Chapter 1

TOWARDS A THEORY OF ORGANISMS AND EVOLVING AUTOMATA OPEN PROBLEMS AND WAYS TO EXPLORE

Heinz Mühlenbein

Abstract We present 14 challenging problems of evolutionary computation, most of them derived from unfinished research work of outstanding scientists such as Charles Darwin, John von Neumann, Alan Turing, Claude Shannon, and Anatol Rapaport. The problems have one common theme: Can we develop a unifying theory or computational model of organisms (natural and artificial) which combines the properties structure, function, development, and evolution? There exist theories for each property separately and for some combinations of two. But the combination of all four properties seems necessary for understanding living organisms or evolving automata. We discuss promising approaches which aim in this research direction. We propose stochastic methods as a foundation for a unifying theory.

1. Introduction

The aim of this book is very ambitious. Its title is not: important problems in evolutionary computation, but *Hilbert problems* in evolutionary computation. What makes Hilbert's problems so famous and unique? Hilbert designed his problems with the goal that "they could serve as examples for the kinds of problems the solutions of which would lead to advancements of disciplines in mathematics." If we have a closer look at Hilbert's twenty-three problems today, then we observe that some of the problems indeed lead to important research, but a few of them did not. One of the reasons seems to be how the problems have been formulated. Most of them are well defined, but some are more vaguely posed, making a solution difficult.

In fact, the paper became famous because of question number two: *Can it be proven that the axioms of arithmetic are consistent?* Hilbert's question is a sub-problem of the general research program Hilbert had in mind: *Can mathematics be axiomatized?* The general problem was taken on by Russel and

Whitehead and lead to three volumes of the *Principia Mathematica*. Gödel dealt with the more specific problem two and proved that the answer is negative. This put an end to the effort of Russel and Whitehead. The implication of Gödel's result with regard to mathematics and the theory of computation in general is still a subject of hot discussions.

In contrast, problem number six just reads: *Can physics be axiomatized?* In the explanation of the question Hilbert writes: "to axiomize those physical disciplines, in which mathematics already plays a dominant role; these are first and foremost probability and mechanics." To our surprise we see the calculus of probability as a part of physics! A closer inspection reveals that Hilbert's moderate goal was a mathematically sound application of probability to kinetic gas theory. This research has been carried out by physicists, but without ever referring to Hilbert or to a Hilbert problem. It lead to statistical physics as it appears today.

My goal is modest. I will propose problems, mainly in evolutionary computation, and name each after a famous scientist who has formulated or investigated the problem. This does not imply that the problem so named is the most important the scientist has worked on. Nor do I claim that the scientist has considered the problem to be the most important one he has worked on. I only want to demonstrate that most of the challenging problems have been identified very early and are with us for quite a time. And my second message is: we have to look much more into older papers. Older scientific papers should not be considered as "fossils". It is a fundamental misconception that science is continuously accumulating all the important available knowledge and condensing the knowledge in surveys or textbooks. Many important scientific ideas and papers enter main stream science after 20 or more years.

I will consider in the paper both – natural and artificial organisms. The emphasis will be on artificial automata. In order not just to summarize the problems, I will also describe in sections 11 till 13 a scientific method I consider as a promising candidate for solving some of the problems presented. It is the theory of probability, used and extended in scientific disciplines as different as *probabilistic logic, statistical physics, stochastic dynamical systems* and *function optimization using search distributions.* These sections will be fairly selfish, because in selecting from the huge available literature the work of my research group will be over-represented.

2. Evolutionary computation and theories of evolution

The goal of evolutionary computation is to make the development of powerful problem solving programs easier. There have been tried at least three approaches to achieve this goal.

1 **Use a theory** - develop a theory of problem solving and implement it on a computer

2 **Copy the brain** - analyze the human brain and make a copy of it on a computer

3 **Copy natural evolution** - analyze natural evolution and implement the most important evolutionary forces on a computer

In the history of artificial intelligence research one of the three approaches was dominant at any one time. Evolutionary computation belongs to the third approach. It relies on theories of evolution and of computation. The theory of computation is well advanced, so the problems of evolutionary computation lie in theories of evolution. If there existed a convincing constructive theory of evolution, then evolutionary computation would be just a matter of implementation - which of the major evolutionary forces to implement in what detail.

But do we possess a constructive theory of evolution? Here the opinions differ extremely. The main stream theory of evolution is called *New* or *Modern Synthesis*. Its followers claim that it reconciles Darwin's idea of continuous small variations with gene flows derived from population genetics. The second major force of the Modern Synthesis is still Darwin's concept of *natural selection*. But are these two forces sufficient to explain the wonders of evolution at least in some broad terms?

There is no doubt that the modern synthesis is able to explain the change of gene frequencies on a small time scale. *If there is enough diversification, then the theory correctly predicts further changes for a short time.* But can it explain the evolution for a long time? Here the crucial question is: How could it come to such a diversification, starting from a tiny cell? I like to formulate the problem with Darwin's famous ending sentence of *The Origin of Species by Means of Natural Selection* (Darwin, 1859).

"There is grandeur in this view of life, with its several powers, having been originally breathed into a few forms or into one; and that, whilst this planet has gone cycling on according to the fixed laws of gravity, from so simple a beginning endless forms most beautiful and most wonderful have been, and are being, evolved."

Let me be more specific and cite some major problems which a theory of evolution would have to explain. Maynard Smith and Szathmary have called them the *the major transitions in evolution* (see table 1.1, Smith and Szathmary, 1995).

The authors "solve" some of the problems with a very narrow version of the modern synthesis. "We are supporters of the gene centered approach proposed by Williams and refined by Dawkins, 1989." In the gene centered approach, also called the *selfish gene concept*, the genes are the major actors. They possess an internal force to proliferate as much as possible.

before	→	after
replicator molecules	→	population of molecules in compartments
independent replicator	→	chromosomes
RNA as gene and enzyme	→	DNA and protein
procaryote	→	eucaryote
asexual clones	→	sexual population
protist	→	plants, animals, fungi
solitary individuals	→	colonies
societies of primates	→	human societies

Table 1.1. Major transitions in evolution

This caricature of a theory of evolution is used by the authors to explain the transition from solitary individuals to colonies, for example. The argument is as follows: If a female produces two offspring, but n females can produce 3n offspring, then cooperation between the females pays off. Even if there is a fight between females and one becomes a queen, cooperation is still preferred (1/n of 3n is larger than 2). Thus in the gene centered analysis a colony with a single queen has a selective advantage.

There are many flaws in the selfish gene concept. It is not constructive, it does not investigate if the selection advantage of a particular gene can be realized in a phenotype. Rabbits with wings would obviously have a selective advantage. Why did it not happen? Two genes can also oppose each other - gene 1 might increase by action a_1, and gene 2 by the opposite action a_2. Which gene wins? Consider a female and its offspring as an example. The offspring are threatened. Should the mother protect the offspring, even on the risk of her life? The notorious formula of Hamilton gives the result that the mother should sacrify her life if more than two offspring are threatened (Smith and Szathmary, 1995). Hamilton argues as follows: in each offspring there are only one half of the genes of the mother. Thus the genes of the mother multiply if she protects at least three offspring.

Ironically Darwin itself has devoted a whole chapter of his "The Origin of Species" to the problem insect colonies pose to natural selection. His explanation is constructive. He shows how many small changes in behavior can lead to very peculiar behavior, even to slave making ants! This example shows dramatically the extreme simplification done by the selfish gene concept. It is my strong opinion that the selfish gene concept does not enrich Darwin's theory, but reduces it to a caricature.

The selfish gene concept has been opposed by a small group in biology, most notably the late Stephen J. Gould. Recently even philosophers of science formulate a basic critic. I just cite Griffiths, 2002. "The synthetic theory bypassed what were at the time intractable questions of the actual relationship between stretches of chromosomes and phenotypic traits. Although it was accepted

that genes must, in reality, generate phenotypic differences through interaction with other genes and other factors in development, genes were treated as *black boxes* that could be relied on to produce phenotypic variation with which they were known to correlate."

