Conservation of Information in Search: Measuring the Cost of Success

William A. Dembski, *Senior Member, IEEE*, and Robert J. Marks II, *Fellow, IEEE*

***Abstract*—Conservation of information theorems indicate that any search algorithm performs, on average, as well as random search without replacement *unless* it takes advantage of problem-specific information about the search target or the search-space structure. Combinatorics shows that even a moderately sized search requires problem-specific information to be successful. Computers, despite their speed in performing queries, are completely inadequate for resolving even moderately sized search problems without accurate information to guide them. We propose three measures to characterize the information required for successful search: 1) *endogenous information*, which measures the difficulty of finding a target using random search; 2) *exogenous information*, which measures the difficulty that remains in finding a target once a search takes advantage of problem-specific information; and 3) *active information*, which, as the difference between endogenous and exogenous information, measures the contribution of problem-specific information for successfully finding a target. This paper develops a methodology based on these information measures to gauge the effectiveness with which problem-specific information facilitates successful search. It then applies this methodology to various search tools widely used in evolutionary search.**

***Index Terms*—Active information, asymptotic equipartition property, Brillouin active information, conservation of information (COI), endogenous information, evolutionary search, genetic algorithms, Kullback–Leibler distance, no free lunch theorem (NFLT), partitioned search.**

I. Information as a Cost of Search

OVER 50 years ago, Leon Brillouin, a pioneer in information theory, wrote "The [computing] machine does not create any new information, but it performs a very valuable transformation of known information" [5]. When Brillouin's insight is applied to search algorithms that do not employ specific information about the problem being addressed, one finds that no search performs consistently better than any other. Accordingly, there is no "magic-bullet" search algorithm that successfully resolves all problems [10], [44].

In the last decade, Brillouin's insight has been analytically formalized under various names and in various ways in regard to computer-optimization search and machine-learning algorithms.

1) Schaffer's *Law of Conservation of Generalization Performance* [39] states "a learner . . . that achieves at least mildly better-than-chance performance [without specific information about the problem in question]. . . is like a perpetual motion machine."
2) The *no free lunch theorem* (NFLT) [46] likewise establishes the need for specific information about the search target to improve the chances of a successful search. "[U]nless 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" [23]. Search can be improved only by "incorporating problem-specific knowledge into the behavior of the [optimization or search] algorithm" [46].
3) English's *Law of Conservation of Information* (COI) [15] notes "the futility of attempting to design a generally superior optimizer" without problem-specific information about the search.

When applying Brillouin's insight to search algorithms (including optimization, machine-learning procedures, and evolutionary computing), we use the umbrella-phrase COI. COI establishes that one search algorithm will work, on average, as well as any other when there is no information about the target of the search or the search space. The significance of COI has been debated since its popularization through the NFLT. On the one hand, COI has a leveling effect, rendering the average performance of all algorithms equivalent. On the other hand, certain search techniques perform remarkably well, distinguishing themselves from others. There is a tension here, but no contradiction. For instance, particle-swarm optimization [2], [14], [16], [22] and genetic algorithms [1], [16], [19], [36]–[38], [45], [49], [51] perform well on a wide spectrum of problems. Yet, there is no discrepancy between the successful experience of practitioners with such versatile search algorithms and the COI-imposed inability of the search algorithms themselves to create novel information [7], [13], [15]. Such information does not magically materialize but instead results from the action of the programmer who prescribes how knowledge about the problem gets folded into the search algorithm.

Practitioners in the earlier days of computing sometimes referred to themselves as "penalty function artists" [21], a designation that reflects the need for the search practitioner to inject knowledge about the sought-after solution into the search procedure. Problem-specific information is almost always

Manuscript received May 31, 2008. First published August 14, 2009; current version published August 21, 2009. This paper was recommended by Associate Editor Y. Wang.

W. A. Dembski is with the Department of Philosophy, Southwestern Baptist Theological Seminary, Fort Worth, TX 76115 USA (e-mail: wdembski@designinference.com).

R. J. Marks II is with the Department of Electrical and Computer Engineering, Baylor University, Waco, TX 76798 USA (e-mail: robert_marks@baylor.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2009.2025027

embedded in search algorithms. Yet, because this information can be so familiar, we can fail to notice its presence.

COI does not preclude better-than-average performance on a specific problem class [8], [18], [20], [31], [35], [43]. For instance, when choosing the best k of n features for a classification problem, hill-climbing approaches are much less effective than branch-and-bound approaches [25]. Hill-climbing algorithms require a smooth fitness surface uncharacteristic of the feature-selection problem. This simple insight when smooth fitness functions are appropriate constitutes knowledge about the search solution; moreover, utilizing this knowledge injects information into the search. Likewise, the ability to eliminate large regions of the search space in branch-and-bound algorithms supplies problem-specific information. In these examples, the information source is obvious. In other instances, it can be more subtle.

Accuracy of problem-specific information is a fundamental requirement for increasing the probability of success in a search. Such information cannot be offered blindly but must accurately reflect the location of the target or the search-space structure. For example, to open a locked safe with better-than-average performance, we must either do the following.

1) Know something about the combination, e.g., knowing all of the numbers in the combination are odd is an example of target information.
2) Know something about how the lock works, e.g., listening to the lock's tumblers during access is an example of search-space structure information.

In either case, the information must be accurate. If we are incorrectly told, for example, that all of the digits to a lock's combination are odd when in fact they are all even, and we therefore only choose odd digits, then the probability of success nosedives. In that case, we would have been better simply to perform a random search.

Recognition of the inability of search algorithms in themselves to create information is "very useful, particularly in light of some of the sometimes-outrageous claims that had been made of specific optimization algorithms" [7]. Indeed, when the results of an algorithm for even a moderately sized search problem are "too good to be true" or otherwise "overly impressive," we are faced with one of two inescapable alternatives.

1) The search problem under consideration, as gauged by random search, is not as difficult as it first appears. Just because a convoluted algorithm resolves a problem does not mean that the problem itself is difficult. Algorithms can be like Rube Goldberg devices that resolve simple problems in complex ways. Thus, the search problem itself can be relatively simple and, from a random-query perspective, have a high probability of success.
2) The other inescapable alternative, for difficult problems, is that problem-specific information has been successfully incorporated into the search.

As with many important insights, reflection reveals the obviousness of COI and the need to incorporate problem-specific information in search algorithms. Yet, despite the widespread dissemination of COI among search practitioners, there is to date no general method to apply COI to the analysis and design of search algorithms. We propose a method that does so by measuring the contribution of problem-specific target and search-space structure information to search problems.

II. Measuring Active Information

If we have a combination lock with five thumb wheels, each numbered from zero to nine, our chances of opening the lock in eight tries or less is not dependent on the ordering of the combinations tried. Using the results of the first two failed combinations, for example, [0, 1, 2, 3, 4] and [4, 3, 2, 1, 0], tells us nothing about what the third try should be. After eight tries at the combination lock with no duplicate combinations allowed, no matter how clever the method used, the chances of choosing the single combination that opens the lock is $p = 1 - (99\,992/100\,000) = 8.00 \times 10^{-5}$ corresponding to an *endogenous information* of $I_\Omega = -\log_2 p = 13.6$ b. The endogenous information provides a baseline difficulty for the problem to be solved. In this example, the problem difficulty is about the same as sequentially forecasting the results of 14 flips of a fair coin.

