from $x$ by mutation equals $p^d \cdot (1-p)^{n-d}$. The idea of mutation is to produce randomly small changes. Hence, $p \leq 1/2$ (usually $p$ is much smaller than $1/2$).

Selection is the possibly randomized process to determine the members of the next generation. Let $y_1^{old}, \ldots, y_m^{old}$ be the members of the old generation and let $z_1, \ldots, z_k$ be the children produced from $y_1^{old}, \ldots, y_m^{old}$ by mutation. The selection process is allowed to depend on these individuals only via their fitness values and the property whether the individual is a child or a parent. The main property of selection is that the chance of individuals to be chosen is positively correlated with the fitness. More precisely, if $f(x) \geq f(x')$ and either $x$ and $x'$ are children or $x$ and $x'$ are parents, the individual $x$ has at least the same chance as $x'$ to be chosen. Often the same is true if $x$ is a child and $x'$ is a parent. There may be rules to prevent duplicates.

Selection is also the process to choose individuals for mutation. This is done in the same way as described above with the only exception that only the members of the last generation are available.

**Theorem 4.** *The hypothesis for search heuristics on symmetric decreasing functions holds for all evolution strategies for $\alpha = \beta = 1/2$.*

**Proof:** We fix an evolution strategy by choosing the population size $S = S(n)$, the mutation probability $p = p(n)$, and the selection scheme. For a point of time $t = t(n)$ we ask for the "success" probability $p^* = p^*(n)$, namely the probability that an individual with at least $(1/2+\varepsilon)n$ ones has been produced. Since we are interested in (small) upper bounds on $p^*$, we may change the Markoff process describing the behavior of the evolution strategy in such a way that the success probability increases. The idea is to obtain a Markoff process which is easier to handle.

Selection is only based on the fitness of the individuals (and the property whether an individual is a parent or a child) and mutation works on single individuals. Moreover, the fitness function is symmetric. Hence, we can replace each individual with $s$ ones by the string $0^{n-s}1^s$ without influencing the success probability.

It is easy to analyze the initialization step. By Chernoff's bound the probability that a random individual (as created in the initialization phase) has at least $(1/2 + \varepsilon/2)n$ ones is bounded above by $\exp(-\varepsilon^2 n/12)$. Since $S(n)$ is polynomially bounded, the probability that at least one individual of the first generation has at least $(1/2+\varepsilon/2)n$ ones is bounded above by $\exp(-\Omega(\varepsilon^2 n))$. If some individual with at least $(1/2+\varepsilon/2)n$ ones is produced in the initialization phase, we consider this as a success of the algorithm. Hence, we assume in the following that no individual of the first generation has at least $(1/2 + \varepsilon/2)n$ ones.

Informally, we believe that the individual $I = 0^{n-s}1^s$ is better for our optimization task than $I' = 0^{n-s'}1^{s'}$, if $s > s'$. Formally, we prove that for $I$ it is at least as likely to obtain by mutation a string with at least $s''$ ones as for $I'$. For this reason we compare $I$ and $I'$:

$$\begin{aligned} I &= \boxed{0 \cdots 0}\boxed{1 \cdots 1}\boxed{1 \cdots 1} \\ I' &= \underbrace{\boxed{0 \cdots 0}}_{n-s}\underbrace{\boxed{0 \cdots 0}}_{s-s'}\underbrace{\boxed{1 \cdots 1}}_{s'} \end{aligned}$$

Mutation works in the same way on the first $n - s$ bits and the last $s'$ bits. Independently from this, mutation flips each of the $s - s' > 0$ bits in the middle part independently with probability $p$. Since $p \leq 1/2$ (this assumption is essential here), the probability of flipping at most $d$ bits is at least as large as flipping at least $(s - s') - d$ bits.

The fitness function is decreasing with the number of ones. Then fitness-based selection only can prefer individuals with less ones. By the statement above, we conclude that we can assume without loss of generality that selection does not depend on the fitness of the individuals.

By our statement on mutation, we only increase the success probability by replacing each individual with less than $(1/2+\varepsilon/2)n$ ones by an individual with exactly $(1/2 + \varepsilon/2)n$ ones.

In the last step, we consider the situation that the algorithm produces an individual $I^*$ with at least $(1/2+\varepsilon)n$ ones. This individual has a history (such a history-based approach has been used for the first time by Rabani, Rabinovich, and Sinclair (1995)), i. e., there is a sequence $I_0, I_1, \ldots, I^*$ of individuals such that $I_0$ belongs to the initial population and $I_{i+1}$ is produced from $I_i$ by mutation. By assumption $||I_0||_1 = (1/2+\varepsilon/2)n$, $||I_i||_1 \geq (1/2+\varepsilon/2)n$, and $||I^*||_1 \geq (1/2+\varepsilon)n$. We consider the subsequence starting with the last individual with exactly $(1/2+\varepsilon/2)n$ ones. This sequence is denoted (after renumbering) by $I_0, I_1, \ldots, I_{t^*} = I^*$ where $||I_0||_1 = (1/2+\varepsilon/2)n$, $||I_i||_1 > (1/2+\varepsilon/2)n$ for $i > 0$, and $||I_{t^*}||_1 \geq (1/2+\varepsilon)n$. Because of the second property individual $I_i$ is produced by mutation from $I_{i-1}$ and not by mutation followed by a replacement as described above.

