The Information Cost of No Free Lunch

***Abstract*— The No Free Lunch Theorem (NFLT) is a great leveler of search algorithms, showing that on average no search outperforms any other. Yet in practice searches do outperform others. In consequence, some have questioned the significance of the No Free Lunch Theorem (NFLT) to the performance of search algorithms. To properly assess the significance of the NFLT for search, one must understand the precise sources of information that affect search performance. Our purpose in this paper is to elucidate the NFLT by introducing an elementary theoretical framework for identifying and measuring the information utilized in search. The theory we develop shows that the NFLT can be used to measure, in bits, the fundamental difficulty of search, known as "endogenous information." This quantity in turn enables us to measure, in bits, the effects of prescribed implicit knowledge for assisting a search, known as "active information." Such knowledge often concerns search space structure as well as proximity to and identity of target. Active information can be explicitly teleological or can result implicitly from knowledge of the search space structure. The evolutionary simulations Avida and *ev* are shown to contain large amounts of active information.**

I. Introduction

The No Free Lunch Theorem (NFLT) dictates than, on average, one search technique performs as well as any other [20][29][35]. As with many important insights, the No Free Lunch Theorem (NFLT) seems obvious upon reflection. Obviousness and familiarity, however, should not necessarily lead to dismissal nor disregard. Like Gödel's incompleteness theorem, the NFLT reveals an impossibility, namely, that there exists no magic bullet search algorithm [7] that performs better on average than any other seach algorithm. Gödel's result establishes important limitations for computer science, as revealed in the *Turing halting problem* [6] and *Rice's theorem* [27]. Likewise, the NFLT establishes important limitations for search algorithms, as revealed in the amount of active information that must be infused into a search for it to be successful.

Häggström [17] helpfully illustrates the main point of the NFLT. If we see cards from a well shuffled poker deck laid face down on a table, our chances of locating the ace of spades (A♠) by turning over five cards does not depend on the order of cards turned. Seeing the king of clubs (K♣) turned up, for instance, offers no insight into which card to turn up next. Indeed, any method for turning up cards is as good as any other and does not improve the probability of success. Thus, no matter how clever the method used, the chances of choosing A♠ is the same.

The inherent difficulty of finding A♠, which is formulated probabilistically, can be expressed in bits. Shannon information relates a probability p to an information measure $I = -\log_2(p)$ where the base 2 logarithm gives units in bits. Independent of the search strategy, the probability of success in turning five cards from a standard deck of 52 cards is $p = 1 - \frac{47}{52} = 0.0962$. This probability corresponds to an *endogenous information* of $I_\Omega = -\log_2 p = 3.4$ bits. Endogenous information characterizes the inherent difficulty of solving a problem (in this case, finding A♠ under the condition of pure randomness).

Thus, to increase the probability of finding A♠, more information is required. Such an information source might indicate, after each turn of a card, whether one is getting physically closer or farther from the target (i.e., A♠) by saying "warmer" or "colder," as in the children's game. Alternatively, one might learn that the cards, laid in a line, are such that adjacent cards differ at most only by two. Thus, one would know that cards next to an uncovered J♦ are only 9s, 10s, Jacks, Queens, and Kings. Given this additional information, one would refrain from turning over the adjacent cards on either side if one were trying to find A♠.

We call such additional information *active information*. In this example, the applied *active information* prohibits the A♠ from being on either side of a Jack. Active information is the information that a search contributes over and above endogenous information. Active information can thus be used to advantage for finding A♠ with greater probability than random search.

Following Häggström [23], let the search space Ω be partitioned into a set of acceptable solutions T and a set of unacceptable solutions $\overline{T}$. The search problem is to find a point in T as expediently as possible. With no active information, the distribution of the space is assumed uniform. This assumption dates to the 18th century and Bernoulli's *principle of insufficient reason* [2] which states, "in the absence of any prior knowledge, we must assume that the events [in Ω] ... have equal probability." This is equivalent to an assumption of maximum (information theoretic) entropy on the search space [25]. Many criticisms of the NFLT are aimed at the uniformity assumption [33]. Knowledge of deviation from uniformity, however, is a type of implicitly applied active information that is consistent with the uniformity premise of the NFLT.

