OPTIMIZATION THROUGH EVOLUTION AND RECOMBINATION[*]

H. J. Bremermann

Department of Mathematics
University of California
Berkeley, California

From *Self-Organizing Systems-1962*
M. C. Yovits, G. T. Jacobi, and G. D. Goldstein (eds).
Spartan Books, Washington D. C. (1962)

Part I. Limitations On Data Processing Arising From Quantum Theory

In the second part of this talk I will speak about evolution. In the first part I will present a conjecture which states that there is a maximum rate at which data processing can proceed. This maximum rate applies to all data processing systems, manmade as well as biological.

Practical computers do not reach this rate, which, however, seems to constitute an ultimate physical limitation on the progress of computer design.

Biological systems are subjected to the same limitation. Hence we get an upper bound on the amount of data processing that is going on in nature and on the amount of bits that have been processed since the beginning of life on earth.

I hope that this conjecture may stimulate the discussion towards a tightening of concepts in artificial data processing and biological evolution alike.

The conjecture is the following: *No data processing system whether artificial or living can process more than* (2×10^{47}) *bits per second per gram of its mass.*

This figure refers to a self-contained system where the power supply is included in the total mass. "Processing of n bits" is defined as the transmission of that many bits over one or several channels within the computing system.

For example, in a conventional digital computer information flows back and forth between the arithmetic unit and the memory. These connections constitute channels. The arithmetic unit itself may be considered to be a channel between its input and output terminals. In an analogue computer there are integrator units, curve readers, and curve drawing units which are connected by channels. We count the number of bits per second that flow through all the channels within a computer and this we define as the processing rate.

Some people to whom I have communicated the bound have criticized it as being too imprecise. They would favor sharper estimates for specific systems. Such sharp bounds are certainly desirable. The importance of the bound that I am giving lies in its universality. It is independent of the details of constructions. It applies to serial computers as well as to parallel machines. Bounds even if not sharp can be interesting. Some of the most important formulas in mathematics are inequalities rather an equalities.

The number 2×10^{47} bits per second is a small number when compared with processing rates that one would need to carry out certain search processes. For example, Minsky [9], p. 9, gives the number of all possible move sequences in chess as about 10^{120}. A mosaic of 100 × 100 cells, each of which may either be black or white, has $2^{10,000}$ » 10^{3000} possible patterns. (A resolution of 100 × 100 is not very fine grained. Standard television resolves about 525 × 360 points).

Theorem proving and problem solving also lead to exponentially growing "problem trees." If our conjecture is true then it seems that the difficulties that are currently encountered in the field of pattern recognition and theorem proving will not be resolved by sheer speed of data processing of some future supercomputers.

We have stated our bound per gram and second. There are about $p \times 10^7$ seconds in a year{FNA}. The age of the earth is about 10^9-10^{10} years, its mass less than 6×10^{27} grams. Hence even a computer of the size of the earth could not process more than about 10^{93} bits during a time equal to the age of the earth.

It has been argued that the "number of elementary particles in the universe is finite, and hence that there is a limit to what a computer can do." This is probably true. However, I prefer to have a quantitative statement.

We may arrive at the number as follows:

In order that information can be acted upon by a machine it must be represented physically. Von Neumann [16] has called observables that are used to represent information "markers". Examples of frequently used markers are: The presence or absence of a whole in a paper tape or punch card; the state of a flip-flop, magnetization of a ferrit core or of a strip on a magnetic tape, charged spots in a Williams tube and optical storage.

The information obtained by measuring a random variable that is capable of n values, each taken with probability $p_1 \ldots p_n$, equals

$$H(p_1, \ldots p_n) =$$

This function assumes its maximum for $p_1 = p_2 \ldots = p_n = 1/n$ and $H(1/n, \ldots 1/n) = \log_2 n$.

Hence if a marker can assume n different states of which only one is present at any one moment then it can represent at most $\log_2 n$ bits of information.

Suppose that energy levels are used as markers. Suppose the levels have to lie within a certain interval, say (O, E_{max}). Suppose further that energy levels can be measured with an accuracy DE only. Then at most $n = E_{max} / DE$ different levels can be distinguished. If at each instant no more than one level is occupied, then a maximum of $\log (n + 1)$ bits may be represented. If instead of one marker with energy levels in (O, E_{max}) two markers with levels in $(O, 1/2\ E_{max})$ each are used then 2 log $(n/2 + 1)$ bits can be represented. The optimal use of a given amount of energy E_{max} is attained if n markers with values in (O, DE) are used. In this case n bits can be represented.