I will discuss this problem later with my proposal of a system theory of evolution. The major conclusion of this section is: there exists no convincing theory of evolution today. The "theory" its proponents call "Modern Synthesis" is an extremely simplified version of Darwin's theory. It separates organisms and environment. Natural selection is modeled by a *fitness function*, whereas Darwin used the term only in a metaphoric sense. In fact, Darwin noticed the misinterpretation of his theory even during his life. He wrote in the last (1872) edition of "The Origin of Species":"As my conclusions have lately been much misrepresented, and it has been stated that I attribute the modification of species exclusively to natural selection, I may be permitted to remark that in the first edition of this work, and subsequently, I placed in a most conspicuous position — namely at the close of the Introduction — the following words: "I am convinced that natural selection has been the main but not the exclusive means of modification." This has been of no avail. Great is the power of steady misinterpretation."

Therefore evolutionary computation has to be largely experimental. This was already pointed out by John von Neumann, 1954. "Natural organism are, as a rule, much more complicated and subtle, and therefore much less well understood in detail, than are artificial automata. Nevertheless, some regularities, which we observe in the organization of the former may be quite instructive in our thinking and planning of the latter; and conversely, a good deal of our experiences and difficulties with our artificial automata can be to some extend projected on our interpretations of natural organisms."

3. Darwin's continental cycle conjecture

I will describe my first problem in Darwin's terms. In the chapter "Circumstances favourable to Natural Selection" Darwin writes: "A large number of individuals by giving a better chance for the appearance within any given period of profitable variations, will compensate for a lesser amount of variability in each individual, and is, I believe, an extremely important element of success."

On the other hand Darwin observes that a large number of individuals in a large continental area will hinder the appearance of new adaptations. This happens more likely in small isolated areas. He writes: "Isolation, also, is an important element in the process of natural selection. In a confined or isolated area, if not large, the organic and inorganic conditions of life will be in a great degree uniform; so that natural selection will tend to modify all individuals

of a varying species throughout the area in the same manner in relation to the same conditions. But isolation probably acts more efficiently in checking the immigration of better adapted organisms. Lastly, isolation, by checking immigration and consequently competition, will give time for any new variety to be slowly improved."

Darwin then continues: "Hence an oceanic island at first sight seems to have been highly favourable for the production of new species." But Darwin notes a conflict: "to ascertain whether a small isolated area or a large open area like a continent, has been most favourable for the production of new organic forms, we ought to make the *comparison within equal times*; and this we are incapable of doing. "

Despite of the above observation Darwin concludes: "I conclude, that for terrestrial productions a large continental area, which will probably undergo many oscillations of level, and which consequently will exist for long periods in a broken condition, will be the most favourable for the production of many new forms of life, likely to endure long and spread widely." Darwin reasons as follows: "For the area will first have existed as a continent, and the inhabitants, at this period numerous in individuals and kinds, will have been subjected to very severe competition. When converted by subsidence into large separate islands, there will still exist many individuals of the same species on each island;... and time will be allowed for the varieties in each to become well modified and perfected. When by renewed elevation, the islands shall be re-converted into a continental area, there will be again severe competition: the most favoured or improved varieties will be enabled to spread: there will be much extinction of the less improved forms ..."

Problem 1 [Darwin]: *Can we demonstrate or even prove the correctness of Darwin's Continent-Island cycle conjecture?*

The reader should have observed how carefully Darwin discusses the arguments. I strongly recommend to read Darwin's "The Origin of Species". The most profound critique of modern "Darwinism" can be found in Darwin's book![1]

It seems difficult to test Darwin's conjecture in nature. I propose therefore to use simulations as first step. I have used the iterated prisonert's dilemma game to investigate problem 1 (Mühlenbein, 1991a). The results indicate that Darwin's conjecture might be correct. But the simulation model needs a lot more refinement.

Darwin mentions at many places of the "Origin" that space is as important for evolution as time. This has been shown in the context of genetic algorithms by Mühlenbein, 1991b. Space is also an important element of Wright's *shifting balance theory* of evolution Wright, 1937. Without referring to Darwin a subset of the problem, that is the difference of the evolution in a large con-

tinent and small isolated islands, has been recently investigated by Parisi and Ugolini, 2002.

4. The system view of evolution

The next set of problems I will derive more abstract. The major weakness of "Darwinism" in the form of the modern synthesis is the separation of the individuals and the environment. In the most simple model each individual O_i (mainly characterized by its genes) is assigned a fitness f predicting the performance of this individual within the environment E and given the other individuals. This can be written as:

$$\begin{aligned} O_i(t+1) &= f(\mathbf{O}, E(t)) \\ E(t+1) &= g(E(t) \end{aligned}$$

It seems impossible to obtain numerical values for the fitness. Therefore theoretical biology has made many simplifications. The environment is kept fixed, i.e $g(E(t)) = const$, the influence of other individuals is described by some averages of the population, etc.. The shortcomings of the dichotomy individual-environment in the Modern Synthesis have already been discussed. The problem is still more difficult because each individual is in addition *developing in a close interaction with its environment*.

The development problem has been addressed recently by Oyama, 2000, in her *developmental system theory*. Unfortunately the theory is very informal, it has been formulated from a philosopher's point of view. Therefore I will describe the next problem as it has been stated in the final address of Anatol Rapaport, the then retiring president of General System Science Society (Rapaport, 1970).

Problem 2 [Rapaport+1]:*Can we formulate a theory of organisms, which incorporates being, acting, evolving, and developing?*

Rapaport identified only three properties. He combined evolving and developing into a single property becoming. The problem needs an explanation. It goes back to Whitehead, 1948. In his book "Science and the Modern World" Whitehead warned that the store of fundamental ideas on which the then contemporary science was based was becoming depleted. Whitehead suggested that the concept of *organism*, hitherto neglected in physical science, might be a source of new ideas. Whitehead tried to define what an organism characterizes.

We will describe the definition of Rapaport. "According to a soft definition, a system is a portion of the world that is perceived as a unit and that is able to maintain its identity in spite of changes going on in it. An example of a system par excellence is a living organism. But a city, a nation, a business firm, a university are organisms of a sort. These systems are too complex

to be described in terms of succession of states or by mathematical methods. Nevertheless they can be subjected to methodological investigations."

Rapaport then defines: "Three fundamental properties of an organism appear in all organism-like systems. Each has a *structure*. That is, it consists of inter-related parts. It maintains a short-term steady state. That is to say, it reacts to changes in the environment in whatever way is required to maintain its integrity. It *functions.* It undergoes slow, long term changes. It grows, develops, or evolves. Or it degenerates, disintegrates, dies.

Organisms, ecological systems, nations, institutions, all have these three attributes: *structure, function, and history*, or, if you will, *being, acting, and becoming*."

Rapaport's becoming captures both – the development of an organism from the fertilized egg to the grown-up organism, and the evolution of the species in a succession of many generations. Despite its very intricate relationship, development and evolution have to be separated.

To my knowledge, Rapaport's talk did not lead to a scientific effort to build such a theory of organisms. The reader will guess the reason: it is the sheer complexity of the task! Instead research in biology remained concentrated on a single property or to a combination of two properties. Thus population genetics combines being and evolving, population dynamics combines being and acting. The developmental system theory mentioned earlier combines being and developing.

The investigation of question two leads to another problem: In what language should we frame a theory of organisms? Three approaches can be tried:

- The descriptive approach, using natural language
- The micro-simulation approach
- The mathematical approach

Today the descriptive approach has gained momentum, characterized by the developmental system theory mentioned above Oyama, 2000. Artificial Life uses micro-simulation. But in micro-simulations it is very difficult to distinguish between the microscopic event and the more general pattern happening in many simulations. Rapaport and, earlier, von Neumann advocated the mathematical approach. I go a step further and propose stochastic system theory as the research foundation. Whereas population genetics has been a stochastic theory for almost 75 years, population dynamics is still mainly investigated with the help of deterministic differential equations.
Thus I partition Rapaport's problem into three problems.

Problem 3a: *Can we develop a stochastic system theory, combining the properties being and acting of organisms or automata in a 2-d space?*

Problem 3b: *Can we develop a stochastic system theory, combining the properties being and developing of organisms or automata in a 2-d space?*

Problem 3c: *Can we develop a stochastic system theory, combining the aspects being, acting and evolving of organisms or automata in a 2-d space?*

The answer to the first question is a definite yes. It is already an active area of research. We will discuss it later in more detail in the context of cellular automata. The second problem seems to be much more difficult. Von Neumann was the first who worked on this problem for the case of automata.