The strings $I_0, I_1, \ldots, I_{t^*-1}$ altogether contain at least $(1/2+\varepsilon/2)nt^*$ ones and at most $(1/2-\varepsilon/2)nt^*$ zeros. We like to estimate the probability that starting with $I_0$ we get an individual $I_{t^*}$ which is a success. All single bits of all $I_i$, $i < t^*$, have a chance to be mutated. The mutation probability is $p$. It is a necessary condition that altogether at least $n\varepsilon/2$ more bits are flipping from 0 to 1 than bits are flipping from 1 to 0.

For such a success, it is necessary that at most $nt^*p/2$ ones flip or that at least $nt^*p/2$ zeros flip. For constant $p > 0$, we can estimate the probability of both events (by Chernoff's bounds) by $\exp(-\Omega(nt^*p))$. If $p = \Omega((t^*n^{1/2})^{-1})$, this probability is exponentially small. If $p = O((t^*n^{1/2})^{-1})$, $nt^*p/2 = O(n^{1/2})$. In this case, we use the fact that at least $\varepsilon n/2$ zeros have to flip. Again, by Chernoff's bounds, this probability is bounded by $\exp(-\Omega(n^{1/2}))$. If the algorithm produces $\exp(o(n^{1/2}))$ individuals, the success probability still is bounded by $\exp(-\Omega(n^{1/2}))$. □

**Corollary 1.** *Let $f$ be a function which equals a symmetric decreasing function on all inputs $a$ where $||a||_1 < (1/2+\varepsilon)n$ and which has the property that $||b||_1 \geq (1/2+\varepsilon)n$ for all optimal points $b$. Then the probability that simulated annealing or an evolution strategy finds the optimum of $f$ within $\exp(o(n^{1/2}))$ steps is bounded above by $\exp(-\Omega(n^{1/2}))$.*

**Proof** As long as a search heuristic does not evaluate the considered function on an input with at least $(1/2+\varepsilon)n$ ones, it cannot distinguish $f$ from symmetric decreasing functions. Hence, the corollary follows from Theorem 3 and Theorem 4. □

Finally, our example function will fulfill the assumptions of Corollary 1 for $\varepsilon = 1/6$. It is an instance of one of the best known NP-equivalent optimization problems namely the MAXSAT problem. For reasons of completeness, we define all necessary notions. A literal is a Boolean variable $x_i$ or a negated Boolean variable $\overline{x}_i$. A literal is satisfied by an assignment or input $a = (a_1, \ldots, a_n)$ if its Boolean value equals 1. A clause is a disjunction (Boolean OR) of some literals, i. e., a clause is satisfied by an input $a$ iff at least one of its literals is satisfied. An instance of the MAXSAT problem is specified by a sequence of clauses $c_1, \ldots, c_m$ over the variables $x_1, \ldots, x_n$ and the task is to find an input satisfying simultaneously as many clauses as possible.

The following instance of MAXSAT has been presented by Papadimitriou (1994). It consists of $n$ clauses of length 1 and $n(n-1)(n-2)$ clauses of length 3 each, more precisely the clauses

- $x_i$, $1 \leq i \leq n$, and
- $x_i \vee \overline{x}_j \vee \overline{x}_k$, $(i,j,k) \in \{1, \ldots, n\}^3$, $i \neq j \neq k \neq i$ .

All clauses of this special instance have exactly one positive literal. Such clauses are called Horn clauses and correspond to typical database queries. Altogether, this example function is a non-artificial instance of a well-known problem and has a simple description. Moreover, it is easy to optimize for humans. The reader will find out in a second that the all one string $1^n$ is the only one to satisfy all clauses. Nevertheless, a popular randomized search heuristic developed especially for MAXSAT needs expected exponential time for this instance (Papadimitriou (1994)). The same can be proved for the algorithm due to Schöning (1999) which is the best known for MAXSAT (with respect to expected worst case time).