II. Active Information

"Familiarity zones," like comfort zones, can become so ingrained that we take for granted things that we have no right to take for granted. The *no information* required for uniformity conceptually parallels the difficulty of understanding *nothing* that science says existed before the big bang. A common visualization of the universe before the big bang is a large black void empty space. No. This is a flawed visualization. Before the big bang there was *nothing*. A *large black void empty space* is *something*. So space is purged from our visualization. This brings us conceptually to the idea "There was nothing. Then, all of a sudden..." No. This doesn't work either. *All of a sudden* presupposes there was time and physics says that time was created at the big bang. The concept of *nothing* is seen to be conceptually difficult largely because the idea is so divorced from our familiarity zones.

Similar erroneous presuppositions have been applied to interpretation of the NFLT. In refuting the NFLT, for example, critics talk of search structures having "links" in the optimization space and smoothness constraints allowing for use of "hill-climbing" optimization [17]. Making such assumptions about underlying search structures is not only common but also vital to the success of optimizing searchers (e.g., adaptive filters and the training of layered perceptron neural networks — [26]). Such assumptions, however, are useless when searching to find a sequence of, say, 7 letters from a 26-letter alphabet to form a word that will pass successfully through a spell checker, or choosing a sequence of commands from 26 available commands to generate a logic operation such as XNOR [21].

With no metric to determine nearness, the search landscape for such searches can be binary—either success or failure. There are no sloped hills to climb. We note that such search problems on a binary landscape are in a different class from those useful for, say, adaptive filters. But, in the context of the NFLT, knowledge of the class in which an optimization problem lies provides active information about the problem.

Active information is indispensable for increasing the probability of success of a search. Active information applied to a search should not be prescribed blindly, but must accurately reflect constraints on the target location. If, in the search for A♠, we are told that we are getting "warmer" when in fact we are getting "colder," the active information contributed to the search can, depending on the algorithm crafted around this information, be negative and thus result in a probability of success less than that of random search.

The NFLT presupposes the absence of active information. By contrast, when critics of the NFLT talk about a "geographical structure," a "link structure," or "clustering" tendencies [17], they introduce active information. A statement like "In almost all concrete optimization problems, we have some prior information" does not so much refute as reinforce the NFLT.

III. The Impact of the NFLT

Concerning the NFLT, Ho and Pepyne write "unless you can make prior assumptions about the ... [problems] you are working on, then no search strategy, no matter how sophisticated, can be expected to perform better than any other" [20]. According to Wolpert and Macready search can be improved only by "incorporating problem-specific knowledge into the behavior of the [optimization or search] algorithm" [35]. Anticipating the NFLT, Schaffer [29] notes "a learner [without active information] ... that achieves at least mildly better-than-chance performance ... is like a perpetual motion machine." The "prior assumptions" and "problem specific knowledge" required for "better-than-chance performance" in evolutionary search derives from active information that, when properly fitted to the search algorithm, favorably guides the search.

Häggström, a critic of the NFLT, says that the NFLT has been "hyped" by researchers such as Wolpert and Macready [35], Ho and Pepyne[20], and Dembski [9]. The opposite is true. The impact of the NFLT on the field of evolutionary algorithms has been significant. Here are some of the immediate impacts.

- The NFLT puts to rest the inflated claims for the information-generating power of evolutionary simulations such as Avida [21] and *ev* [30]. The NFLT gauges the amount of active information that needs to be introduced to render an evolutionary search successful [23]. Like an athlete on steroids, many such programs are doctored, intentionally or not, to succeed [22].
- Christensen and Oppacher note the NFLT is "very useful, especially in light of some of the sometimes outrageous claims that had been made of specific optimization algorithms" [4]. Search algorithms do not contribute information to the search, and the NFLT exposes the inappropriateness of such claims.
- The NFLT shows that claims about one algorithm outperforming another can only be made in regard to benchmarks set by particular targets and particular search structures. Performance attributes and empirical performance comparisons cannot be extrapolated beyond such particulars. There is no all-purpose "magic bullet" search algorithm for all problems [7], [32].