5. Von Neumann's self-reproducing automata

Von Neumann started his research with the concept of *"complification"*. He used the term very informally. We proceed in the same way. It is outside the scope of this paper to discuss all the measures proposed for complexity. Also the term automaton will be used in a broad manner. Von Neumann observed: "If automaton A can produce B, then A in some way must have contained a complete description of B. In this sense some decrease in complexity must be expected as one automaton makes another automaton." But organisms reproduce themselves with no decrease in complexity. Moreover, organisms are indirectly derived from others which had lower complexity.

Problem 4 [von Neumann]: *Can we construct automata which are able to produce automata more complex than themselves?*

Von Neumann tried several approaches to enable a scientific investigation of the above problem. The main theory was collected by Burns and expended into a theory of *self-reproducing automata*Burns, 1970. But it is more instructive to look at von Neumann's own description, summarized in the article "The General and Logical Theory of Automata". von Neumann, 1954.

Von Neumann started his research with a result of Turing. Turing wanted to give a precise definition of what is meant by a computing automaton. His solution was the *Universal Turing Machine* UTM. It consists of an automaton reading and writing symbols on an infinite tape. Von Neumann decided that his automaton should have the power to simulate the UTM in a discrete cellular 2-d space. Thus he investigated the problem how to construct an automaton which reproduces itself in 2-d space and has the power of UTM.

Von Neumann's construction proceeded as follows von Neumann, 1954:

(a) *Construct an automaton A, which when furnished the description of any other automaton in terms of appropriate functions, will construct that entity.*

(b) *Construct an automaton B, which can make a copy of any instruction α that is furnished to it. This facility will be used when α furnishes a description of another automaton.*

(c) *Combine the automata A and B with a control mechanism γ, which does the following. γ will first cause A to construct the automaton which is de-*

scribed by α. Next γ will cause B to copy the instruction α. Finally γ will separate this construction from the system $A + B + \gamma = D$

(d) *Form an instruction α_D, which describes this automaton D, and insert α_D into A within D. Call the aggregate which now results E.*

E is clearly self-reproducing. But E cannot do anything besides reproduction. It needs a program. Therefore von Neumann proposed an extension: Replace the instruction α_D by an instruction α_{D+F} which describes automaton D plus another automaton F. This automaton reproduces itself and then behaves like automaton F. Now if a "mutation" within the F part takes place, it changes E_F into $E_{F'}$. This "mutant" is still self-reproductive.

Von Neumann believed that with this construction he had made crude steps in the direction of a systematic theory of automata, especially towards forming a rigorous concept of what constitutes "complication." At a first glance, the construction seems to be an solution of the automatic programming problem. But why did not von Neumann's self-reproducing automata have any practical relevance? The answer is simple: The construction does not solve the most important problem: How do the programs get into the machine? The *development of programs is the problem, not their self-reproduction.* Von Neumann's automata can in principle compute anything, but the programs have to be provided from the outside! Who provides these descriptions? A single built-in program F is surely not enough, because von Neumann did not introduce selection. Therefore the value of the mutant program F' for problem solving is not checked.
von Neumann solved only part of the problem. Therefore we extend problem 4.

Problem 5:*What conditions are required to enable von Neumann's automata to grow in complexity without external interventions?*

A worthwhile extension of von Neumann's approach would be to use a population of automata which interact with each other and have to solve a set of problems to survive and produce offspring. Thus I believe that for a solution of problem 5 one needs both, Turing and Darwin. Turing provides the concept of a universal automaton and Darwin provides the concept of a changing environment metaphorically leading to natural selection.

The importance of von Neumann's construction for today's research has also been emphasized by McMullin McMullin, 2001.

6. Turing's intelligent machine

Von Neumann's approach of using self-reproduction and the Universal Turing Machine was not the only method proposed to build intelligent machines. In fact, von Neumann discussed the usage of artificial neural networks as another possibility. Before I describe this work, it is instructive to discuss how

Turing himself approached the problem in his article "Computing machinery and intelligence"Turing, 1950. At first Turing defined the concept of intelligence. A machine is intelligent if it passes a test Turing defined precisely: the *Turing test* is an "imitation" game, played by three objects A, B and C. C is the interrogator, A or B might be a machine. The machine passes the test if the interrogator is not able to find out that a machine answers to his questions. This gives our next problem.

Problem 6 [Turing]: *Is it possible to create machines which pass the Turing test?*

Turing believed that the answer to the above question is positive and proposed a method to construct such a machine. It is described in the section "Learning Machines" of his article Turing, 1950. Turing's proposal seems to be almost unknown, although it is contained in this very famous article. I find it very fascinating. The arguments brought forward by Turing have been used a number of times in artificial intelligence research, but obviously without knowing that Turing already formulated them.

"As I have explained, the problem is mainly one of programming. Estimates of the storage capacity of the brain vary from 10^{10} and 10^{15}. I would be surprised if more than 10^{9} was required to satisfactory playing of the imitation game ... At my present rate I produce about a thousand digits of program a day, so that about sixty workers, working steadily through the fifty years might accomplish the job, if nothing went into the wastepaper basket. Some more expeditious method seems desirable."

Turing did not try to formalize a possible solution to problem 6. Any program passing the test will do. It is the *efficiency problem* which leads Turing to consider natural organisms, in this case the human mind. "In the process of trying to imitate an adult mind we are bound to think a good deal about the process which has brought it to the state that it is in. We may notice three components:

(a) The initial state of the mind, say at birth,
(b) The education to which it has been subjected,
(c) Other experience

Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's?.... We have thus divided our problem into two parts, the child programme and the education process. These two remain very closely connected. We cannot expect to find a good child machine at the first attempt... There is an obvious connection between this process and evolution, by the identifications

Structure of the child machine	=	hereditary material
Changes of the child machine	=	mutations
Natural selection	=	judgment of the experimenter

One may hope, however that this process will be more expeditious than evolution. The survival of the fittest is a slow method for measuring advantages. . . Opinions may vary as to the complexity which is suitable in the child machine. One might try to make it as simple as possible consistently with the general principles. Alternatively one might have a complete system of logical inference programmed in."

Turing reported: "*I have done some experiments with one such child machine, but the teaching method was too unorthodox for the experiment to be considered really successful.*"

The imitation game is the final test, one needs some intermediate goals. "We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with?. . . Many people think that a very abstract activity, like the playing of chess, would be the best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English."

Today chess playing has been solved by brute force programming. This solution is feasible due to the strict rules of chess that enable fast and efficient game tree search. The progress in games like *GO* is much slower. But we are still left with the language understanding problem.

Problem 7 [Turing]: *Is it possible to create a machine which can be taught to understand English?*

Turing's proposal belongs to the "copy the evolution" approach. But his evolution does not start with a cell, but with a well-designed child. Turing's approach is very informal, he believed that he could program an intelligent system using about 10^9 bits. I call this attitude the *programmer's approach*. The system is programmed without a theory. One just assumes that anything can be programmed. This attitude seems to be dominant today. For Turing evolution is just a technique to shorten the programming time.

7. What can be computed by an artificial neural network?

We now turn back to von Neumann and his approach to machine intelligence. In contrast to Turing, von Neumann works more like a natural scientist. He tries to formalize solution strategies. Thus his solutions are not finished programs, but theories.

In 1948 formal neural networks were already very popular in the research community because of the work of McCulloch and Pitts. Von John von Neumann, 1954, investigated the power of neural networks in his famous talk "The general and logical theory of automata". In the section "Formal Neural Networks" von Neumann notes: "The McCulloch-Pitts result[2] proves that anything that can be exhaustively and unambiguously described, anything that can be completely and unambiguously put into words, is ipso facto realizable by a suitable finite neural network... Thus the remaining problems are these two. First, if ascertain modes of behavior can be effected by a finite neural network, the question still remains whether the network can be realized within a practical size... Second, the question arises whether every existing modes of behavior can be put completely and unambiguously into words...

There is no doubt that any special phase of any conceivable form of behavior can be described completely and unambiguously in words. This description may be lengthy, but it is always possible... It is, however, an important limitation, that this applies only to every element separately, and it is far from clear how it will apply to the entire syndrome of behavior."