To increase the probability of finding the correct combination, more information is required. Suppose, for example, we know that all of the single digits in the combination that opens the lock are even. There are $5^5 = 3125$ such combinations. Choosing the single successful combination in eight tries or less is $q = 1 - (3117/3125) = 2.56 \times 10^{-3}$, or *exogenous information* $I_S = 8.61$ b. The difference, $I_+ = I_\Omega - I_S = 5.00$ b, is the *active information* and assesses numerically the problem-specific information incorporated into the search that all the digits are even.

More formally, let a probability 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 problem-specific information, the distribution of the space is assumed uniform. This assumption dates to the 18th century and Bernoulli's *principle of insufficient reason* [4], which states that "in the absence of any prior knowledge, we must assume that the events [in Ω] ... have equal probability" [34]. Problem-specific information includes both target-location information and search-space structure information. Uniformity is equivalent to an assumption of maximum (informational) entropy on the search space. The assumption of maximum entropy is useful in optimization and is, for example, foundational in Burg's *maximum entropy method* [6], [9], [34].

The probability of choosing an element from T in Ω in a single query is then

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

where $|\cdot|$ denotes the cardinality of a set. On the average, without use of active information, all queries will have this same probability of success.

For a single query, the available endogenous information is

$$I_\Omega = -\log(p). \tag{2}$$

Endogenous information is independent of the search algorithm used to find the target [39], [46]. To increase the probability of a search success, knowledge concerning the target location or search-space structure must be prescribed. Let q denote the probability of success of a query by incorporating into the search problem-specific information. The difficulty of the problem has changed and is characterized by the exogenous information

$$I_S = -\log(q). \tag{3}$$

If the problem-specific information incorporated into a search accurately reflects the target location, the probability of success of a query will increase. Thus, $q > p$ and the corresponding exogenous information will be smaller than the endogenous information (i.e., $I_\Omega > I_S$). The difference between these two information measures we call the active information. It is conveniently represented as

$$I_+ = -\log\left(\frac{p}{q}\right). \tag{4}$$

This definition satisfies important boundary criteria that are consistent with properties we expect of problem-specific information.

1) For a perfect search, $q = 1$, and the active information is $I_+ = I_\Omega$. The perfect search therefore extracts all available information from the search space.
2) If there is no improvement, the probability of success is the same as that of a random query and $q = p$. The active information in (4) is appropriately zero, and no knowledge of the target location or search-space structure has been successfully incorporated into the search.
3) If the active information degrades performance, then we can have $q < p$ in which case the active information is negative.

As with other log-ratio measures, such as decibels, the active information measure is taken with respect to a reference or baseline, in this case the chance of success of a single random query p. Other baselines can be used.

Single queries generalize directly to multiple queries or samples. Obtaining one or more sixes in four or less rolls of a die, for example, can be considered a target T in a fourfold Cartesian product of the sample space of a single die. Thus, the sequence of four queries can be considered as a single query in this larger search space. References to a single query can therefore correspond to a number of experiments rather than a single measurement.

Active information captures numerically the problem-specific information that assists a search in reaching a target. We therefore find it convenient to use the term "active information" in two equivalent ways:

1) as the specific information about target location and search-space structure incorporated into a search algorithm that guides a search to a solution;
2) as the numerical measure for this information and defined as the difference between endogenous and exogenous information.

Search often requires numerous queries to achieve success. If Q queries are required to extract all of the available endogenous information, then the *average active information per query* is

$$I_\oplus = \frac{I_\Omega}{Q} \text{ bits per query.}$$

III. Examples of Active Information in Search

We next illustrate how active information can be measured and offer, as examples, the active information for search using the common tools of evolutionary search in commonly used search algorithms, including repeated queries, importance sampling, partitioned search, evolution strategy, and use of random mutation. In each case, the source of active information is identified and is then measured.

A. Repeated Queries

Multiple queries clearly contain more information than a single query. Active information is therefore introduced from repeated queries. Assuming uniformity, 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 states that, without added information, the probability of success on the average will be the same no matter what procedure is used to perform the Q queries. The NFLT assumes sampling without replacement [46]. For $p \ll 1$ and $Q \ll |\Omega|$, sampling with and without replacement are nearly identical, and we can write to a good approximation

$$p_{wo} \approx p_w = 1 - (1 - p)^Q. \tag{5}$$

For $pQ \ll 1$, we can approximate $1 - (1 - p)^Q \approx pQ$. Then, for $q = p_{wo}$ in (4), we obtain the interesting result

$$I_+ \approx \log(Q). \tag{6}$$

When the stated assumptions apply, the active information from multiple queries is therefore independent of both the size of the sample space and the probability of success. The repeated random queries also provide diminishing returns. The active information from two queries is 1 b. The active information from 1024 queries is only 1 b more than that from 512.

B. Subset Search

A simple example of active information, due to Brillouin [5], occurs when the target is known to reside in some subset $\Omega' \subset \Omega$. In subset search, we know that $T \subset \Omega' \subset \Omega$ and $q = |T|/|\Omega'|$. Active information is therefore introduced by

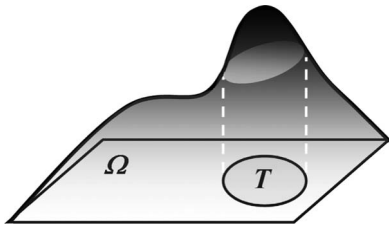


Fig. 1. Importance sampling imposes a probability structure on Ω where more probability mass is focused around the target T. The probability of a successful search increases, and active information is introduced. Without importance sampling, the distribution over Ω is uniform. Note that the placement of the distribution about T must be accurate. If placed at a different location in Ω, the distribution over T can dip below uniform. The active information is then negative.

restricting the search to a smaller number of possibilities. The *Brillouin active information* is

$$I_+ = -\log\left(\frac{\frac{|T|}{|\Omega|}}{\frac{|T|}{|\Omega'|}}\right) = -\log\left(\frac{|\Omega'|}{|\Omega|}\right). \tag{7}$$

Subset search is a special case of importance sampling.

C. Importance Sampling

The simple idea of *importance sampling* [42] is to query more frequently near to the target. As shown in Fig. 1, active information is introduced by variation of the distribution of the search space to one where more probability mass is concentrated about the target.

Our use of importance sampling is not conventional. The procedure is typically used to determine expected values using Monte Carlo simulation using fewer randomly generated queries by focusing attention on queries closer to the target. We are interested, rather, in locating a single point in T. The probability of doing so is

$$q = \sum_{\vec{x}\in T} h(\vec{x}) \tag{8}$$

where $h(\vec{x})$ denotes the probability mass function with the outer product image $\{q_1 \quad q_2 \quad q_3 \quad \cdots \quad q_N\}^{\times L}$. The active information follows from substituting (8) into (4).