Although commonly used evolutionary algorithms such as particle swarm optimization [10] and genetic algorithms [16] perform well on a wide spectrum of problems, there is no discrepancy between the successful experience of practitioners with such versatile algorithms and the NFLT imposed inability of the search algorithms themselves to create information [4], [11]. The additional information often lies in the experience of the programmer who prescribes how the knowledge about the search is to be folded into the search algorithm. The NFLT takes issue with claims that one search procedure invariably performs better than another or that remarkable results are due to the search procedure alone [1], [3], [18], [19], [22], [24], [28], [30], [31].

IV. Measuring Active Information

The active information prescribed in an evolutionary search can be measured [23]. With a uniform prior, the probability of choosing an element from T in Ω in a single query is

$$p = \frac{|T|}{|\Omega|} \tag{1}$$

where $|S|$ denotes the cardinality of the set S. On average, with no active information, all queries will have this same probability of success. The probability of at least one success in Q queries without replacement is

$$p_{wo} = 1 - \frac{(|\Omega| - Q)!}{|\Omega|!} \frac{(|\Omega| - |T|)!}{(|\Omega| - |T| - Q)!}$$

The NFLT assumes sampling without replacement [35]. For $p \ll 1$ and $Q \ll |\Omega|$ sampling with and without replacement are nearly identical. Thus, we can write, to a good approximation,

$$p_{wo} \approx p_w = 1 - (1 - p)^Q. \tag{2}$$

As an example, consider finding the following phrase taken from Shakespeare's *Hamlet* [23].

$$\textbf{ME*THINKS*IT*IS*LIKE*A*WEASEL} \quad (3)$$

Using an alphabet of $N = 27$ characters (26 letters and a space), the probability of choosing these $L = 29$ specific characters in a single query is

$$p = N^{-L} = 27^{-29} = 3.0935 \times 10^{-42}.$$

To increase this probability, Dawkins [8] uses a *partitioned search* [23] for the phrase by randomly choosing letters and, if there is a match, keeping the letter. For example, if the first set of randomly chosen letters has an **M** in the first position and an **L** in the last, our search for the letters in these positions is finished. These letters are kept as is, and the search continues only for the letters not yet successfully identified. Assuming uniformity, the probability of successfully identifying a specified letter with sample replacement at least once in Q queries is $1-(1-\frac{1}{N})^Q$ and the probability of identifying all $L = 29$ letters in Q queries is

$$p_{ps} = \left(1 - \left(1 - \left(\frac{1}{N}\right)\right)^Q\right)^L \quad (4)$$

To compare to purely random queries, we can rewrite (2) as [23]

$$p_w = 1 - \left(1 - \left(\frac{1}{N}\right)^L\right)^Q \quad (5)$$

Although similar in appearance, $p_{ps} \geq p_w$ with equality only for $Q = 1$. For the parameters under consideration and moderate values of Q, we have $p_w \ll p_{ps}$. If there are $Q = N^L = 3.2326 \times 10^{41}$ queries, then to an excellent approximation[1] $p_w = 1 - e^{-1} = 0.6321$, whereas solving for Q in (4) gives the same probability of success after only $Q = 110$ queries when $p_{ps} = 0.6321$. For this probability, the ratio of the Q's reveals the random search is 2.9387×10^{41} per cent worse than partitioned search. Partitioned search contributes an *enormous* amount of information.

The random queries probability in (5) is small and therefore corresponds to a large amount of information. For a fixed Q, the partitioned search is more probable and therefore has less information. The difference in these values is the active information, in bits, front loaded in the partitioned search.

Here is a simple example of measuring active information in bits. Suppose we knew *apriori* the 12 characters used in (3) and did not have to search all 27 characters of the alphabet. The characters used in the phrase are

$$\{\textbf{ME*THINKSLAW}\} \quad (6)$$

For a single query, if we choose randomly from this smaller character set, $p_w = (1/12)^{29} = 5.0553 \times 10^{-32}$ or $I_w = 104$ bits of information. For a single query to the set of 27 characters, $p_w = (1/27)^{29} = 3.0935 \times 10^{-42}$ or $I_w = 138$ bits of information. Reducing the characters from $N = 27$ to $N = 12$ therefore front loads $138 - 104 = 34$ bits of active information to the search.