Any self-contained computing system of mass m is subject to a limitation in the maximum energy that it can utilize as a marker. This follows simply from the fact that mass and energy are related by Einstein's equation $E = mc^2$, where c is the light velocity in vacuum. Thus

$$E_{max} = mc^2 \text{ » } m \times 9 \times 10^2\ cm^2\ sec^{-2} < m \times 10^{21}\ cm^2\ sec^{-2}.$$

Heisenberg's uncertainty principle, on the other hand, gives a lower limit for the accuracy with which energy can be measured. It asserts

$$DEDt \text{ ³ } h,$$

where DE is the uncertainty in energy. At the duration of the measurement, h is Planck's constant. DE is defined as the mean deviation from the expectation value of E.

Thus a computer of mass m using energy levels as markers can measure no more than

$$(E_{max} / h)\ Dt \text{ bits}$$

during the time interval Dt. (Note that $E_{max}h^{-1}\ Dt$ is a dimensionless quantity).

Information received through a channel must be measured by the receiver. This is done by measuring a time varying signal, for example a time varying voltage or the presence or absence of photons or particles. It seems that any such measurement is equivalent to an energy measurement. Frequency and energy are related by $E = hn$. Hence, if we measure frequency we also measure energy. If we utilize polarisation and spin we have instead of a linear channel a three or more dimensional channel. Such a channel is equivalent to n linear channels if its dimension is n. it does not matter whether we have one or several channels. The total energy available for all channels is still E_{max}.

Thus any machine of mass m processes no more than

$$E_{max}\ h^{-1} \text{ bits per unit of time} < (m\ 10^{21}) / (6 \times 6 \times 10^{-27}) \text{ bits/sec} < 2m\ 10^{47} \text{ bits/sec},$$

where m is measured in grams. This is a generous estimate, mainly for two reasons: We need measuring apparatus for measuring the markers and part of the mass of the computer has to serve this purpose. Secondly, in view of the definition of DE the probability of measuring the value of the marker correctly is less than one. In digital computations the accuracy requirements are very high. To achieve high accuracy redundancy has to be included into the program, thus reducing the effective processing rate.

Criticism: According to Shannon's theory [12,13,14] the rate of information transmission through a channel of bandwidth n_{max} is $n_{max}\ \log_2 (1 + S/N)$ bits per second, where S/N is the signal to noise power ratio. Since energy and frequency are related by E = hn we have $mc^2\ h^{-1}\ \log_2(1+ S/N)$. We thus obtain the same rate of transmission for S/N = 1. On the other hand we have not included in our consideration any thermal noise. In order to reconcile the results we must interpret N as quantum-mechanical noise arising. from quantum mechanical fluctuations of the observable used for the information transmission in the channel.

The argument does not penetrate all the ramifications of the problem and I have formulated the result as a "conjecture."

Bledsoe's absolute bound on memory access time. I have communicated the above conjecture to W. W. Bledsoe in November 1960. He immediately pointed out an interesting consequence. (Bledsoe [1]). The time that is necessary to measure information in the memory and to communicate it to the place where it is used is the "memory access time." Bledsoe has shown that if the above conjecture is true, then there is an absolute lower limit for the, memory access time DT that is valid for all computers:

DT ³ 10^{-20} sec.

There is only one assumption involved, namely that the density d of the material employed In the computer is less than 20. This is true on earth.

Let our bound 2 ×10^{47} bits per second be denoted by b. It implies that the largest number of bits that can be stored per gram of matter with an access time of DT equals

b × DT bits/g

If the density, of matter is d, then we have at most

db × DT bits/cm^3

A computer of volume V therefore has a memory of no more than

Vdb × DT bits.

Bledsoe argues: Information must be transported within the computer. The speed of light is an upper bound for the speed of communication. The average length of the communication path in a computer of volume V is about $(1/4\ V)^{1/3}$. For optimal performance communication time and measuring time should be about equal. $(1/4\ V)^{1/3}$ » cDT. There is an (empirical) bound on the density of matter. The smallest machine conceivable is a 2 bit machine. When we combine these figures we obtain an absolute lower bound for the access time in any computer:

d × $4c^3\ (DT_{min})^3 b\ DT_{min}$ ³ 2.