Example 1: Let the target T be the first of $|\Omega| = 8$ possible outcomes. Then, $p = 1/8$ and $I_\Omega = 3$ b. If importance sampling is applied so that the probability of choosing the target doubles, then the resulting active information is $I_+ = 1$ b. If importance sampling is inappropriately applied and the probability of choosing the target is half of p, then the resulting active information is negative: $I_+ = -1$ b.

D. FOO

Frequency-of-occurrence (FOO) search is applicable for searching for a sequence of L sequential elements using an alphabet of N characters. With no active information, a character must be assumed to occur at the same rate as any other character. Active information can be bettered by querying in accordance to the FOO of the characters [5], [41]. The endogenous information using a uniform distribution is

$$I_\Omega = L\log_2 N. \tag{9}$$

A priori knowledge that certain characters occur more frequently than others can improve the search probability. In such cases, the average information per character reduces from $\log_2 N$ to the *average information* or *entropy*

$$H_N = -\sum_{n=1}^{N} p_n \log(p_n) \tag{10}$$

where p_n is the probability of choosing the nth character. The active information per character is then[1]

$$\frac{I_+}{L} = \log_2(N) - H_N \qquad \text{bits per character.} \tag{11}$$

If capitalization and punctuation are not used, English has an alphabet of size $N = 27$ (26 letters and a space). Some letters, like E, occur more frequently than others, like Z. Use of FOO information reduces the entropy from $\log_2(27) = 4.76$ b to an entropy of about $H_N = 4.05$ b per character [5]. The active information per character is

$$\frac{I_+}{L} = 0.709 \text{ b per character for English.} \tag{12}$$

DNA has an alphabet of length $N = 4$ corresponding to the four nucleotide A, C, G, and T. When equiprobable, each nucleotide therefore corresponds to 2 b of information per nucleotide. The human genome has been compressed to a rate of 1.66 b per nucleotide [27] and corresponds to an active information per nucleotide of

$$\frac{I_+}{L} = 0.34 \text{ b per nucleotide.} \tag{13}$$

Active FOO information can shrink the search space significantly. For $L \gg 1$, the *asymptotic equipartition property* [9], [41] becomes applicable. In English, for example, if the probability of choosing an E is 10%, then, as L increases, the proportion of E's in the message will, due to the law of large numbers [33], approach 10% of the elements in the message. Let ℓ_n be the number of elements in a *typical* message of length L. Then, for large L, we can approximate $\ell_n \approx p_n L$ and

$$L = \sum_{n=1}^{N} \ell_n. \tag{14}$$

Signals wherein (14) is approximately true are called *typical*. The approximation becomes better as L increases. For a fixed L, the number of typical signals can be significantly smaller than $|\Omega|$, thereby decreasing the portion of the search space requiring examination.

To see this, let $\omega \subset \Omega$ denote the set of typical signals corresponding to given FOO probabilities. The added

[1]This expression is recognized as the Kullback–Leibler distance between the structured search space and the uniform distribution [9].

FOO information reduces the size of the search space from $|\Omega| = N^L$ to[2]

$$|\omega| = 2^{LH_N} \tag{15}$$

where H_N is measured in bits. When search is performed using the FOO, the fractional reduction of size of the search space is specified by the active information. Specifically

$$\frac{|\omega|}{|\Omega|} = \frac{2^{LH_N}}{N^L} = 2^{-I_+} \tag{16}$$

where I_+ is measured in bits. Equation (16) is consistent with (7). For English, from (12), the reduction due to FOO information is $|\omega|/|\Omega| = (2^{-0.709})^L = (0.612)^L$, a factor quickly approaching zero. From (13), the same is true for the nucleotide sequence for which the effective search-space size reduces by a proportion of $|\omega|/|\Omega| = (2^{-0.340})^L = (0.975)^L$. In both cases, the effective reduced search space, though, will still be too large to practically search even for moderately large values of L.

Yockey [50] offers another bioinformatic application. The amino acids used in protein construction by DNA number $N = 20$. For a peptide sequence of length L, there are a total of N^L possible proteins. For 1-iso-cytochrome c, there are $L = 113$ peptides and $|\Omega| = N^L = 20^{113} = 1.04 \times 10^{147}$ possible combinations. Yockey shows, however, that active FOO information reduces the number of sequences over 36 orders of magnitude to 6.42×10^{111} possibilities. The search for a specific sequence using this FOO structure supplies about a quarter ($I_+ = 119$ b) of the required ($I_\Omega = 491$ b) information. The effective search-space size is reduced by a factor of $|\omega|/|\Omega| = 2^{-119} = 1.50 \times 10^{-36}$. The reduced size space is still enormous.

Active information must be accurate. The asymptotic equipartition property will work only if the target sought obeys the assigned FOO. As L increases, the match between the target and FOO table must become closer and closer. The boundaries of ω quickly harden, and any deviation of the target from the FOO will prohibit in finding the target. An extreme example is the curious novel *Gadsby* [48] which contains no letter E. Imposing a 10% FOO for an E for even a short passage in *Gadsby* will nosedive the probability of success to near zero. Negative active information can be fatal to a search.

1) Searching a Type Class: More restrictive than active FOO information is a message of length L where the *exact* count of each character is known but their order is not. With this information about the target, the search space shrinks even further to the space $\varpi \subset \omega \subset \Omega$. The cardinality of this *type class* is bounded [9].

$$|\varpi| \leq (L+1)^N. \tag{17}$$

[2]The derivation is standard [9], [41], [50]. The number of ways $\{\ell_n | 1 \leq n \leq N\}$ objects can be arranged is given by the multinomial coefficient $|\omega| = L!/\prod_{n=1}^N \ell_n!$. For large M, from Stirling's formula, $\ln M! \to M \ln M$ and $|\omega| \approx e^{L \ln L} \exp(\prod_{n=1}^N \ell_n \ln \ell_n)$. Applying (14) gives $|\omega| \approx e^{LH_N}$ where the entropy is measured in nats. A nat is the unit of information when a natural log is used, for example, in (2). When measured in bits ($\log_2$), we obtain (15).

From this inequality, we can bind the active information. From (7)

$$I_+ = -\log\left(\frac{|\varpi|}{|\Omega|}\right) \geq -\log\left(\frac{(L+1)^N}{N^L}\right) = I_\Omega - N\log(L+1) \tag{18}$$

where we have used (9).

E. Partitioned Search

Partitioned search [12] is a "divide and conquer" procedure best introduced by example. Consider the $L = 28$ character phrase

$$\texttt{METHINKS} * \texttt{IT} * \texttt{IS} * \texttt{LIKE} * \texttt{A} * \texttt{WEASEL}. \tag{19}$$

Suppose that the result of our first query of $L = 28$ characters is

$$\texttt{SCITAMROFN} * \texttt{IYRANOITULOV}\mathbf{E} * \mathbf{S}\texttt{AM}. \tag{20}$$

Two of the letters $\{\mathbf{E}, \mathbf{S}\}$ are in the correct position. They are shown in a bold font. In partitioned search, our search for these letters is finished. For the incorrect letters, we select 26 new letters and obtain

$$\texttt{OO}\mathbf{T} * \texttt{DENGISEDE}\mathbf{S}\texttt{EHT} * \mathbf{E}\texttt{RA}*\texttt{N}\mathbf{E}\texttt{T}\mathbf{S}\texttt{I}\mathbf{L}. \tag{21}$$

Five new letters are found, bringing the cumulative tally of discovered characters to $\{\mathbf{T}, \mathbf{S}, \mathbf{E}, *, \mathbf{E}, \mathbf{S}, \mathbf{L}\}$. All seven characters are ratcheted into place. The 19 new letters are chosen, and the process is repeated until the entire target phrase is found.