Von Neumann then discusses more specifically the concept of identification of analogous geometrical entities. He takes as example the concept of a triangle.

"There is no difficulty in describing how an organism might be able to identify any two rectilinear triangles, which appear on the retina, as belonging to the category "triangle". There is also no difficulty in adding to this, that numerous other objects, will also be classified and identified as triangles — triangles whose sides are curved, triangles whose sides are not full drawn ... This, in turn, however constitutes only a small fragment of the more general concept of *analogy*. Nobody would attempt to describe and define within any practical amount of space the general concept of analogy which dominates our interpretation of vision. There is no basis for saying whether such an enterprise would require thousands or millions or altogether impractical numbers of volumes. *Now it is perfectly possible that the simplest and only practical way actually to say what constitutes a visual analogy consist in giving a description of the connections of the visual brain*."

Problem 8 [von Neumann]: *Can an artificial neural network be designed which gives similar results on visual problems as the human brain ?*

Turing also used an "analysis" of the human brain in order to show that an intelligent machine can be programmed in 10^9 bits. He wrongly assumed that the performance of the brain can be characterized by its number of neurons, about 10^9. He did not consider the interconnection structure as relevant. The only problem left to him is to obtain this program of 10^9 digits. Von Neumann is much more careful. It is not the number of neurons which matters, but their interconnection structure. Today we know that even the interconnection structure is not sufficient to define uniquely how the neurons process the visual input. We need to know the *dynamic interaction of all the neurons involved*.

8. Limits of computing and common sense

I consider von Neumann's discussion about computability extremely important. The limitations of computing are given by the finiteness of the resources, in space and in time! Implicitly von Neumann points out that finiteness is not enough, we need reasonable time and reasonable space in our real world. The finiteness of our world puts an upper limit to the largest program which can be computed (this is von Neumann's length of the chain of reasoning mentioned earlier).

The theory of computability lead to the development of *complexity theory*. It is still a very lively research area, I just mention some basic results. One of the most important problem in computer science is the *Ptime versus NPtime* question: given a problem whose solution can be verified in polynomial time, is there an algorithm which actually finds such a solution (this means in polynomial time according to the size of the input.)? If both conditions can be proven, we have a problem from class **P**, if only the first condition is fulfilled we have an **NP** problem.

This basic classification has been refined in a number of ways. I just mention the inclusion

$$LOGTIME \subseteq LOGSPACE \subseteq PTIME \subseteq NPTIME \subseteq PSPACE \subseteq EXPTIME \subseteq E$$

Polynomial time means $T \approx O(n^k)$, exponential time $t \approx a^{f(n)}$. But if n is very large, even $O(n^k)$ can be a very large number, meaning that the problem cannot be computed in reasonable time. The largest meaningful value of T has been computed several times, using the finiteness of the universe and the laws of physics. One of the first to compute explicitly the upper limit was Bremerman Mьhlenbein, 1996.

Bremerman's bound: *No data processing system, whether artificial or living, can process more than* $2 * 10^{47}$ *bits per second per gram of its mass.*

Bremerman used this bound to calculate the total number of bits processed by a hypothetical computer the size of the earth within a time period equal to the estimated age of the earth. He computed 10^{93} bits. I shall call this number *Bremermann's limit.* Programs which are finite, but require more than 10^{93} steps for solving, are no solutions. This implies that the mathematical class of finite programs has to be divided into those below Bremerman's limit and above Bremerman's limit.

Von Neumann had serious doubts that complex behaviors like the concept of analogy can be described by a reasonable number of words, meaning that the description can be read and processed in a lifetime. Despite the warning issued by von Neumann there have been many attempts to put so much knowledge into a machine that it could behave intelligently. The earliest proposal was made in 1958 by McCarthy in his article "Programs with common sense"McCarthy, 1959[3]. The most recent effort was that by Lenat, 1995. With a team of up to 10 people he tried to code "common sense" knowledge into a rule-based database. After almost 10 years of effort, he was still far away from the goal, formulated as the next problem.

Problem 9 [McCarthy,Lenat]: *Is it possible to put so much knowledge into a computer, that it is able to read a newspaper and improve itself from thereon?*

Looking back to von Neumann's discussion, I believe that the answer to this question is negative. I do not recommend to work on this problem, because proving that something is impossible is very difficult. Instead I recommend a sub-problem, formulated by Shannon, 1953, in his paper "Computers and Automata".

Problem 10 [Shannon]: *Can we organize machines into a hierarchy of levels, as the brain appears to be organized, with the learning of the machine gradually progressing up through the hierarchy?*

Hierarchy is used by Shannon very informally. He means levels of abstractions. Each level might use a different calculus. The machine should be able to do inference on a lower level after a limited number of examples. This feature should then be used for learning at the next level. Up to now there are no convincing theories how to solve this problem.

9. A logical theory of adaptive systems

In the paper "Outline for a Logical Theory of Adaptive Systems" Holland, 1970b, tried to continue the scientific endeavor initiated by von Neumann. Holland wrote: "The theory should enable to *formulate key hypotheses and problems particularly from molecular control and neurophysiology. The work in theoretical genetics should find a natural place in the theory. At the same time, rigorous methods of automata theory, particularly those parts concerned with growing automata should be used.*"

Thus Holland's proposal is the first attempt to work on our problem 2. It tries to combine *being, acting, developing, and evolving.* This is so important that I will describe the proposal in detail. Holland's emphasis (like von Neumann's) is foremost on theories and systems, he does not claim to solve grand challenge applications with the proposed methods. This can be tried after the theories have been developed.

"Unrestricted adaptability (assuming nothing is known of the environment) requires that the adaptive system be able initially to *generate any of the programs of some universal computer*... With each generation procedure we associate the population of programs it generates;... In the same vein we can treat the environment as a population of problems." It is especially the last sentence which relates Holland's ideas to Darwin's.

Now let us have a closer look at Holland's model. First, there is a finite set of generators (programs) $(g_1, \ldots, g_k)$. The generation procedure is defined in terms of this set and a *graph* called a *generation tree.* Each permissible combination of generators is represented by a vertex in the generation tree. Holland now distinguishes between auxiliary vertices and main vertices. Each auxiliary vertex will be labeled with two numbers, called the *connections* and *disconnection probabilities.* This technique enables to create new connections or to delete existing connections. Each main vertex is labeled with a variable referred to as *density*. The interested reader is urged to read the original paper (Holland, 1970b).

Holland claims that from the generation tree and the transition equations of any particular generation procedure, one can *calculate the expected values of the densities of the main vertices as a function of time.* Holland writes: "From the general form of the transition equations one can determine such things as conditions under which the resulting gen-

eration procedures are *stationary processes.*" Thus Holland already tried to formulate a stochastic theory of program generation! This is an idea still waiting to be explored.

The above process is not yet adaptive. Adaptation needs an environment posing problems. Holland's extension is similar in spirit to von Neumann's self-reproducing automata. Holland introduces *supervisory programs* which can construct *templates* which alter the probabilities of connections. Templates play the role of catalysts or enzymes. Thus program construction is also influenced by some kind of "chemical reactions."

Holland further proposes that the *environment is treated as a population of problems*. These problems are presented by means of a finite set of initial statements and an algorithm for checking whether a purported solution of the problem is in fact a solution. "When we consider the interaction of an adaptive system with its environment we come very soon to questions of partial solutions, subgoals etc. The simplest cases occur when there is an a priori estimate of the nature of the partial solution and a measure of the closeness of its approach to the final solution."

Holland then observes that a *rich environment* is crucial for the adaptation. "Mathematical characterization of classes of rich environments relative to a given class of adaptive systems constitutes one of the major questions in the study of adaptive systems. . . . An adaptive system could enhance its rate of adaptation by somehow enriching the environment. Such enrichment occurs if the adaptive system can generate subproblems or subgoals whose solution will contribute to the solution of the given problems of the environment."