Hence if d £ 20, then

DT_{min} ³ 10^{-20} sec.

This figure does not immediately give a bound on the processing rate in larger machines since data may be processed in parallel. The minimum access time, however, limits the processing rate in serial machines such as Turing machines.

Thermodynamical Limitations

Brillouin [17] has shown that each observation on a system of harmonic oscillators has a cost in entropy. Consider a system consisting of only one quantized harmonic oscillator. An observation whether the oscillator is at an energy state greater than a given number has a cost in entropy

DS ³ A_1k.

Here *k* is Boltzman's constant and ;

A_1 = ln 2 for "high temperature" hn < kT and (hn) / (kT) for "low temperature" hn³ kT, where T is the temperature of a surrounding thermostat. For a system of n oscillators the constant A_l is replaced by a constant A_n, which increases monotonely with *n* (approximately like log n). The reliability normalization corresponds to a signal to noise ratio S/N = 1. (The noise is due to thermal and quantum mechanical fluctuations.)

The cost in energy that has to be dissipated is DE = DS × T, thus

DE ³ (ln 2) kT for high temperatures; and hn for low temperatures

If we interpret the outcome of such an observation as one bit of information then the cost of energy per bit is DE. The system considered is a very special one. Nevertheless, entropy considerations that are true for harmonic oscillators tend to be true in general. Landauer [8] considering a very different but also rather special system arrived at a minimum cost of kT per bit. He seems not to have included quantum mechanical fluctuations in his considerations.

Thus, in order to achieve low cost in energy dissipation per bit a computer must operate a) at low *temperatures* and b) avoid high frequency quanta. Since a fast transmission rate through a channel means high frequency quanta, the computer must use *many channels* to minimize cost.

Example: A computer operating at a temperature T may employ quanta up to

$$D = kTh^{-1}$$

without losing efficiency due to cost arising from high frequency quanta.

At T = 300° (» room temperature)
n_{max} » 1.38 × 10^{-16} × 300 × 10^{26} sec^{-1}
4 × 10^{12} sec^{-1}

Existing computers operate at frequencies up to 10^6-10^9 sec^{-1}. Thus they work still in the thermodynamical range. At 1°K we have n_{max} » 10^{10} sec^{-1}. Ultra-fast microminiaturized computers with cryotron circuitry some day may reach switching speeds of 10^{-10} sec and thus enter the region where "quantum noise" becomes larger than "thermal noise."

Part II. Evolution Processes and Optimization

The preceding discussion lets us see optimization problems in a new light. Suppose we have a real valued function $f(x_1, . . x_n)$. Suppose the variables take values 0 or 1. Then the function can assume up to 2^n different values. Given the task to find one of those vectors of $(x_1, . . . x_n)$ for which $f(x_1, . . . x_n)$ assumes its maximum. Suppose also that $f(x_1, . . . x_n)$ can be computed for each n-tuple $x_1 . . . x_n$. For small n this problem can be solved by inspection: We compute $f(x_1, . . . x_n)$ for all possible value combinations of $x_1 .... x_n$ and select an n-tuple (or one of possible several) for which f is largest. If n is small ($n < 10$) the task is rather trivial. For $n = 30$ there are already a billion numbers to look at, for $n = 300$, 2^{300}» 10^{90}. If each of these possibilities requires at least one bit of data processing, then the task cannot be carried out within the restrictions discussed in part I.

Biological evolution is an optimization process. In recent years it has been discovered that the method of coding hereditary information is universal throughout nature. All cells that divide from bacteria to mammals contain DNA. The DNA is a long double stranded molecule which looks Me a long ladder with crosslinks when straightened out. There are four different kinds of crosslinks (adenine-tymine, tymine-adenine and guanine-cytosine and cytosine-guanine). Arbitrary sequences of these four different kinds of nucleotide bridges seem to be chemically possible. It is believed that the exact sequence of these bridges constitutes the hereditary information. It has been shown that removal or replacement of a single bridge can cause a mutation. A gene seems to correspond to about 10^5 nucleotide bridges. Human somatic cells are estimated to have about $4´10^9$ nucleotide bridges (Muller [10]) and about 10^4 to $4´10^4$ genes. Since there are four kinds of bridges each is worth up to two bits. Hence there is no more than 10^{10} bits of information in the human genes. Other genetic systems contain similar amount of nucleotide bridges.