Assuming uniformity, the probability of successfully identifying a specified letter with sample replacement at least once in Q queries is $1 - (1 - 1/N)^Q$, and the probability of identifying all L characters in Q queries is

$$q = \left(1 - \left(1 - \left(\frac{1}{N}\right)\right)^Q\right)^L. \tag{22}$$

For the alternate search using purely random queries of the entire phrase, a sequence of L letters is chosen. The result is either a success and matches the target phrase, or does not. If there is no match, a completely new sequence of letters is chosen. To compare partitioned search to purely random queries, we can rewrite (5) as

$$p = 1 - \left(1 - \left(\frac{1}{N}\right)^L\right)^Q. \tag{23}$$

For $L = 28$ and $N = 27$ and moderate values of Q, we have $p \ll q$ corresponding to a large contribution of active information. The active information is due to knowledge of partial solutions of the target phrase. Without this knowledge, the entire phrase is tagged as "wrong" even if it differs from the target by one character.

The enormous amount of active information provided by partitioned search is transparently evident when the alphabet

is binary. Then, independent of L, convergence can always be performed in two steps. From the first query, the correct and incorrect bits are identified. The incorrect bits are then complemented to arrive at the correct solution. Generalizing to an alphabet of N characters, a phrase of arbitrary length L can always be identified in, at most, $N-1$ queries. The first character is offered, and the matching characters in the phrase are identified and frozen in place. The second character is offered, and the process is repeated. After repeating the procedure $N-1$ times, any phrase characters not yet identified must be the last untested element in the alphabet.

Partitioned search can be applied at different granularities. We can, for example, apply partitioned search to entire words rather than individual letters. Let there be W words each with L/W characters each. Then, partitioned search probability of success after Q queries is

$$p_W = \left(1 - \left(1 - N^{-L/W}\right)^Q\right)^W. \tag{24}$$

Equations (22) and (23) are special cases for $W = L$ and $W = 1$. If $N^{-L/W} \ll 1$, we can make the approximation $p_W \approx Q^W N^{-L}$ from which it follows that the active information is

$$I_+ \approx W \log_2 Q. \tag{25}$$

With reference to (6), the active information is that of W individual searches: one for each word.

F. Random Mutation

In random mutation, the active information comes from the following sources.

1) Choosing the most fit among mutated possibilities. The active information comes from knowledge of the fitness.
2) As is the case of prolonged random search, in the sheer number of offspring. In the extreme, if the number of offspring is equal to the cardinality of the search space and all different, a successful search can be guaranteed.

We now offer examples of measuring the active information for these sources of mutation-based search procedures.

1) Choosing the Fittest of a Number of Mutated Offspring: In evolutionary search, a large number of offspring is often generated, and the more fit offspring are selected for the next generation. When some offspring are correctly announced as more fit than others, external knowledge is being applied to the search, giving rise to active information. As with the child's game of finding a hidden object, we are being told, with respect to the solution, whether we are getting "colder" or "warmer" to the target.

Consider the special case where a single parent gives rise to Φ children. The single child with the best fitness is chosen and used as the parent for the next generation. If there is even a small chance of improvement by mutation, the number of children can always be chosen to place the chance of improvement arbitrarily close to one. To show this, let $\Delta\kappa$ be the improvement in fitness. Let the cumulative distribution of $\Delta\kappa$ be $F_{\Delta\kappa}(x) = \Pr[\Delta\kappa \leq x]$. Requiring there be some chance that a single child will have a better fitness is assured by requiring $\Pr[\Delta\kappa > 0] > 0$ or, equivalently, $F_{\Delta\kappa}(0) < 1$. Generate Φ children, and choose the single fittest. The resulting change of fitness will be $\Delta\kappa_{\max} = \max_{1 \leq \varphi \leq \Phi} \Delta\kappa_\varphi$, where $\Delta\kappa_\varphi$ is the change in fitness of child φ. It follows that $F_{\Delta\kappa_{\max}}(x) = F^{\Phi}_{\Delta\kappa}(x)$ so that the probability the fitness increases is

$$Pr[\Delta\kappa_{\max} > 0] = 1 - F^{\Phi}_{\Delta\kappa}(0). \tag{26}$$

Since $F_{\Delta\kappa}(0) < 1$, this probability can be placed arbitrarily close to one by choosing a large enough number of children Φ. If we define success of a single generation as better fitness, the active information of having Φ children as opposed to one is

$$I_+ = -\log\left(\frac{1 - F^{\Phi}_{\Delta\kappa}(0)}{1 - F_{\Delta\kappa}(0)}\right). \tag{27}$$

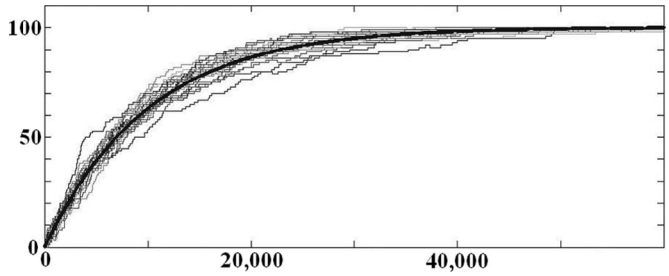


Fig. 2. Twenty-five emulations of optimization using simple mutation as a function of the number of generations. The bold line is (28) with $k_0 = 0$. One parent births two children, the fittest of which is kept. There are cases where the fitness decreases, but they are rare. Parameters are $L = 100$ and $\mu = 0.00005$.

2) Optimization by Mutation: Optimization by mutation, a type of stochastic search, discards mutations with low fitness and propagates those with high fitness. To illustrate, consider a single-parent version of a simple mutation algorithm analyzed by McKay [32]. Assume a binary $(N = 2)$ alphabet and a phrase of L bits. The first query consists of L randomly selected binary digits. We mutate the parent with a bit-flip probability of μ and form two children. The fittest child, as measured by the Hamming distance from the target, serves as the next parent, and the process is repeated. Choosing the fittest child is the source of the active information. When $\mu \ll 1$, the search is a simple Markov birth process [34] that converges to the target.

To show this, let $k[n]$ be the number of correct bits on the nth iteration. For an initialization of k_0, we show in the Appendix that the expected value of this iteration is

$$\overline{k[n]} = k_0 + (L - k_0)\left(1 - (1 - 2\mu)^n\right) \tag{28}$$

where the overbar denotes expectation. Example simulations are shown in Fig. 2.