It is very interesting to note that Holland distinguished three kinds of programs – supervisory programs, templates, and programs for the problem solution. The supervisory programs use a probabilistic generation tree to generate programs, the templates are used as catalyst to "skew" the generation process. Holland perceived a hierarchy of programs Holland, 1970a:

1 productive systems – the generator system is able to produce other generators

2 autocatalytic systems – the generator system produces generators which are used in the construction

3 self-duplicating systems – the generator system produces duplicates of itself

4 general adaptive systems – has still to be defined

"The beginning of such a definition (of adaptive systems) lies in the following consideration: with the help of concepts such as autocatalytic and self-duplicating generator systems it is possible to define such concepts as steady-state equilibria and homeostasis for embedded automata. . . If the generator system for such an automaton has a hierarchical structure, then a small change in structure produces a small change in proportion to the "position" of the change in the hierarchy. . . By making changes first at the highest level and then at progressively lower levels of the hierarchy, it should be possible to narrow down rather quickly to any automaton in this category having some initially prescribed behavior."

I believe that Holland's very first proposal is a very good starting point for future research. It puts forward many ideas not yet contained in current research. Holland's proposal to use stochastic systems, their steady-state equilibria and homeostasis is in my opinion still a very promising approach for solving difficult problems by evolutionary computation. But as it often happens in science, understanding these concepts in a solid theory is more difficult than anticipated. I will discuss this approach with simpler models in the next sections. Holland itself never implemented his general model. Therefore the next problem is still open.

Problem 11 [Holland]: *Try to implement Holland's model and prove its usability by a convincing application.*

Holland never implemented the proposed system. After working about eight years on this theory he turned to a simpler evolution model, in fact the Modern Synthesis mentioned before. The environment is hidden in a *fitness function*. Evolution reduces then to an optimization problem. This research lead to *genetic algorithms*. Holland believed that his genetic algorithms have an almost optimal adaptation rate taking into account the information which is available (Holland, 1973; Holland, 1992). But we will prove in Section 13 that it is our Boltzmann distribution algorithm which fulfills his criterion for optimality!

Nobel laureate Gell-Man criticized at the Santa Fe institute, that genetic algorithms are unsuited to investigate self-organized evolution, because they use a simple fitness function for a genotype. Therefore Holland, 1992, later developed *Echo*. Unfortunately *Echo* lacks the theoretical foundation of Holland's first proposal. Therefore I will not discuss it in this paper.

10. The λ-Calculus for creating artificial intelligence

In another chain of reasoning we might ask ourselves: Maybe there is a way of creating human like intelligence without copying nature too much. Instead of starting with the Universal Turing Machine, we can start with the calculus developed by Church and later called the λ-calculus. It was implemented as part of the LISP language by John McCarthy. The λ-calculus has the same computational power as the Turing machine, but it is based on substitution. LISP is an interpretative language, thus the LISP environment can be seen as a very complex self-reproducing automaton.

For the next problem I recommend to read Minsky's survey "Steps toward artificial intelligence" (Minsky, 1961). I only cite: "It is my conviction that no scheme for learning, or for pattern recognition, can have very general utility unless there are provisions for recursive, or at least hierarchical, use of previous results. We cannot expect a learning system to come to handle very hard problems without preparing it with a reasonable graded sequence of problems of growing difficulty. The first problem must be one which can be solved in reasonable time with the initial resources. The next must be capable of solution in reasonable time by using reasonably simple and accessible combinations of methods developed in the first, and so on."

In my opinion we have even to go a step further. There seems to be no big gain if the set of problems is hand crafted by a human. The program itself should create some of the sub-problems. We now have to formulate a task for this model. I rephrase a question from Shannon, 1953

Problem 12 [Shannon]: *Can we program a digital computer so that eventually 99 percent of the orders it follows are written by the computer itself and which solves difficult problems (e.g performs comparable to the human eye or understands the English language?)*

I added the two applications in brackets, because Shannon forgot in his question to specify the applications to be solved. But without an application the above problem can easily be solved by a program which randomly generates instructions.

LISP was the first language used by Koza for *Genetic Programming*. But within the framework of our discussion, Koza's model is too restricted. It works only for one problem at a time. For each problem we need examples describing the input-output relations of the problem

to be solved. The population of solutions is changed according to the mechanisms used by genetic algorithms.

11. Probabilistic logic

All problems up to now have been formulated in the very early days of electronic computers. For the early researcher a possible solution of each of these problems was either a theory or a successful application in pattern recognition or language understanding.

Furthermore, in order to develop and understand the model, either *classical mathematics* or *abstract automata* defined by a flexible language have been used. Several times *stochastic systems* have been proposed for the mathematical analysis.

Von Neumann explicitly expressed the feeling, having in mind artificial automata as model organisms, that a new theory is urgently needed (von Neumann, 1954): "This new system of formal logic will move closer to another discipline which has been little linked in the past with logic. This is *thermodynamics*, primarily in the form it was received from Boltzmann, and is that part of theoretical physics which comes nearest in some of its aspects to manipulating and measuring information. Its techniques are much more analytical than combinatorial."

Von Neumann's prediction has become true. Probability has been extended to *probabilistic logic* (Jaynes, 1957).

11.1 Von Neumann's probabilistic logics

To my knowledge von Neumann was the first to use the term probabilistic logic in his paper "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components"von Neumann, 1956. I shortly describe his model.

"With every basic organ is associated a number ϵ such that in any operation the organ will fail to function correctly ... Suppose the organ receives a stimulation at time t and no later ones. Let the probability that the organ is still excited after s cycles be denoted by p_s. Then the recursion formula

$$p_{s+1} = (1-\epsilon)p_s + \epsilon(1-p_s)$$

is valid." It is easy to show that the equation has the solution

$$p_{s+1} = (1-2\epsilon)^{s+1}(p_0 - \frac{1}{2}) + \frac{1}{2}$$

Therefore von Neumann concludes that $p_s \to 0.5$ for $s \to \infty$, meaning in von Neumann's opinion that the component functions randomly. But let us now investigate the problem in a precisely defined automaton setting. The automaton has two states $\{0, 1\}$. At each step the automaton changes with probability ϵ from the given state to the opposite state. If we observe the automaton, we would see that the automaton changes states only after ϵ^{-1} steps on the average. Such a behavior is very different from that of a random automaton, which changes states at each step with probability 0.5. But both automata have a *limit distribution* with $p_s \approx 0.5$. The difference between the distributions becomes apparent if higher order marginal distributions like $p(x_s, x_{s-1})$ are considered, where x_s denotes the state of the automaton at step s.

Von Neumann's analysis did not capture the reliability problem. Therefore his "solution" to the problem of unreliable components did not have any practical value. Von Neumann approached probabilistic logic from the most difficult point of view, namely the stochastic view. This means to define logic with time dependent dynamics! It is much easier to define probabilistic logic from the logic point of view, without time and dynamics. This is discussed next.

11.2 The conditional probability computer

As early as 1956 Uttley proposed the *conditional probability computer*. It consists of n binary input units $\mathbf{x} = (x_1, \ldots, x_n)$. Internally *all possible* conditional probabilities $p(\mathbf{y}|\mathbf{z})$ are computed, where $\mathbf{y}$ and $\mathbf{z}$ are disjoint sub-vectors of $\mathbf{x}$. The relation between probability theory and logic is simple, but fundamental: identify a conditional probability expression with a clause in propositional calculus. For simplicity let $\mathbf{z}$ and $\mathbf{y}$ denote terminal symbols.

DEFINITION 1.1 *Let* $0 \leq p(\mathbf{x}) \leq 1$ *denote the probability of* $\mathbf{x}$ *Then* $p_i(x_k, t) = \sum_{\mathbf{x}, X_i = x_k} p(\mathbf{x}, t)$ *defines the univariate marginal distributions of variable* X_i. *Let* $\mathbf{x}_\xi$ *be a sub-vector of* $\mathbf{x}$. *Then the* **marginal distribution** *is defined as* $p(\mathbf{x}_\xi, t) = \sum_{\mathbf{x}, X_\xi = x_\xi} p(\mathbf{x}, t)$ *Let* $\mathbf{y}, \mathbf{z}$ *be disjoint sub-vectors of* $\mathbf{x}$. *Then* **conditional probabilities** *are defined as* $p(\mathbf{y}|\mathbf{z}) = p(\mathbf{y}, \mathbf{z})/p(\mathbf{z})$ *for* $p(\mathbf{z}) > 0$.