Unless the evolution of life has relied on some "divine guidance" it must have had to contend with the difficulties involved in optimization that I have explained.

Suppose "fitness" of a genotype could be measured numerically. Let $f(x_1 . . . x_n)$ be this function. Then, if all nucleotide bridges have to be counted, we have $2^{10\text{\^{}}10}$ possibilities. If genes rather than nucleotide bridges are counted, then there are still $2^{10,000}$» 10^{3000} possible values. As we have seen, this number to much too large for a straightforward search or for a trial and error process.

Now every freshman taking calculus learns that the way to find a relative maximum of a differentiable function is to look for its critical points. A point $(x_1^{(0)} . . . x_n^{(0)})$ is a critical point if and only if

$$\P\ f/\P\ x_n\ (x_1^{(0)}) = 0 \text{ for } n = 1, . . . n.$$

Every local extrema is a critical point, but not vice versa. Suppose we deal with a haploid system and asexual evolution.

Suppose the environment remains constant, that means the fitness function $f(x_1, \ldots x_n)$ is not time dependent. Suppose we "mutate" (change) one gene" (binary variable) at a time. Then the evolution process leads quickly to a stable point.

Let $D_n f\ x_1^{(0)}, \ldots x_n^{(0)} = f(x_1 \ldots {\sim}x_n \ldots x_n) - f(x_1, \ldots {\sim}x_n \ldots x_n)$ where $x_1 \ldots x_n$are binary variables and $\sim x_n$ denotes the complement of x_n. We will call a point $(x_1^{(0)\wedge}n, \ldots x_n^{(0)})$ such that n

$$D_n f\ x_1^{(0)}, \ldots x_n^{(0)} \text{ £ } 0 \text{ for all } n$$

a "point of stagnation."

Obviously a local maximum is a point of stagnation, but the converse is not true. A point of stagnation need not even be a saddle point as the following example shows:

$$f(0,0) = 1/2,\ f(0,1) = f(1,0) = 0,\ f(1,1) = 1.$$

In this case, to make any further progress we have to change two variables at once. In general we may have to change any number of variables between 2 and *n* at once to make any further progress. If we have to change *k* variables at once, there are

different combinations to try. (n/k) is largest for $k = n/2$. By Stirling's formula which gives a good approximation for large n:

$$\ln(n!) \text{ » } n(\ln n\text{-}1).$$

Thus

$$\log_2 \quad \text{ » } n.$$

Thus we are caught again by the flood of possibilities that have to be tried. Instead of changing exactly one "gene" (binary variable) at a time we may operate such that each gene has a certain probability p to change. In many cases optimal speed of improvement is reached for $p = 1/n$ where n = number of genes. (Compare Bremermann [5]). The probability for any one pair of genes to change simultaneously is p^2, for triplets p^3. Hence this mode of operation is extremely slow to overcome points of stagnation. W. W. Bledsoe [2,3] has investigated this question in some special cases.

It is thus fairly obvious that any evolution process whether occurring in nature or whether applied to numerical problems is bound to have difficulties with points of stagnation. Difficulties have indeed been experienced by several colleagues who have tried to solve complicated optimization problems through evolution (compare Minsky's comments [9]). Friedberg [6,7] tried to let a computer evolve a very simple program. However, it took the machine 1000 times longer than it would have taken had it relied on pure chance alone. In many cases where evolution has been tried the fitness function was very complicated. For example, $f(x_1, \dots x_n)$ has been taken to be the "performance" of a game strategy depending upon $x_1 \dots x_n$ as parameters. $f(x_1, \dots x_n)$ can only be sampled by actually playing the game. To avoid unnecessary complications I have chosen for my first experiments fitness functions that are well understood theoretically and where a wealth of efficient numerical techniques exist.

With the help of Mr. S. Salaff I have carried out computation experiments to solve systems of linear equations by evolution. This has been done not in order to compete with existing numerical techniques, but in order to understand fully what is going on. Given a system of linear equations

When a "trial vector" $x_1^{(0)} \dots x_n^{(0)}$ is inserted into the system one obtains residues r_i

As fitness function we take

$$f(x_1^{(0)}, \dots x_n^{(0)}) =$$

if the system has a unique solution, then

$$f(x_1^{(0)}, \dots x_n^{(0)}) = 0$$

if and only if $(x_1^{(0)} \dots x_n^{(0)})$is the solution.