The endogenous information of the search is $I_\Omega = L$ bits. Let us estimate that this information is achieved by a perfect search in G generations when $\overline{k[G]} = \alpha L$ and α is very close to one. Assume $k_0 = \beta L$. Then

$$G = \frac{\log(1-\alpha) - \log(1-\beta)}{\log(1-2\mu)}.$$

A reasonable assumption is that half of the initially chosen random bits are correct, and $\beta = 1/2$. Set $\alpha = (L - (1/2))/L$. Then, since the number of queries is $Q = 2G$, the number of queries for a perfect search ($I_+ = I_\Omega$) is

$$Q \simeq \frac{2\log(L)}{\log(1-2\mu)}.$$

The average active information per query is thus

$$I_\oplus \simeq \frac{L}{\log(L)}\log(1-2\mu)$$

or, since $\ln(1-2\mu) \simeq -2\mu$ for $\mu \ll 1$

$$I_\oplus \simeq \frac{2\mu L}{\ln(L)}.$$

For the example shown in Fig. 2, $I_\oplus \approx 0.0022$ b per query.

3) Optimization by Mutation With Elitism: Optimization by mutation with elitism is the same as optimization by mutation in Section III-F2 with the following change. One mutated child is generated. If the child is better than the parent, it replaces the parent. If not, the child dies, and the parent tries again. Typically, this process gives birth to numerous offspring, but we will focus attention on the case of a single child. We will also assume that there is a single bit flip per generation.

For L bits, assume that there are k matches to the target. The chance of improving the fitness is a Bernoulli trial with probability $(N-k)/N$. The number of Bernoulli trials before a success is geometric random variable equal to the reciprocal $N/(N-k)$. We can use this property to map the convergence of optimization by mutation with elitism. For discussion's sake, assume we start with an estimate that is opposite of the target at every bit. The initial Hamming distance between the parent and the target is thus L. The chance of an improvement on the first flip is one, and the Hamming distance decreases to $L-1$. The chance of the second bit flip to improve the fitness is $L/(L-1)$. We expect $L/(L-1)$ flips before the next improvements. The sum of the expected values to achieve a fitness of $L-2$ is therefore

$$\text{flips}(L-2) = 1 + \frac{L}{L-1}.$$

Likewise

$$\#\text{ flips}(L-3) = 1 + \frac{L}{L-1} + \frac{L}{L-2}$$

or, in general

$$\#\text{ flips}(L-\ell) = \sum_{m=0}^{\ell-1} \frac{L}{L-m}. \tag{29}$$

It follows that

$$\#\text{ flips}(L-\ell) = L[\mathcal{H}_L - \mathcal{H}_{L-\ell}] \tag{30}$$

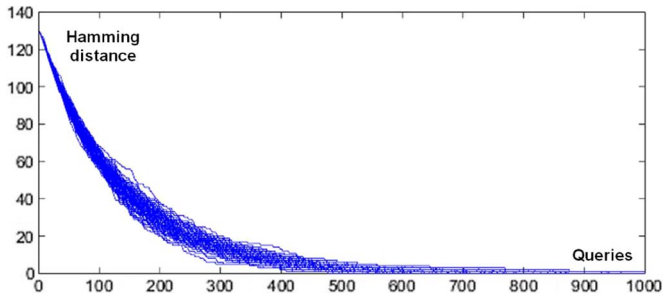


Fig. 3. One thousand implementations of optimization by mutation with elitism for $L = 131$ b.

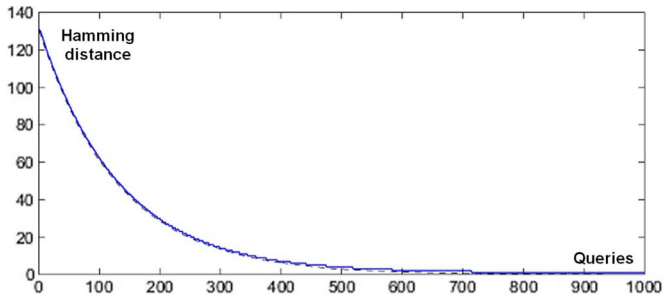


Fig. 4. Average of the curves shown in Fig. 3 superimposed on the stair-step curve from (30). The two are nearly graphically indistinguishable.

where[3]

$$\mathcal{H}_L = \sum_{k=1}^{L} \frac{1}{k}$$

is the Lth harmonic number [3]. Simulations are shown in Fig. 3, and a comparison of their average with (30) is in Fig. 4. For $\ell = L$, we have a perfect search and

$$\#\text{ flips}(0) = L\mathcal{H}_L.$$

Since the endogenous information is L bits, the active information per query is therefore

$$I_\oplus = \frac{1}{\mathcal{H}_L}\text{bits per query.}$$

This is shown in Fig. 5.

G. Stair-Step Search

By stair-step search, we mean building on more probable search results to achieve a more difficult search. Like partitioned search, active information is introduced through knowledge of partial solutions of the target.

We give an example of stair-step search using FOO. Consider the following sequence of $K = 3$ phrases:

1) STONE_;
2) TEN_TOES_;
3) _TENSE_TEEN_TOOTS_ONE_TONE_TEST_SET_.

[3]The conventional notation for the Lth harmonic number is H_L. We use $\mathcal{H}_L$ to avoid any confusion with the symbol for entropy.

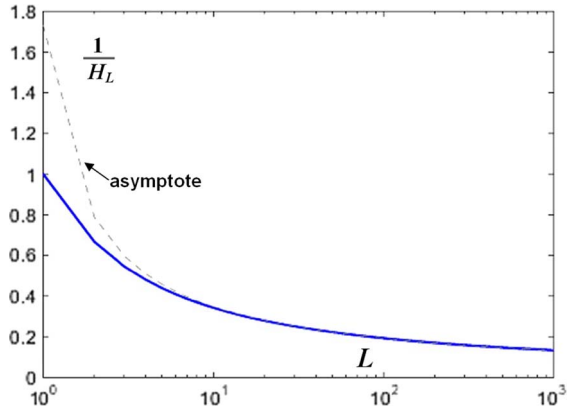


Fig. 5. $1/\mathcal{H}_L$ is the active information per query for optimization by mutation with elitism for a bit string of duration L. A plot of $1/\mathcal{H}_L$ versus L is shown here. The asymptote follows from $\mathcal{H}_L \longrightarrow \ln(L) + \gamma$ as $L \to \infty$ where $\gamma = 0.5772156649\ldots$ is the Euler–Mascheroni constant.

The first phrase establishes the characters used. The second establishes the FOO of the characters. The third has the same character FOO as the second but is longer. From an $N = 27$ letter alphabet, the final phrase contains $I_\Omega = 36 \log_2 27 = 171$ b 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.