Prescription of the programmer's knowledge into formation of active information must be accurate to be useful. If, for example, we use the reduced character set {**QE*THINKSLAW**} instead of (6), the probability of success will plummet to zero. The letter **M**, necessary for success, is absent. We would then have *negative* active information. Active information must be applied with knowledge about the target sought or the search structure in which the search is performed. If information is added randomly, the probability of the search succeeding is as likely to increase as decrease. This is the essence of the NFLT.

External knowledge about a target location or fitness characteristic can be prescribed to a search algorithm in different ways and result in different degrees of active information. For the weasel phrase in (5), an implementation by Dawkins took 43 steps using the stochastic partitioned search algorithm [8]. Longer phrases would require, on average, more steps. Partitioned search, however, is an inefficient use of the the external knowledge provided for partitioned search. The external intelligence used for partitioned search is sufficient to perform a deterministic perfect search in $N - 1 = 26$ steps independent of the length of the message, L. One simply begins with **A** and freezes every position in which we find an **A** . The process is repeated for **B** all the way through **Z**. Any character not yet exposed after **Z** must be a space. Sans capitalization and punctuation, the entire works of Shakespeare, including (5), can be generated in 27 steps. For the weasel phrase in (5), the deterministic approach provides $\log_2(43/26)/29$ =0.0250 more bits of active information per query.

Another criticism of the NFLT, especially in regards to biological evolution, is denial that biological evolution constitutes a targeted search, i.e. the process is nonteliological. The NFLT thus no longer applies since its point is to gauge how well searches perform at locating targets. Hence, after illustrating biological evolution with his ME*THINKS*IT*IS*LIKE*A*WEASEL computer simulation, which clearly does constitute a targeted search, Dawkins immediately adds: "Life isn't like that. Evolution has no long-term goal. There is no long-distant target, no final perfection to serve as a criterion for selection" ([8], p. 50).

Dawkins here fails to distinguish two equally valid and relevant ways of understanding targets: (1) targets as humanly constructed patterns that we arbitrarily impose on things to suit our interests and (2) targets as patterns that exist independently of us and therefore regardless of our interests. In other words, targets can be extrinsic (i.e., imposed on things from outside) or intrinsic (i.e., inherent in things as such).

In the field of evolutionary computing, to which the weasel example belongs, targets are given extrinsically by programmers who attempt to solve problems of their choice and preference. But in biology, not only has life come about without our choice or preference, but there are only so many ways that matter can be configured to be alive and, once alive, only so many ways it can be configured to serve biologically

[1] Because $\lim_{\varepsilon \to 0}(1-\varepsilon)^{1/\varepsilon} = \exp(-1)$.

significant functions. Most of the ways open to biological evolution, however, are dead ends. It follows that survival and reproduction sets the intrinsic targets for biological evolution.

The idea of intrinsic targets is consistent Cambridge paleobiologist Simon Conway Morris' observation of evolutionary convergence [5] to only a few biological endpoints. Geographically separated evolution of similar features (such as the camera eye of humans and squids) are numerous. Conway Morris [5] therefore theorizes that if the evolutionary process were restarted from the beginning, the life forms of today, including humans, would re-evolve. From the perspective of the NFLT, these limited number of endpoints on which evolution converges constitute intrinsic targets, crafted in part by the environment and by laws of physics and chemistry.

V. Active Information in Computer Models of Evolution

To illustrate use of active information in simulations of evolution, we will examine the Avida [21]and *ev* [30]programs. Both are teleological. The Avida program celebrates performance of creation of an XNOR operation which they call EQU whereas *ev* has a fixed binary string for its target. Both infuse active information into the search.

A. Avida

The NAND gate is one of two logic functions from which all other logic can be synthesized. The other is the NOR. Chips containing a *sea of gates* [12], [13], all NAND, are therefore capable of universal logic. This property has attracted attention to evolutionary development of logic operations [14], [15], [36].

The Avida program is an example of evolutionary development of logic using the NAND gate as its only logic element [21]. For a single genome, the program uses a small alphabet of computer operations to generate commands on a Turing type machine. The instruction tape runs in a continuous loop. There are also a number of external registers for each organism.