In order to apply Corollary 1 we translate the MAXSAT instance into a polynomial $\text{COUNT}_n : \{0,1\}^n \to \mathbb{N}$ which counts the number of satisfied clauses. It is easy to verify that

$$\text{COUNT}_n(a) = \sum_{1 \le i \le n} a_i + \sum_{1 \le i \le n} \sum_{1 \le j \le n, j \ne i} \sum_{\substack{1 \le k \le n \\ k \ne i, k \ne j}} (1 - (1 - a_i) a_j a_k).$$

The description of the MAXSAT instance proves that $\text{COUNT}_n$ is symmetric. Hence, we like to describe $\text{COUNT}_n$ by a function $\text{COUNT}^*_n : \{0, \dots, n\} \to \mathbb{N}$ such that $\text{COUNT}_n(a) = \text{COUNT}^*_n(||a||_1)$. Let $s = ||a||_1$. Then

$$\text{COUNT}^*_n(s) = s^3 - (n+1)s^2 + (n+1)s + n(n-1)(n-2),$$

since

$$(1 - (1 - a_i)a_j a_k) = 1 - a_j a_k + a_i a_j a_k$$

and

$$\begin{aligned}
\text{COUNT}_n(a) &= \sum_{1 \le i \le n} a_i + n(n-1)(n-2) - 2(n-2) \sum_{1 \le j < k \le n} a_j a_k + \\
&\quad 6 \sum_{1 \le i < j < k \le n} a_i a_j a_k \\
&= s + n(n-1)(n-2) - 2(n-2)\binom{s}{2} + 6\binom{s}{3} \\
&= s + n(n-1)(n-2) - (n-2)s^2 + (n-2)s + s^3 - 3s^2 + 2s \\
&= \text{COUNT}^*_n(s).
\end{aligned}$$

Finally, it follows by standard arguments that $\text{COUNT}^*_n$ is decreasing for all $s$ where $0 \le s < (2/3)n$. Hence, Corollary 1 can be applied for $\varepsilon = 1/6$.

**Corollary 2.** *The probability that simulated annealing or an evolution strategy finds the optimum of* $\text{COUNT}_n$ *within* $\exp(o(n^{1/2}))$ *steps is bounded above by* $\exp(-\Omega(n^{1/2}))$.

The function $\text{COUNT}_n$ cannot be optimized efficiently by any search heuristic fulfilling the stated hypothesis and, therefore, we assume that no "reasonable" search heuristic is efficient for $\text{COUNT}_n$.

# 6 Conclusions

The NFL theorem is a simple theorem ruling out statements that some search heuristics have some advantage on the average of "all" functions. However, the NFL scenario is not a realistic one. For realistic black box scenarios, in particular those defined by some restrictions on the complexity of the considered functions, NFL theorems will not hold, but at least a free appetizer is possible in some situations. The ANFL theorem proves that one cannot expect much by well-chosen heuristics in complexity restricted black box scenarios. Search heuristics implement a guess on the class of functions they are confronted with. If the guess is correct, they can be much better than other search heuristics using other guesses. However, there are simple non-artificial functions where some frequently used search heuristics can be proved to need exponential time with overwhelming probability and where it is conjectured that this holds for all "reasonable" search heuristics.

## References

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

S. Droste, T. Jansen, and I. Wegener (1999). Perhaps not a free lunch but at least a free appetizer. In W. Banzhaf, J. Daida, A. Eiben, M. Garzon, V. Honavar, M. Jakiela, and R. Smith (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99)*, 833–839. Morgan Kaufmann, San Francisco.

D. Fogel (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ.

F. Glover and M. Laguna (1993). Tabu search. In *Modern Heuristic Techniques for Combinatorial Problems*. Scientific Publications, Oxford.

D. Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.

T. Hagerup and C. R. Rüb (1989). A guided tour of Chernoff bounds. *Information Processing Letters 33*, 305–308.

D. S. Hochbaum (Ed.) (1996). *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, Boston, MA.

M. Li and P. Vitányi (1993). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, Berlin.

R. Motwani and P. Raghavan (1995). *Randomized Algorithms*. Cambridge University Press, Cambridge.

C. H. Papadimitriou (1994). *Computational Complexity*. Addison-Wesley, Reading, MA.

C. H. Papadimitriou and K. Steiglitz (1998). *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, Englewood Cliffs, NJ.

Y. Rabani, Y. Rabinovich, and A. Sinclair (1995). A computational view of population genetics. *Random Structures and Algorithms 12(4)*, 314–334.

N. Radcliffe and P. Surry (1995). Fundamental limitations on search algorithms: evolutionary computing in perspective. In J. van Leeuwen (Ed.), *Lecture Notes in Computer Science*, Volume 1000. Springer, Berlin, Germany.

U. Schöning (1999). A probabilistic algorithm for $k$-SAT and constraint satisfaction problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS '99)*, 410–414. IEEE Press, Piscataway, NJ.

H.-P. Schwefel (1995). *Evolution and Optimum Seeking.* Wiley, New York, NY.

P. van Laarhoven and E. Aarts (1987). *Simulated Annealing: Theory and Practice.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

D. L. Whitley (1997). Permutations. In T. Bäck, D. B. Fogel, and Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, C1.4:1–8. Institute of Physics Publishing and Oxford University Press, Bristol, New York.

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