Generally, let $p_n[k]$ be the probability of the nth character in phrase k and let the probability of achieving success for the all of the first k phrases be $P_k = \Pr[k, k-1, k-2, \ldots, 1]$. Then

$$P_k = P_{k|k-1} P_{k-1} \tag{31}$$

where the conditional probability is $P_{k|k-1} = \Pr[k|k-1, k-2, \ldots, 1]$. The process is first-order Markov and can be written as

$$P_{k|k-1} = \Pr[k|k-1] = \prod_{n=1}^{N} p_n^{\ell_n[k]}[k-1] \tag{32}$$

where $\ell_n[k] = L_k p_n[k]$ is the occurrence count of the nth letter in the kth phrase and L_k is the total number of letters in the kth phrase. Define the information required to find the kth phrase by $I_k = -\log(P_k)$. From (31) and (32), it follows that

$$I_k = I_{k-1} + L_k H(k, k-1) \tag{33}$$

where the cross entropy between phrase k and $k-1$ is [32]

$$H(k, k-1) = \sum_{n=1}^{N} p_n[k] \log_2 p_n[k-1]. \tag{34}$$

The solution to the difference equation in (33) is

$$I_K = \sum_{k=1}^{K} L_k H(k, k-1) \tag{35}$$

where $H(1, 0) := H(1)$. The active information follows as

$$I_+ = I_\Omega - I_K. \tag{36}$$

Use of the $K = 3$ phrase sequence in our example yields $I_K = 142$ b corresponding to an overall active information of $I_+ = 29$ b. Interestingly, we do better by omitting the second phrase and going directly from the first to the third phrase. The active information increases to $I_+ = 50$ b.[4] Each stair step costs information so that a step's contribution to the solution must be weighed against the cost.

IV. Critiquing Evolutionary-Search Algorithms

Christensen and Oppacher [7] note the "sometimes-outrageous claims that had been made of specific optimization algorithms." Their concern is well founded. In computer simulations of evolutionary search, researchers often construct a complicated computational software environment and then evolve a group of agents in that environment. When subjected to rounds of selection and variation, the agents can demonstrate remarkable success at resolving the problem in question. Often, the claim is made, or implied, that the search algorithm deserves full credit for this remarkable success. Such claims, however, are often made as follows: 1) without numerically or analytically assessing the endogenous information that gauges the difficulty of the problem to be solved and 2) without acknowledging, much less estimating, the active information that is folded into the simulation for the search to reach a solution.

A. Monkey at a Typewriter

A "monkey at a typewriter" is often used to illustrate the viability of random evolutionary search. It also illustrates the need for active information in even modestly sized searches. Consider the production of a specified phrase by randomly selecting an English alphabet of $N = 27$ characters (26 letters and a space). For uniform probability, the chance of randomly generating a message of length L is $p = N^{-L}$. The chances are famously small. Presentation of each string of L characters requires $-\log_2(p)$ b of information. The expected number of repeated Bernoulli trials (with replacement) before achieving a success is a geometric random variable with mean

$$Q = \frac{1}{p} \text{ queries.} \tag{37}$$

Thus, in the search for a specific phrase, we would expect to use

$$B = \frac{-\log_2(p)}{p} = N^L \log_2 N^L \text{ bits.} \tag{38}$$

[4]Deleting the first phrase also increases the active 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 $I_+ = 38$ b.

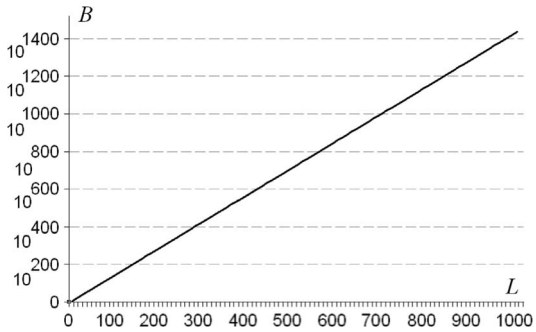


Fig. 6. For $N = 27$, the number of bits B expected for a search of a phrase of length L. From (38), as $L \to \infty$, we have the asymptote $\log(B) \to L \log N$. This explains the linear appearance of this plot.

Astonishingly, for $N = 27$, searching for the $L = 28$ character message in (19) with no active information requires on average $B = 1.59 \times 10^{42}$ b—more than the mass of 800 million suns in grams [11]. For $B = 10^{100}$ b, more than there are atoms in the universe, solving the transcendental equation in (38) reveals a search capacity for a phrase of a length of only $L = 68$ characters. Even in a multiverse of 10^{1000} universes the same size as ours, a search can be performed for only $L = 766$ characters. A plot of the number of bits B versus phrase length L is shown in Fig. 6 for $N = 27$.

Active information is clearly required in even modestly sized searches.

Moore's law will not soon overcome these limitations. If we can perform an exhaustive search for a B bit target in 1 h, then, if the computing speed doubles, we will be able to search for $B + 1$ b in the same time period. In the faster mode, we search for B bits when the new bit is one, and then for B more bits when the new bit is zero. Each doubling of computer speed therefore increases our search capabilities in a fixed period of time by only 1 b. Likewise, for P parallel evaluations, we are able to add only $\log_2 P$ b to the search. Thus, 1024 machines operating in parallel adds only 10 b. Quantum computing can reduce search by a square root [17], [24]. The reduction can be significant but is still not sufficient to allow even moderately sized unassisted searches.

V. Conclusion

Endogenous information represents the inherent difficulty of a search problem in relation to a random-search baseline. If any search algorithm is to perform better than random search, active information must be resident. If the active information is inaccurate (negative), the search can perform worse than random. Computers, despite their speed in performing queries, are thus, in the absence of active information, inadequate for resolving even moderately sized search problems. Accordingly, attempts to characterize evolutionary algorithms as creators of novel information are inappropriate. To have integrity, search algorithms, particularly computer simulations of evolutionary search, should explicitly state as follows: 1) a numerical measure of the difficulty of the problem to be solved, i.e., the endogenous information, and 2) a numerical measure of the amount of problem-specific information resident in the search algorithm, i.e., the active information.

Appendix

Derivation of (28) in Section III-F2.

Since $\mu \ll 1$, terms of order μ^2 and higher are discarded. Let $Z_{k[n]}[n]$ be the number of correct bits that become incorrect on the nth iteration. We then have the conditional probability

$$\Pr\left[Z[n] = q | k[n]\right] = \begin{cases} (1-\mu)^{k[n]} \approx 1 - \mu k[n], & q = 0 \\ \mu k[n](1-\mu)^{N-k[n]-1} \approx \mu k[n], & q = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{39}$$

Similarly, let $O[n]$ be the number of incorrect bits that become correct on the nth iteration. Then

$$\Pr\left[O[n] = q | k[n]\right] \approx \begin{cases} 1 - \mu\left(N - k[n]\right), & q = 0 \\ \mu\left(N - k[n]\right), & q = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{40}$$

For a given $k[n]$, $O[n]$ and $Z[n]$ are independent Bernoulli random variables. The total chance in fitness is

$$\Delta k[n] = Z[n] - O[n]. \tag{41}$$

Since $\mu \ll 1$, the random variable $\Delta k[n]$ can take on values of $(-1, 0, 1)$. The parent is mutated twice, and two mutated children are birthed with fitness changes $\Delta k_1[n]$ and $\Delta k_2[n]$. The child with the highest fitness is kept as the parent for the next generation which has a fitness of

$$k[n+1] = k[n] + \Delta\varkappa[n] \tag{42}$$

where $\Delta\varkappa[n] = \max(\Delta k_1[n], \Delta k_2[n])$. It then follows that

$$\Pr[\Delta\varkappa = q | k] \approx \begin{cases} 2(L-k)\mu(1-L\mu) + (L-k)\mu^2 k, & q = 1 \\ 2(1-L\mu)k\mu + (1-L\mu)^2, & q = 0 \\ k^2\mu^2, & q = -1 \end{cases} \tag{43}$$

where we have used a more compact notation, e.g., $k[n] = k$. Ridding the equation of μ^2 terms gives

$$\Pr[\Delta\varkappa = q | k] \approx \begin{cases} 2(L-k)\mu, & q = 1 \\ 1 - 2(L-k)\mu, & q = 0 \\ 0, & q = -1. \end{cases} \tag{44}$$

Thus, for $\mu \ll 1$, there is a near-zero probability of generating a lower fitness, since doing so requires both offspring to be of a lower fitness and has a probability on the order of μ^2.