The variables in this case are continuous. We may, however, discretize them and change them by increments Dx_j. Or we may convert the whole problem, after discretizing it, into a binary problem. In either case experiments have borne out what is to be expected: Depending upon the step size we reach a point of stagnation which may or may not be near the solution. For well conditioned systems (ratio of largest to smallest eigenvalue close to one) and with some previous knowledge about the range of the variables and for n not large (no larger than about 10) the method gets us in a few minutes machine time close to the solution. For badly conditioned system the process gets trapped at a point of stagnation. The following diagram shows a level line of the fitness function of a badly conditioned system in two variables and illustrates how the process gets trapped:

A reduction in step size results in further progress but after a while the process gets trapped again. Nature must have to contend with the same difficulty. It seems plausable to conjecture that sexual mating has the purpose to overcome situations where asexual evolution is stagnant. In higher animals we have the diploid mechanism. In single-celled animals we often find a haploid system where from time to time two individuals join to exchange genetic material. Even viruses, often harmful, serve a useful purpose in the process of transduction. The individuals then may go on for many generations without further exchange, till finally another exchange of genetic material takes place.

With the help of Mr. Salaff, who has done all the programming, I have tried to apply such a "mating game" to artificial evolution processes. While asexual evolution (hill climbing) has been tried a good deal, little has been done, to my knowledge, with mating.

At first we have tried linear systems. Starting from different initial values we have evolved by asexual evolution approximate solutions till we reached points of stagnation. Different starts usually lead to different points of stagnation. We then have mated populations of such solutions according to various schemes: In pairs, collectively, by linear superposition and by random combinations. After mating we applied another round of asexual evolution. None of these schemes, however, gave any spectacular results.

In disappointment I left linear systems aside and turned to another set of problems: *Linear programming*.

Given a set of linear inequalities

where m ³ n. The task is to minimize a cost function

$$f(x_1, \ldots x_n) = c_1x_1 + \ldots c_n\,x_n + c_0$$

under the condition that the point $x_1 \ldots x_n$satisfies the constraints (the inequalities). Geometrically the inequalities define a convex set and the task is to find that point in the convex set which has minimum distance from the plane

$$c_1x_1 + \ldots c_nx_n - c_0 = 0$$

The problem is well understood theoretically. We have the simplex method, due to G. Dantzig, by which linear programming problems can be solved.

Let us consider asexual evolution: We start at a randomly selected point in the convex set. We change the variables in small steps. After a mutation we test whether the cost has decreased. If not, we make a step in the opposite direction. After decreasing the cost we test whether the constraints are still satisfied. If this is the case we go to the new point, if not we try another variable (or direction). After a number of tries we end up at (or close to) the boundary of the convex set-but usually not at the solution of the problem. We have reached a point of stagnation. To make further progress in cost reduction and in order to stay in the set, two variables have to be changed simultaneously. If we go to the trouble of computing the dependency we reach after a while the intersection of two boundary planes which is a n-2 dimensional plane and we have two dependencies, and so on.

What happens if instead we play the "mating game"? Suppose we have different random starts and end up on different boundary planes. Then a simple linear superposition brings us back into the set. This is due to the fact that the set is convex. A new round of asexual evolution gives us a new "population" of points on the boundary planes. Then we iterate the mating process, then the asexual evolution and so on. There are a number of fine points that affect the problem and which I am not discussing here. For small problems (4 variables or less) the process works well and brings us close to the solution in a few seconds machine time. Already for 8 variables, however, the whole population tends to "degenerate," to drift over to one of the boundary planes. In this case we do not get back Into the set and the process stagnates again. S. Salaff and I are presently trying to employ additional selection routines that, we hope, will minimize this effect. If successful we shall have given an example, where the "mating game" really works. I plan to write up the details in a Technical Report. W. W. Bledsoe is investigating the effect of mating theoretically. He presently is concerned with the effect of mating on gene distributions in large populations. He has shown that random mating transforms a binomal distribution of genes into a binomial distribution. Bledsoe [18].

As a further development I seethe use *of heuristics*. I would call "heuristical" any method, constraint, or guidance principle that helps to eliminate unpromising possibilities in a search process. Heuristics have been applied successfully by Newell, Shaw and Simon and by Gelernter. I expect that a *theory of* heuristics will be needed and that the mathematical motion *of homomorphism* may play a central role.