1) ***Stair Step Search****:* The active information in Avida comes largely from *stair case search*. By stair step search, we mean building on more probable search results to achieve a more difficult search. We give an example using a *frequency of occurrence* (FOO) stair case [23]. Consider the sequence of K=3 phrases

1. STONE_
2. TEN_TOES_
3. _TENSE_TEEN_TOOTS_ONE_TONE_TEST_SET_

From an N=27 letter alphabet, the final phrase contains $I_\Omega = 36 \log_2 27 = 171$ bits of endogenous information. To find this phrase, we will first search for phrase 1 until a success is achieved. Then, using the FOO of phrase 1, search for phrase 2. The FOO of phrase 2 is used to search for phrase 3. For the three phrases shown, the first (k=1) phrase establishes the alphabet, The second (k=2) establishes the FOO of the alphabet. The FOO for the second and third phrase is the same.

Entry	Logic	OUT	NANDS	Score
1	NOT	either $\overline{X}$ or $\overline{Y}$	1	2
2	NAND	$\overline{X\&Y}$	1	2
3	AND	X&Y	2	4
4	OR_N	either X+$\overline{Y}$ or $\overline{X}$+Y	2	4
5	OR	X+Y	3	8
6	AND_N	either X&$\overline{Y}$ or $\overline{X}$&Y	3	8
7	NOR	$\overline{X+Y}$	4	16
8	XOR	X$\oplus$Y=(X&$\overline{Y}$)+($\overline{X}$&Y)	4	16
9	EQU	$\overline{X \oplus Y}$=(X&Y)+($\overline{X}$&$\overline{Y}$)	5	32

Table I: NAND logic illustrated in Figures 1 through 3. The "&" denotes a logic AND, the "+" is an OR and the overline denotes complement. In the three "either-or" operations, a success is claimed if either one of the two operations is performed. The number in the "NANDS" column is the $\log_2$ of the "Score" column.

If L_k is the number of characters in the kth phrase, then [23]

$$I_+ = I_\Omega - I_X. \tag{7}$$

where the active information is

$$I_X = \sum_{k=1}^{K} L_k H(k, k-1)$$

where the cross entropy [25] between phrase k and $k-1$ is

$$H(k, k-1) = \sum_{n=1}^{N} p_n[k] p_n[k-1] \tag{8}$$

The initial terms are $H(1,0)$=$H(1)$ and $p_n[0] = \frac{1}{N}$.

Use of the K=3 phrase sequence in our example yields $I_K = 142$ bits corresponding to an overall active information of I_+=29 bits. Interestingly, we do better by omitting the second phrase and going directly from the first to the third phrase. The active information increases to I_+=50 bits[2]. Each stair step costs information and contribution to the solution must therefore be weighed against the expended information.

2) Stair Step Active Information in Avida: The goal of Avida is to synthesize a XNOR operation using NAND gates. Avida calls the XNOR operation EQU. To achieve this, the stair step operations of simpler logic operations shown in Table I are used. The fitness, or score, increases with the minimum number of NAND gates required to perform the operation.

The stair step search is evident through inspection of the minimal logic circuits required for each of the entries in Table I. These are shown in Figures 1 through 3. For example, the final EQU in Figures 3 is composed of the XOR operation, Entry 8 in Table I, with a NAND gate. The XOR operation, in turn, is built from the NOT operation and the NAND operation in entries 1 and 2 in Table I. More complex logic operations, therefore, are built with less complex operations. This, indeed, is consistent with Avida's claim that "Complex functions evolved by building on simpler functions that had evolved earlier" [21].

[2]Deleting the first phrase also increases the added information - but by not as much. If the second phrase is searched using a uniform prior, then the added information for finding the third phrase is I_+=38 bits.