A probabilistic statement that $\mathbf{z}$ *is true given* $\mathbf{y}$ *is a conditional probability with "truth" value* $0 \leq r \leq 1$

$$p(\mathbf{z}|\mathbf{y}) = r$$

As Uttley observed, a conditional probability computer would allow to compute all logical inferences, if we identify "from $\mathbf{y}$ follows $\mathbf{z}$" by the condition $p(\mathbf{z}|\mathbf{y}) > 0.5$. The drawback of this proposal is that it needs $2^n - 1$ units and also exponential space. There have been several attempts to use less units and also to deal with the case of *incomplete input*. Most notably are the efforts of Minsky and Selfridge, and independently by Papert (both papers published in Cherry, 1961. In both papers the assumption is made that all x_i's are independent. This is very unrealistic. It needed a long time to solve this problem.

11.3 Modern probabilistic logic

Modern probabilistic logics can be seen as a candidate for von Neumann's new system of formal logic. It connects probability theory with logic by assigning probabilities to clauses, e.g. $p(\mathbf{z}|\mathbf{y}) = 0.6$. Let n be the number of binary concepts. In addition let a number of clauses be specified. The specifications are called the *constraints*.

For any specification we have a set of probability models (P-models) which can either be empty (i.e the constraints violate the laws of probability), contain a single P-model, or contain a number of P-models (the specification is *incomplete*.) If the P-model is unique, we can compute the probability of an arbitrary propositional sentence by summing up probabilities. The probability of a conditional statement $p(\mathbf{z}|\mathbf{y})$ can be obtained by dividing the probability $p(\mathbf{z}, \mathbf{y})$ by the probability $p(\mathbf{y})$.

But unique P-models are unrealistic. The specification has to set all of the $2^n - 1$ variables defining the distribution. Consequently, for incomplete specifications the missing information must be added by some automatic completion procedure. This is achieved by the *maximum entropy principle*. The entropy of a distribution is defined by

$$H(p) = -\sum_x p(\mathbf{x}) \ln(p(\mathbf{x})) \tag{1.1}$$

The maximum entropy principle formulates the *principle of indifference.* If no constraints are specified, the uniform random distribution is assumed.

Maximum entropy principle: *Find the maximal entropy distribution for $p(\mathbf{x})$ which satisfies the given marginals.*

This principle has a long history in physics and probabilistic logic. The interested reader is referred to Jaynes, 1957. The following theorem holds (Cover and Thomas, 1989).

THEOREM 1.2 *If the given constraints are consistent, then there exists a unique distribution $q(\mathbf{x})$ of maximum entropy.*

Consistent means that the marginal distributions derived from the constraints fulfill all the constraints which can be derived from the laws of probability theory. This means the constraints should not contradict each other. The most popular algorithm to compute the maximum entropy distribution is called *iterative proportional fitting*. To give the reader a flavor of the theory we present a simple example.
Example: Given the three expressions 'having a full-time job' ft, 'working in a technical domain' t and 'male' m, the following information is specified

$$\begin{aligned} P(t|ft) &= 0.55 \\ P(t|m) &= 0.55 \\ P(t|ft \cup m) &= 0.45 \end{aligned}$$

Then the maximum entropy solution gives, for instance $P(t|ft \cap m) \approx 0.84$. [4]

The maximum entropy principle solves the incomplete data problem. But unfortunately iterative proportional fitting scales exponentially in the number of variables. Thus a simpler technique has to be found. Such a method has recently been discovered. It uses the principle of *conditional independence*. Its graphical representation is called a *graphical model*. For our discussion the following definition is sufficient.

DEFINITION 1.3 *A graphical model is a graph G, where two variables are connected by an edge if they appear together in one constraint.*

The new method tries to find a factorization of the distribution. There is lots of literature available how this can be done, we just mention Lauritzen, 1996. The algorithm computes *cliques* and generates a *junction tree* J. A junction tree is an undirected tree the nodes of which are clusters of variables. The clusters satisfy the *junction property*: For any two clusters a and b and any cluster h on the unique path between a and b in the junction tree the relation

$$a \cap b \subseteq h \tag{1.2}$$

is true. The edges between the clusters are labeled with the intersection of the adjacent clusters; we call these labels *separating sets* or *separators*.

The modified iterative proportional fitting algorithm uses only the computed clusters of the factorization as marginals. This algorithm produces exactly the same result as the standard iterative proportional fitting. If all factors of the factorization have a number of variables which is independent of the global number n, then the algorithm is polynomial.

The crucial question remains: Which graphical models lead to bounded factorizations? We give here just one negative result Mühlenbein and Mahnig, 2003:

THEOREM 1.4 *Graphical model models which are 2-D grids lead to factorizations which have at least one factor with $\sqrt{n}$ variables. Thus for these problems the computational amount to compute the maximum entropy distribution is still exponential.*

12. Stochastic analysis of cellular automata

Another new application of stochastic systems and probabilistic logic are cellular automata. The stochastic analysis of cellular automata was already advocated by Wolfram, 1994, in his paper "Twenty Problems in the Theory of Cellular Automata". The next problem combines Wolfram's problems ten and eleven.

Problem 13 [Wolfram]: *What is the correspondence between cellular automata and stochastic systems, and how are cellular automata affected by noise and other imperfections?*

We have worked on this problem. In order to provide the reader with more detailed information, I will discuss a simple example. It is taken from (Mühlenbein and Höns, 2002).

12.1 The nonlinear voter model

We consider a model of two species (or two opinions). For the spatial distribution we assume a one-dimensional stochastic cellular automaton (SCA) defined by a circle of n cells. Each cell is occupied by one individual, thus each cell is characterized by a discrete value $\sigma_i \in \{0, 1\}$. We set $x_{n+1} := x_1$ and $x_0 := x_n$. The state of cell x_i at time $t + 1$ is defined by the states of cells x_{i-1}, x_i, x_{i+1} at time t. The state transitions of the voter model depend only on $k(t) = \sigma_{i-1}(t)+\sigma_i(t)+\sigma_{i+1}(t)$. This class of automata is called *totalistic*. For the stochastic voter model the transitions are defined as follows.

$k(t)$	$p(\sigma_i(t+1)=1\|k(t))$
3	$1-\epsilon$
2	$1-\alpha$
1	α
0	ϵ

$p(\sigma_i = 1|k(t)))$ denotes the transition probability given k. ϵ is a small stochastic disturbance parameter. The model is defined by α. If $\alpha < 0.5$ one speaks of positive frequency dependent invasion. This model is also called the *majority vote model*, because the individuals join the opinion of the majority in the neighborhood. For $\alpha > 0.5$ the model is called a negative frequency dependent invasion process. In this case the minority opinion has more weight. The deterministic cellular automata are given by $\epsilon = 0$ and $\alpha = 0, 1$. The voter model has been intensively investigated by micro simulations.

We will first analyze the voter model by the theory of Markov chains. Let $\mathbf{x} = (x_1, \ldots, x_n)$ denote a vector, $x_i \in \Lambda_i = \{0, 1, 2, \ldots, m_i\}$. We use the following conventions. Capital letters X_i denote the names of variables, lower case letters x_i assignments. The *distinction between the name of a variable and an assignment* is essential for the definition of marginal distributions. When there cannot be a confusion between name or assignment, we will use lower case letters and abbreviations. For notational simplicity we will assume *binary variables* $x_i \in \{0, 1\}$. Important definitions will be given for the general case.

The time evolution of the distribution is given for one step by the equation

$$p(\mathbf{x}, t+1) = \sum_{x'} p(\mathbf{x}, t+1|\mathbf{x}', t)p(\mathbf{x}', t) \tag{1.3}$$

$M(t) = (p(\mathbf{x}, t+1|\mathbf{x}', t))$ defines a $2^n \times 2^n$ matrix.

DEFINITION 1.5 *The stochastic process is a Markov process if $M(t)$ is independent of t.*

The stochastic voter model is a Markov process. For a Markov process we have

$$p(\mathbf{x}, t) = M^t p(\mathbf{x}, 0) \tag{1.4}$$

For $0 < \epsilon, \alpha < 1$ we have $p(\mathbf{x}|\mathbf{x}') > 0$. Therefore the *theorem of Frobenius-Perron* can be applied. The largest eigenvalue of the matrix is 1. Its unique eigenvector defines the stationary distribution. Thus we have the following theorem.