The mutation search is then recognized as a Markov birth process.

The conditional expected value of (44) is $E[\Delta\varkappa|k] = 2(L-k)\mu$. We can then take the expectation of (42) and write $\overline{k[n+1]} = \overline{k[n]} + 2(L - \overline{k[n]})\mu$ or, equivalently, $\overline{k[n+1]} = (1-2\mu)\overline{k[n]} + 2L\mu$, where the overline denotes expectation. The solution of this first-order difference equation is (28).

REFERENCES

[1] H. M. Abbas and M. M. Bayoumi, "Volterra-system identification using adaptive real-coded genetic algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 4, pp. 671–684, Jul. 2006.

[2] S. Agrawal, Y. Dashora, M. K. Tiwari, and Y.-J. Son, "Interactive particle swarm: A pareto-adaptive metaheuristic to multiobjective optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 258–277, Mar. 2008.

[3] M. Aigner, *Discrete Mathematics*. Providence, RI: Amer. Math. Soc., 2007.

[4] J. Bernoulli, "Ars conjectandi (the art of conjecturing)," Tractatus De Seriebus Infinitis1713.

[5] L. Brillouin, *Science and Information Theory*. New York: Academic, 1956.

[6] J. P. Burg, "Maximum entropy spectral analysis," Ph.D. dissertation, Stanford Univ., Stanford, CA, May, 1975.

[7] S. Christensen and F. Oppacher, "What can we learn from no free lunch? A first attempt to characterize the concept of a searchable," in *Proc. Genetic Evol. Comput.*, 2001, pp. 1219–1226.

[8] T.-Y. Chou, T.-K. Liu, C.-N. Lee, and C.-R. Jeng, "Method of inequality-based multiobjective genetic algorithm for domestic daily aircraft routing," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 299–308, Mar. 2008.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York: Wiley-Interscience, 2006.

[10] J. C. Culberson, "On the futility of blind search: An algorithmic view of 'no free lunch'," *Evol. Comput.*, vol. 6, no. 2, pp. 109–127, 1998.

[11] J. D. Cutnell and J. W. Kenneth, *Physics*, 3rd ed. New York: Wiley, 1995.

[12] R. Dawkins, *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design*. New York: Norton, 1996.

[13] S. Droste, T. Jansen, and I. Wegener, "Perhaps not a free lunch but at least a free appetizer," in *Proc. 1st GECCO*, pp. 833–839.

[14] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.

[15] T. M. English, "Some information theoretic results on evolutionary optimization," in *Proc. CEC*, Jul. 6–9, 1999, vol. 1, p. 795.

[16] M. S. Fadali, Y. Zhang, and S. J. Louis, "Robust stability analysis of discrete-time systems using genetic algorithms," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 5, pp. 503–508, Sep. 1999.

[17] L. K. Grover, "A fast quantum mechanical algorithm for data search," in *Proc. ACM Symp. Theory Comput.*, 1996, pp. 212–219.

[18] P. Guturu and R. Dantu, "An impatient evolutionary algorithm with probabilistic tabu search for unified solution of some NP-hard problems in graph and set theory via clique finding," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 3, pp. 645–666, Jun. 2008.

[19] T. D. Gwiazda, *Genetic Algorithms Reference*: Tomasz Gwiazda, 2006.

[20] J. S. Gyorfi and C.-H. Wu, "An efficient algorithm for placement sequence and feeder assignment problems with multiple placement-nozzles and independent link evaluation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 437–442, Mar. 2008.

[21] M. J. Healy, The Boeing Company, Personal Communication.

[22] S.-Y. Ho, H.-S. Lin, W.-H. Liauh, and S.-J. Ho, "OPSO: Orthogonal particle swarm optimization and its application to task assignment problems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 288–298, Mar. 2008.

[23] Y.-C. Ho and D. L. Pepyne, "Simple explanation of the no free lunch theorem," in *Proc. 40th IEEE Conf. Decision Control*, Orlando, FL, 2001, pp. 4409–4414.

[24] Y.-C. Ho, Q.-C. Zhao, and D. L. Pepyne, "The no free lunch theorems: Complexity and security," *IEEE Trans. Autom. Control*, vol. 48, no. 5, pp. 783–793, May 2003.

[25] C. A. Jensen, R. D. Reed, R. J. Marks, II, M. A. El-Sharkawi, J.-B. Jung, R. T. Miyamoto, G. M. Anderson, and C. J. Eggen, "Inversion of feedforward neural networks: Algorithms and applications," *Proc. IEEE*, vol. 87, no. 9, pp. 1536–1549, Sep. 1999.

[26] M. Koppen, D. H. Wolpert, and W. G. Macready, "Remarks on a recent paper on the 'no free lunch' theorems," *IEEE Trans. Evol. Comput.*, vol. 5, no. 3, pp. 295–296, Jun. 2001.

[27] G. Korodi, I. Tabus, J. Rissanen, and J. Astola, "DNA sequence compression," *IEEE Signal Process. Mag.*, vol. 47, no. 1, pp. 47–53, Jan. 2007.

[28] C.-Y. Lee, "Entropy—Boltzmann selection in the genetic algorithms," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 1, pp. 138–149, Feb. 2003.

[29] R. E. Lenski, C. Ofria, R. T. Pennock, and C. Adami, "The evolutionary origin of complex features," *Nature*, vol. 423, no. 6936, pp. 139–144, May 8, 2003.

[30] Y. Li and C. O. Wilke, "Digital evolution in time-dependent fitness landscapes," *Artif. Life*, vol. 10, no. 2, pp. 123–134, Apr. 2004.

[31] B. Liu, L. Wang, and Y.-H. Jin, "An effective PSO-based memetic algorithm for flow shop scheduling," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 18–27, Feb. 2007.

[32] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[33] R. J. Marks, II, *Handbook of Fourier Analysis and Its Application*. London, U.K.: Oxford Univ. Press, 2009.

[34] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991, pp. 537–542.

[35] U. S. Putro, K. Kijima, and S. Takahashi, "Adaptive learning of hypergame situations using a genetic algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 5, pp. 562–572, Sep. 2000.

[36] R. D. Reed and R. J. Marks, II, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press, 1999.

[37] K. Takita and Y. Kakazu, "Automatic agent design based on gate growth-application to wall following problem," in *Proc. 37th Int. Session Papers SICE Annu. Conf.*, Jul. 29–31, 1998, pp. 863–868.