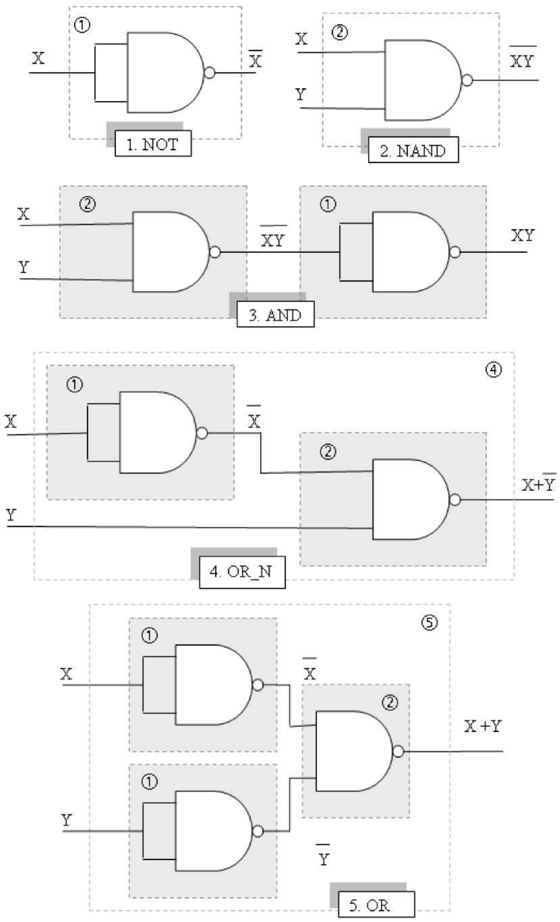

Figure 1: Minimal NAND logic. Logic function is synthesized using the minimum possible number of NAND gates. Each logic circuit corresponds to an entry in Table I. Lower numbered entries are used in the synthesis of higher ordered entries. Continued in Figure 2.

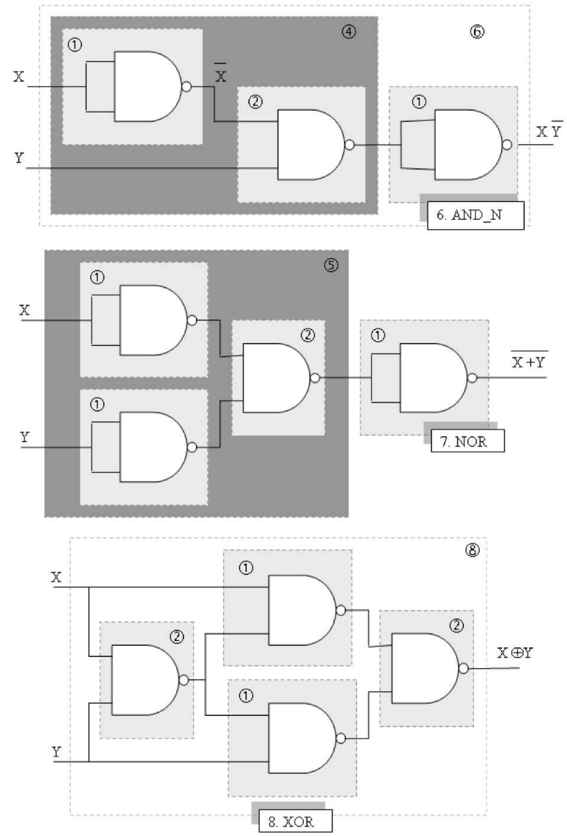

Figure 2: NAND logic. Continued from Figure 1.

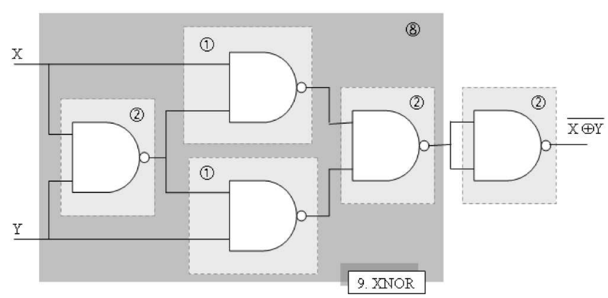

Figure 3: NAND logic for the XNOR operation. Continued from Figure 2.

In describing the Avida operation, we read [21] “Some readers might suggest that we stacked the deck by studying the evolution of a complex feature that could be built on simpler functions that were also useful.” This begs the similar question - would Avida work without the stair step active information being hard wired into the process? The authors of Avida answer this question. In all case studies using stair step active information, “at least one population evolved EQU.” What happens when no active information is applied?

> “At the other extreme, 50 populations evolved in an environment where only EQU was rewarded, and no simpler function yielded energy. We expected that EQU would evolve much less often because selection would not preserve the simpler functions that provide foundations to build more complex features. Indeed, none of these populations evolved EQU, a highly significant difference from the fraction that did so in the reward-all environment.” [21]

Hard wired stair step active information is therefore essential in order for Avida to produce results.