THEOREM 1.6 *The stochastic voter model with $\epsilon > 0$ has a unique limit distribution. It is given by the left eigenvector belonging to the eigenvalue $\lambda_1 = 1$.*

It is numerically impossible to analyze a large cellular automaton by standard Markov techniques. It takes an exponential amount of computation to compute the exact stationary distribution.

We propose a different approach. We approximate the distribution $p(\mathbf{x}, t)$ by distributions using a small number of parameters. For this approximation we use the theory of graphical models mentioned before.

12.2 Stochastic analysis of one dimensional SCA

For notational convenience we set $\theta_i := x_i(t+1)$, and $\sigma_i := x_i(t)$. We will now derive difference equations involving marginal distributions with a few number of parameters. We obtain from the definition of the voter model for the von Neumann neighborhood in 1-D

$$p(\theta_i) = \sum_{\sigma_{i-1},\sigma_i,\sigma_{i+1}} p(\theta_i|\sigma_{i-1},\sigma_i,\sigma_{i+1})p(\sigma_{i-1},\sigma_i,\sigma_{i+1}) \tag{1.5}$$

$p(\theta_i)$ gives the probability of cell i containing a 1. The conditional distribution $p(\theta_i|\sigma_{i-1},\sigma_i,\sigma_{i+1})$ is uniquely defined by the transitions of the cellular automaton, in our case by the voter model with parameters ϵ and α. But on the right side tri-variate marginals appear. For these we obtain

$$\begin{aligned} p(\theta_{i-1},\theta_i,\theta_{i+1}) \quad = \quad & \sum_{\sigma_{i-2},\sigma_{i-1},\sigma_i,\sigma_{i+1},\sigma_{i+2}} p(\theta_{i-1},\theta_i,\theta_{i+1}|\sigma_{i-2},\ldots,\sigma_{i+2}) \\ & p(\sigma_{i-2},\sigma_{i-1},\sigma_i,\sigma_{i+1},\sigma_{i+2}) \end{aligned} \tag{1.6}$$

Thus now marginal distribution of size 5 enter. In order to stop this expansion we approximate the marginal distributions of order 5 by marginal distributions of order 3. From the definition of the SCA we obtain

$$\begin{aligned} p(\theta_{i-1},\theta_i,\theta_{i+1}|\sigma_{i-2},\sigma_{i-1},\sigma_i,\sigma_{i+1},\sigma_{i+2}) = p(\theta_{i-1}|\sigma_{i-2},\sigma_{i-1},\sigma_i) \\ p(\theta_i|\sigma_{i-1},\sigma_i,\sigma_{i+1})p(\theta_{i+1}|\sigma_i,\sigma_{i+1},\sigma_{i+2}) \end{aligned} \tag{1.7}$$

From the theory of graphical models we obtain the approximation

$$p(\sigma_{i-2},\ldots,\sigma_{i+2}) \approx p(\sigma_{i-1},\sigma_i,\sigma_{i+1})p(\sigma_{i-2}|\sigma_{i-1},\sigma_i)p(\sigma_{i+2}|\sigma_i,\sigma_{i+1}) \tag{1.8}$$

Inserting the last two equations into equation (1.6) gives the difference equations for the *tri-variate marginal distributions*. The approximations have to fulfill constraints derived from probability theory.

$$\begin{aligned} \sum_{\sigma_{i-1},\sigma_i,\sigma_{i+1}} p(\sigma_{i-1},\sigma_i,\sigma_{i+1}) &= 1 \\ \sum_{\sigma_{i-1}} p(\sigma_{i-1},\sigma_i,\sigma_{i+1}) &= \sum_{\sigma_{i+2}} p(\sigma_i,\sigma_{i+1},\sigma_{i+2}) \end{aligned}$$

In the same manner approximations of different precision can be obtained. We just discuss the simplest approximation, using *uni-variate marginal distributions*. Here equation (1.5) is approximated by

$$p(\theta_i) \approx \sum_{\sigma_{i-1},\sigma_i,\sigma_{i+1}} p(\theta_i|\sigma_{i-1},\sigma_i,\sigma_{i+1})p(\sigma_{i-1})p(\sigma_i)p(\sigma_{i+1}) \qquad (1.9)$$

The approximation by univariate marginal distributions leads to n difference equations only, but these difference equations are nonlinear. It seems very unlikely that analytical solutions of these equations can be obtained. For *spatially homogeneous* problems we have $p(\theta_i) = p(\theta_{i+1})$. In this case the probabilities do not depend on the locus of the cell. This is the *mean-field limit* known from statistical physics Opper and Saad, 2001. With $x(t) = 1/n \sum_i p(x_i = 1, t)$ we obtain the *mean-field equation*

$$x(t+1) = (1-\epsilon)x(t)^3+\epsilon(1-x(t))^3+3(1-\alpha)x(t)^2(1-x(t))+3\alpha x(t)(1-x(t))^2 \qquad (1.10)$$

For $\epsilon \approx 0$ and $\alpha < 1/3$ the equation has stable fix-points at $x \approx 0$ and $x \approx 1$. For $\alpha > 1/3$ the equation has a stable attractors at $x \approx 0.5$. Thus the mean-field limit approximation indicates a *bifurcation* for $\alpha = 1/3$. This interpretation is tempting, but in reality the relation between the fix-points of equation 1.10 is very complicated Mɯhlenbein and Hɵns, 2002.

The approximation of 2-D spatial distributions is much more complicated than the approximation of 1-D automata. Here the junction tree algorithm is needed. The interested reader is referred to Mɯhlenbein and Hɵns, 2002.

13. Stochastic analysis of evolutionary algorithms

The broad applicability of the new developments in probability theory can be demonstrated by another example, namely *evolutionary algo-*

rithms Mühlenbein et al., 1999; Mühlenbein and Mahnig, 2000; Mühlenbein and Mahnig, 2002a. This application is easier than the analysis of cellular automata. The distribution remains focused because of selection.

Let a function $f : X \to \mathbb{R}_{\geq 0}$ be given. We consider the optimization problem

$$\mathbf{x}_{opt} = \operatorname{argmax} f(\mathbf{x}) \tag{1.11}$$

For the solution Holland proposed in 1973 an algorithm called *genetic algorithm* Holland, 1992. The following discussion is taken from Mühlenbein and Mahnig, 2003.

Genetic algorithms are defined on a microscopic level. Given two strings, a new point is generated by recombination/crossover. A stochastic analysis of a genetic algorithm requires the computation of a recurrence equation

$$p(\mathbf{x}, t+1) = \sum_{\mathbf{x}'} p(\mathbf{x}|\mathbf{x}', t) p(\mathbf{x}', t) \tag{1.12}$$

Here $p(\mathbf{x}|\mathbf{x}', t)$ denotes the probability for a transition from $\mathbf{x}'$ to $\mathbf{x}$ at generation t. Because of selection the transition probabilities are time dependent. Vose Vose, 1999 has derived such an equation for the Simple Genetic Algorithm with proportionate selection, crossover, and mutation. The computation of the crossover probabilities are especially difficult. Since crossover operates on two arbitrary strings $\mathbf{x}$ and $\mathbf{y}$ of the selected population, one has to use the joint distribution $p(\mathbf{x}; \mathbf{y})$ in equation (1.12). But even for the binary case, the transfer matrix $p(\mathbf{x}|\mathbf{x}')$ is of size $2^n \times 2^n$. It is extremely difficult to analyze the distribution using this general equation.

Remark: *Marginal distributions define schemata*

For the researchers working on the theory of genetic algorithm it is important to mention that marginal distributions are equivalent to *schema* probabilities introduced in Holland, 1992. We just give an example for $n = 5$. Let $\xi = (1, 0, *, *, *)$ define a schema. Then the probability of the instances of schema ξ in the population $P(t)$ is by definition equal to the marginal distribution $p(X_1 = 1, X_2 = 0, t)$. Thus Holland's schema analysis is nothing else than a mesoscopic analysis in the space of marginal distributions. We prefer to use the notation common in probability theory. In fact, one of the main reasons that schema theory did not come very far is the imprecise terminology. In our mesoscopic analysis conditional probabilities play an essential role. But the

concept of conditional schema probabilities has not yet entered the traditional schema theory.