[38] F. Rojas, C. G. Puntonet, M. Rodriguez-Alvarez, I. Rojas, and R. Martin-Clemente, "Blind source separation in post-nonlinear mixtures using competitive learning, Simulated annealing, and a genetic algorithm," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 4, pp. 407–416, Nov. 2004.

[39] C. Schaffer, "A conservation law for generalization performance," in *Proc. 11th Int. Conf. Mach. Learn.*, H. Hirsh and W. W. Cohen, Eds., 1994, pp.259–265.

[40] T. D. Schneider, "Evolution of biological information," *Nucleic Acids Res.*, vol. 28, no. 14, pp. 2794–2799, Jul. 2000.

[41] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jul./Oct. 1948, and 623-656.

[42] R. Srinivasan, *Importance Sampling*. Berlin, Germany: Springer-Verlag, 2002.

[43] B. Weinberg and E. G. Talbi, "NFL theorem is unusable on structured classes of problems," in *Proc. CEC*, Jun. 19–23, 2004, vol. 1, pp. 220–226.

[44] A. M. Turing, "On computable numbers with an application to the entscheidungs problem," *Proc. Lond. Math. Soc. Ser. 2*, vol. 42, pp. 230–265, 1936, vol. 43. pp.544-546.

[45] G. S. Tewolde and W. Sheng, "Robot path integration in manufacturing processes: Genetic algorithm versus ant colony optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 278–287, Mar. 2008.

[46] D. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[47] D. H. Wolpert and W. G. Macready, "Coevolutionary free lunches," *IEEE Trans. Evol. Comput.*, vol. 9, no. 6, pp. 721–735, Dec. 2005.

[48] E. V. Wright, *Gadsby*. New York: Lightyear Press, 1997.

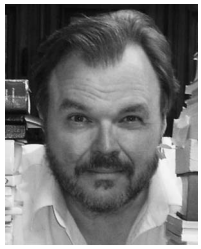
[49] Y.-C. Xu and R.-B. Xiao, "Solving the identifying code problem by a genetic algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 1, pp. 41–46, Jan. 2007.

[50] H. P. Yockey, *Information Theory, Evolution, and the Origin of Life*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[51] F. Yu, F. Tu, and K. R. Pattipati, "A novel congruent organizational design methodology using group technology and a nested genetic algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 1, pp. 5–18, Jan. 2006.

William A. Dembski (M'07–SM'07) received the B.A. degree in psychology, the M.S. degree in statistics, the Ph.D. degree in philosophy, and the Ph.D. degree in mathematics in 1988 from the University of Chicago, Chicago, IL, and the M.Div. degree from Princeton Theological Seminary, Princeton, NJ, in 1996.

He was an Associate Research Professor with the Conceptual Foundations of Science, Baylor University, Waco, TX, where he also headed the first intelligent design think-tank at a major research university: The Michael Polanyi Center. He was the Carl F. H. Henry Professor in theology and science with The Southern Baptist Theological Seminary, Louisville, KY, where he founded its Center for Theology and Science. He has taught at Northwestern University, Evanston, IL, the University of Notre Dame, Notre Dame, IN, and the University of Dallas, Irving, TX. He has done postdoctoral work in mathematics with the Massachusetts Institute of Technology, Cambridge, in physics with the University of Chicago, and in computer science with Princeton University, Princeton, NJ. He is a Mathematician and Philosopher. He is currently a Research Professor in philosophy with the Department of Philosophy, Southwestern Baptist Theological Seminary, Fort Worth, TX. He is currently also a Senior Fellow with the Center for Science and Culture, Discovery Institute, Seattle, WA. He has held National Science Foundation graduate and postdoctoral fellowships. He has published articles in mathematics, philosophy, and theology journals and is the author/editor of more than a dozen books. In *The Design Inference: Eliminating Chance Through Small Probabilities* (Cambridge University Press, 1998), he examines the design argument in a post-Darwinian context and analyzes the connections linking chance, probability, and intelligent causation. The sequel to *The Design Inference* (Rowman & Littlefield, 2002) critiques Darwinian and other naturalistic accounts of evolution. It is titled *No Free Lunch: Why Specified Complexity Cannot Be Purchased Without Intelligence*. He has edited several influential anthologies, including *Uncommon Dissent: Intellectuals Who Find Darwinism Unconvincing* (ISI, 2004) and *Debating Design: From Darwin to DNA* (Cambridge University Press, 2004, coedited with M. Ruse). His newest book, coauthored with J. Wells, is titled *The Design of Life: Discovering Signs of Intelligence in Biological Systems* (Foundation for Thought and Ethics, 2007). His area of interest in intelligent design has grown in the wider culture; he has assumed the role of public intellectual. In addition to lecturing around the world at colleges and universities, he is frequently interviewed on the radio and television. His work has been cited in numerous newspaper and magazine articles, including three front-page stories in *The New York Times* as well as the August 15, 2005 *Time Magazine* cover story on intelligent design. He has appeared on the BBC, NPR (Diane Rehm, etc.), PBS (Inside the Law with Jack Ford and Uncommon Knowledge with Peter Robinson), CSPAN2, CNN, Fox News, ABC Nightline, and the Daily Show with Jon Stewart.



Robert J. Marks II (S'71–M'72–SM'83–F'94) is currently the Distinguished Professor of Engineering at the Department of Engineering, Baylor University, Waco, TX. He served for 17 years as the Faculty Advisor with the Campus Crusade for Christ, University of Washington chapter. He is currently a Distinguished Professor of engineering with the Department of Electrical and Computer Engineering, Baylor University, Waco, TX. His consulting activities include Microsoft Corporation, Pacific Gas & Electric, and Boeing Computer Services. Eleven of his papers have been republished in collections of seminal works. He is the author, coauthor, Editor, or Coeditor of eight books published by MIT Press, IEEE, and Springer–Verlag. His most recent text is *Handbook of Fourier Analysis and Its Applications* (Oxford University Press, 2009). His research has been funded by organizations such as the National Science Foundation, General Electric, Southern California Edison, Electric Power Research Institute, the Air Force Office of Scientific Research, the Office of Naval Research, the Whitaker Foundation, Boeing Defense, the National Institutes of Health, The Jet Propulsion Laboratory, the Army Research Office, and the National Aeronautics and Space Administration (NASA).

Dr. Marks is Fellow of the Optical Society of America. He was given the honorary title of Charter President of the IEEE Neural Networks Council. He is a former Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS. He was the recipient of numerous professional awards, including a NASA Tech Brief Award and a Best Paper Award from the American Brachytherapy Society for prostate-cancer research. He was the recipient of the IEEE Outstanding Branch Councilor Award, the IEEE Centennial Medal, the IEEE Computational Intelligence Society Meritorious Service Award, the IEEE Circuits and Systems Society Golden Jubilee Award, and, for 2007, the IEEE CIS Chapter of the IEEE Dallas Section Volunteer of the Year Award. He was named a Distinguished Young Alumnus of the Rose-Hulman Institute of Technology and is an inductee into the Texas Tech Electrical Engineering Academy. He was the recipient of the Banned Item of the Year Award from the Discovery Institute and a recognition crystal from the International Joint Conference on Neural Networks for contributions to the field of neural networks (2007).