B. *ev*

The program *ev* [30] is an example of a search procedure that front loads an enormous amount of active information into the search. *Ev* purports to simulate the evolution of nucleotide binding sites. Schneider, the program’s author, attributes spectacular results to it, claiming that the program “gains” 131 bits of information “from scratch” [30]. In fact, $I_\Omega = 131$ is merely the endogenous information associated with this problem. With no active information, the expected number of random queries to achieve a success is $Q = 2^{131}$. The active information per query is thus

$$I_\oplus = 131 \times 2^{-131} = 4.8122 \times 10^{-35} \text{ millibits per query.}$$

Active information in *ev* is introduced in two places [22].

1) The Hamming distance between a search estimate and the is available, and
2) The search structure, in the form of a perceptron, is predisposed to finding targets either predominately ones or zeros. The target is predominately zeros.

Ev contributes $I_\oplus = 3$ millibits per query [22]. The *ev* program contains an enormous amount of active information.

VI. Conclusions

Active information, when properly prescribed, successfully guides an evolutionary search to a solution by incorporating knowledge about the target and the underlying structure of the search space. That structure determines how the search deviates from the uniformity assumption of the NFLT. Häggström's "geographical structure[s]," "link structure[s]," search space "clustering," and smooth surfaces conducive to "hill climbing" reinforce rather that refute the quasi-teleological conclusion that the success of evolutionary search depends on the front-loading or environmental contribution of active information. This suggests that in biology, as in computing, there is no free lunch. [9].