Conclusion

Evolution is a fact of life. It has gone a long way to explain biological phenomena. I hope I have demonstrated that there still is a lot to be learned about it. The previous discussion shows that it is quite impossible to attribute a selection advantage to single genes which remains constant for many generations. Such a model is simply not true. This assumption, however, is being used in much of mathematical genetics. Discussions based on this assumption may be valid for short term changes in a population. They must be viewed with suspicion when conclusions are drawn that affect many generations.

Secondly, I believe that we must try to understand evolution in simple cases first before we can hope to use it successfully in complicated cases. Perhaps we can then develop evolution schemes which may be useful in integer programming, non-convex programming or even in machines that learn to play games or recognize patterns.

The experiences of various groups who work on problem solving, theorem proving and pattern recognition all seem to point in the same direction: These problems are tough. There does not seem to be a royal road or a simple method which at one stroke will solve all our problems. My discussion of ultimate limitations on the speed and amount of data processing may be summarized like this: Problems involving vast numbers of possibilities will not be solved by sheer data processing quantity. We must look for quality, for refinements, for tricks, for every ingenuity that we can think of. Computers faster than those of today will be a great help. We will need them. However, when we are concerned with problems in principle, present day computers are about as fast as they ever will be.

We may expect that the technology of data processing will proceed step by step-just as ordinary technology has done. There is an unlimited challenge for ingenuity applied to specific problems. There is also an unending need for general notions and theories to organize the myriad details.

---

* Supported in part by the Preparation Systems Branch, Office of Naval Research.

REFERENCES

1. Bledsoe, W. W.: A Quantum-theoretical Limitation of the Speed of Digital Computers. *IRE Trans. Elec. Comp.*, Vol. **EC-10**, no. 3, Sept. 1961.
2. Bledsoe, W. W.: *The use of biological concepts in the analytic study of systems*, Technical Report, November 1961, Panoramic Research Inc., Palo Alto, Calif.
3. Bledsoe, W. W.: *Lethally dependent genes, using instant selection. Technical Report*, December 1961, Panoramic Research Inc.,. Palo Alto, Calif.
4. Bolgiano, L. P. and Gottschalk, W. M.: Minimum power set by quantum effects. *Proceed. IRE*, vol. **49**, no. 4, April 1961.
5. Bremermann, H. J.: The evolution of intelligence. The nervous system as a model of its environment. *ONR report no. 1*, contract Nonr 477(17), University of Washington,

Seattle, 1958.

6. Friedberg, R. M.: A learning machine, part 1, *IBM J. Res. and Dev.*, vol. **2**, pp. 2-13, January 1958.
7. Friedberg, R. M., Dunham, B., and North, J. H.: A learning machine, part II, *IBM J. Res. and Dev.*, vol. **3**, pp. 282-287, July 1959.
8. Landauer: Irreversability and heat generation in the computing process, *IBM J. Res. and Dev.*, vol. **5**, pp. 183-191, July 1961.
9. Minsky, Marvin: Steps toward artificial intelligence, *Proc. IRE*, vol. **49**, pp. 3-30, January 1961.
10. Muller, H. J.: Evolution by mutation. *Bull. Am. Math. Soc.*, vol. **64**, pp. 137-160, July 1958.
11. Nyquist, H.: Certain topics in telegraph transmission theory, *Trans. AIEE*, vol. **47**, pp. 617-644, April 1928.
12. Shannon, C. E.: A mathematical theory of communication, *Bell Sys. Tech. J.*, vol. **27**, pp. 379-423; July 1948, pp. 623-625, October 1948.
13. Shannon, C. E.: Communication in the presence of noise, *Proc. IRE*, vol. **37**, pp. 10-21, 1949.
14. Shannon, C. E.: Probability of error for optimal codes in a Gaussian channel. *Bell System Tech . J.*, vol.**38**, no. 3, pp. 611-656) 1959.
15. Von Neumann, J.: Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata Studies*, Annals of Math. Studies No. **34**, Princeton 1956.
16. Von Neumann, J.: *The computer and the brain*. Yale University Press, 1958.
17. Brillouin, L.: *Science and information theory*, Academic Press, New York 1956. Second edition 1962.
18. Bledsoe, W. W.: An analysis of genetic populations. To appear.

Editor's Footnotes

A. It should be noted that the number of seconds in a year is slightly more than p $\times 10^7$, being on average ~31,557,600.

First Created: November 19, 1998.
Last Modified: June 12, 2002.
HTML Editor: Robert J. Bradbury