References

[1] L.E. Atlas, R. Cole, Y. Muthusamy, A. Lippman, G. Connor, D.C. Park, M. El-Sharkawi & R.J. Marks II, "A performance comparison of trained multi-layer perceptrons and classification trees", in **Neural Networks, Theoretical Foundations & Analysis**, C. Lau, editor, pp.284-288, IEEE Press (1992), reprinted from Proceedings of the IEEE, vol.78, pp.1614-1619 (1990).
[2] Jakob Bernoulli, "Ars Conjectandi (The Art of Conjecturing)," (1713).
[3] J.L. Breeden, J. L. . "An EP/GA synthesis for optimal state space representations," in Proceedings of the Third Annual Conference on Evolutionary Programming, Edited by A. Sebald and L. Fogel, World Scientific, pp. 216-223 (1994).
[4] S. Christensen and F. Oppacher, "What can we learn from No Free Lunch? A First Attempt to Characterize the Concept of a Searchable," Proceedings of the Genetic and Evolutionary Computation (2001).
[5] Simon Conway Morris, *Life's Solution: Inevitable Humans in a Lonely Universe*, Cambridge University Press (September 8, 2003).
[6] Thomas M. Cover and Joy A. Thomas, **Elements of Information Theory**, 2nd Edition, Wiley-Interscience, 2006.
[7] Culberson, J. C. 1998. "On the futility of blind search: an algorithmic view of 'no free lunch,"' Evolutionary Computation, vol. 6, no. 2, pp. 109-127.
[8] Richard Dawkins, **The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design**, W. W. Norton, (1996).
[9] William A. Dembski, **No Free Lunch**. Rowman & Littlefield Publishers, Inc., 2006.
[10] Russell C. Eberhart, Yuhui Shi, James Kennedy, **Swarm Intelligence**, (Morgan Kaufmann 2001).
[11] T.M. English, "Some information theoretic results on evolutionary optimization," Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99. Volume 1, 6-9 July 1999.
[12] A. E. Gamal, J. L. Kouloheris, D. How, and M. Morf, "Bi-NMOS: A basic cell for BiCMOS sea-of-gates," in Proc. IEEE Custom Integrated Circuits Conf., 1989, pp. 831-834.
[13] T. Hanibuchi, K. Higashitani, M. Hatanaka and A. Tada, "A bipolar-PMOS merged basic cell for 0.8 μm BiCMOS sea of gates," IEEE Journal of Solid-State Circuits, Volume 26, Issue 3, Mar 1991, pp.427 - 431.
[14] K. Takita and Y. Kakazu, "Automatic agent design based on gate growth-application to wall following problem'" Proceedings of the 37th SICE Annual Conference. International Session Papers, 29-31 July 1998, pp. 863 - 868.
[15] K. Takita, Y. Kakazu, "Evolutionary design of autonomous agent based on gate growth," Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999. IROS '99, Volume 3, 17-21 Oct. 1999 pp. 1555 - 1560 vol.3.
[16] Tomasz Dominik Gwiazda, *Genetic Algorithms Reference*, (Tomasz Gwiazda, 2006)
[17] Olle Häggström, "Intelligent Design and the NFL Theorem," Biology and Philosophy 22 (2007): 217–230.
[18] Kuk-Hyun Han and Jong-Hwan Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," IEEE Transactions on Evolutionary Computation, Volume: 6, Issue 6, pp.580- 593 (2002).
[19] R. Hinterding, Z. Michalewicz, A.E. Eiben, "Adaptation in evolutionary computation: a survey," Proceedings of the IEEE International Conference on Evolutionary Computation, 13-16 Apr 1997 pp. 65-69 (1997).
[20] Yu-Chi Ho and D.L. Pepyne, "Simple explanantion of the No Free Lunch Theorem", Proceedings of the 40th IEEE Conference on Decision and Control, Orlando, Florida, 2001.
[21] Richard E. Lenski, Charles Ofria, Robert T. Pennock and Christoph Adami, "The evolutionary origin of complex features," **Nature**, vol 423, 139-144 (8 May 2003).
[22] R.J. Marks II *et al.* "Unacknowledged Information Costs in Evolutionary Computing: A Case Study on the Evolution of Nucleotide Binding Sites" (in review). Available online at `www.BobMarks.org`.
[23] R.J. Marks II *et al.*, "Conservation of Information in Search: Measuring the Cost of Success," (in review. Available online at `www.BobMarks.org`.)
[24] R. Mendes, J. Kennedy, J. Neves, " The fully informed particle swarm: simpler, maybe better," IEEE Transactions on Evolutionary Computation, Volume: 8, Issue: 3, pp.204- 210, (June 2004).
[25] A. Papoulis, **Probability, Random Variables, and Stochastic Processes**, 3rd ed. New York: McGraw-Hill, pp. 537-542, 1991.
[26] Russel D. Reed and R.J. Marks II, **Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks**, MIT Press, Cambridge, MA, 1999.
[27] H.G. Rice, "Classes of Recursively Enumerable Sets and Their Decision Problems." Trans. Amer. Math. Soc. 74, 358-366, 1953.
[28] T.P. Runarsson and Xin Yao, "Stochastic ranking for constrained evolutionary optimization," IEEE Transactions on Evolutionary Computation, Volume: 4, Issue 3, pp. 284-294 (2000).
[29] Cullen Schaffer, "A Conservation Law for Generalization Performance," Machine Learning: Proceedings of the Eleventh International Conference, eds. H. Hirsh and W. W. Cohen. San Francisco: Morgan Kaufmann, 1994, pp., 259–265.
[30] T.D. Schneider, "Evolution of Biological Information," Nucleic Acids Research, Vol. 28, No 14, pp.2794-2799 (2000).
[31] Daniel M. Sunday, "A very fast substring search algorithm," Communications of the ACM, Volume 33 , Issue 8, pp.132 - 142 (August 1990).
[32] A.M. Turing, "On computable numbers with an application to the entscheidungs problem," Proceedings of the London Mathematical Society 2, vol.42 pp.230-265 (1936) and vol. 43. pp.544-546.
[33] B. Weinberg and E.G. Talbi, "NFL theorem is unusable on structured classes of problems," Congress on Evolutionary Computation, CEC2004. Volume 1, 19-23 June 2004 pp. 220 - 226 Vol.1
[34] N. Wiener, "A new theory of measurement: A study in the logic of mathematics," Proceedings of the London Mathematical Society, 17, 181- 205, 1921.
[35] David Wolpert, William G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evolutionary Computation 1(1): 67-82 (1997).
[36] M. Yasunaga, T. Nakamura and I. Yoshihara, "Sonar spectrum recognition chip designed by evolutionary algorithm" International Joint Conference on Neural Networks, 1999. IJCNN '99. Volume 5, 10-16 July 1999 pp. 3182 - 3187 vol.5.