But let us proceed further. Equation (1.12) should not be the end result of a mesoscopic analysis, but just the beginning. We will concentrate on distributions which are defined by a small number of parameters or can be approximated by distributions with a small set of parameters. Since we treat the marginal distributions as *deterministic* variables, the mesoscopic analysis is valid for *infinite* populations only. Fluctuations arising by virtue of finite populations can be investigated in principle, but it is extremely difficult. Due to of the sampling theory in statistics our analysis can be seen as the limit case of large finite populations where the size goes to infinity.

A good candidate for optimization using a search distribution is the Boltzmann distribution.

DEFINITION 1.7 *For $\beta \geq 0$ define the* Boltzmann distribution *of a function $f(x)$ as*

$$p_\beta(x) := \frac{e^{\beta f(x)}}{\sum_y e^{\beta f(y)}} =: \frac{e^{\beta f(x)}}{Z_f(\beta)} \tag{1.13}$$

where $Z_f(\beta)$ is the partition function. To simplify the notation β and/or f can be omitted.

The Boltzmann distribution is usually defined as $e^{-\frac{g(x)}{T}}/Z$. The term $g(x)$ is called the energy and $T = 1/\beta$ the temperature. The Boltzmann distribution is suited for optimization because it concentrates with increasing β around the global optima of the function. In theory, if it were possible to sample efficiently from this distribution for arbitrary β, optimization would be an easy task.

13.1 Boltzmann selection

Our proposed algorithm incrementally computes the Boltzmann distribution by using Boltzmann selection.

DEFINITION 1.8 *Given a distribution p and a selection parameter $\Delta\beta$,* Boltzmann selection *calculates the distribution of the selected points according to*

$$p^s(x) = \frac{p(x)e^{\Delta\beta f(x)}}{\sum_y p(y)e^{\Delta\beta f(y)}} \tag{1.14}$$

Algorithm 1: *BEDA* – Boltzmann Estimated Distribution Algorithm

1 $t \Leftarrow 1$. Generate N points according to the uniform distribution $p(x, 0)$ with $\beta(0) = 0$.
2 **do** $\{$
3 With a given $\Delta\beta(t) > 0$, let
$$p^s(x,t) = \frac{p(x,t)e^{\Delta\beta(t)f(x)}}{\sum_y p(y,t)e^{\Delta\beta(t)f(y)}}.$$
4 Generate N new points according to the distribution $p(x, t+1) = p^s(x, t)$.
5 $t \Leftarrow t + 1$.
6 $\}$ **until** (stopping criterion reached)

We can now define the *BEDA* (Boltzmann Estimated Distribution Algorithm). It can easily be proven that *BEDA* converges to the set of all global optima if $\sum_t \Delta(\beta(t)) \rightarrow \infty$ Mühlenbein and Mahnig, 2002b. *BEDA* is a conceptional algorithm, because the calculation of the distribution requires a sum over exponentially many terms. We next transform *BEDA* into a practical algorithm. This means to reduce the number of parameters of the distribution and to compute an adaptive schedule for β.

13.2 Factorization of the distribution

In this section the factorization method introduced for graphical models is applied.

DEFINITION 1.9 *Let $s_1, \ldots, s_m$ be index sets, $s_i \subseteq \{1, \ldots, n\}$. Let f_i be functions depending only on the variables x_j with $j \in s_i$. Then*

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x}_{s_i}) \tag{1.15}$$

is an additive decomposition *of the fitness function f.*

From the additive decomposition we construct a *graphical model* by connecting those variables which are contained in the same subfunction. This definition is identical to the graphical model earlier introduced in probabilistic logic.
We also need the following definitions

DEFINITION 1.10 *Given $s_1, \ldots, s_m$, we define for $i = 1, \ldots, m$ the sets d_i, b_i and c_i:*

$$d_i := \bigcup_{j=1}^{i} s_j, \qquad b_i := s_i \setminus d_{i-1}, \qquad c_i := s_i \cap d_{i-1} \tag{1.16}$$

We set $d_0 = \emptyset$.

In the theory of decomposable graphs, d_i are called *histories*, b_i *residuals* and c_i *separators* Lauritzen, 1996. In Mühlenbein et al., 1999 we have proven the following theorem.

THEOREM 1.11 (FACTORIZATION THEOREM) *Let $p_\beta(\mathbf{x})$ be a Boltzmann distribution with*

$$p_\beta(\mathbf{x}) = \frac{e^{\beta f(\mathbf{x})}}{Z_f(\beta)} \tag{1.17}$$

and $f(\mathbf{x}) = \sum_{i=1}^{m} f_{s_i}(\mathbf{x})$ be an additive decomposition. If

$$b_i \neq \emptyset \quad \forall i = 1, \ldots, m; \quad d_m = \{x_1, \ldots, x_n\}, \tag{1.18}$$

$$\forall i \geq 2 \; \exists j < i \;\; \textit{such that } c_i \subseteq s_j \tag{1.19}$$

then

$$p_\beta(\mathbf{x}) = \prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i} | \mathbf{x}_{c_i}) = \frac{\prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})}{\prod_{i=2}^{m} p_\beta(\mathbf{x}_{c_i})} \tag{1.20}$$

The constraint defined as equation (1.19) is called the *running intersection property*. This severe assumption is identical to the junction property defined in equation (1.2).

The factorization theorem can be seen as a mathematically complete *schema theorem*. It tells which schemata are necessary to generate the *whole* distribution. The usual schema theorems describe only the evolution of schemata, but not how the distribution can be generated.

With the help of the factorization theorem, we can turn the conceptional algorithm *BEDA* into *FDA*, the Factorized Distribution Algorithm. If the conditions of the factorization theorem are fulfilled, the convergence proof of *BEDA* is valid for *FDA* also. *FDA* can in principle be used with any selection scheme, but then the convergence proof is no longer valid. Therefore we believe that Boltzmann selection is an essential part in using the *FDA*.

Algorithm 2: *FDA* – Factorized Distribution Algorithm

1. Calculate b_i and c_i from the decomposition of the function.
2. $t \Leftarrow 1$. Generate an initial population with N individuals from the uniform distribution.
3. **do** $\{$
4. Select $M \leq N$ individuals using Boltzmann selection.
5. Estimate the conditional probabilities $p(x_{b_i}|x_{c_i}, t)$ from the selected points.
6. Generate new points according to $p(x, t+1) = \prod_{i=1}^{m} p(x_{b_i}|x_{c_i}, t)$.
7. $t \Leftarrow t + 1$.
8. $\}$ **until** (stopping criterion reached)

Since *FDA* uses finite samples of points to estimate the conditional probabilities, convergence to the optimum will depend on the size of the samples (the population size). *FDA* has experimentally proven to be very successful on a number of functions where standard genetic algorithms fail to find the global optimum. In Mühlenbein and Mahnig, 1999 the scaling behavior for various test functions has been studied. For recent surveys the reader is referred to Mühlenbein and Mahnig, 2002a; Mühlenbein and Mahnig, 2003.

13.3 Holland's schema analysis and the Boltzmann distribution

We now turn to the very first analysis of genetic algorithms made by Holland Holland, 1992. We will use here Holland's terminology. (We remind the reader that ξ defines a schema and $P(\xi, t)$ its probability. This is in our notation the marginal distribution $p(\mathbf{x}_\xi, t)$.) He derived the following conjecture about a good population based search algorithm.

(Holland, 1992,p.88): *Each (schema) ξ represented in (the current population) $B(t)$ should increase (or decrease) in a rate proportional to its "observed" "usefulness" $\hat{\mu_\xi}(t) - \hat{\mu}(t)$ (average fitness of schema ξ minus average fitness of the population)*

$$\frac{dP(\xi, t)}{dt} = (\hat{\mu_\xi}(t) - \hat{\mu}(t))P(\xi, t) \tag{1.21}$$

Holland claimed that the simple genetic algorithm behaves according to the above equation. This is not true. Instead we have the surprising result:

THEOREM 1.12 *The Boltzmann distribution* $p(\mathbf{x},t) = e^{tf(\mathbf{x})}/Z_f(t)$ *with* $P(\xi,t) = \sum_{X|X_\xi = x_\xi} p(\mathbf{x},t)$ *fulfills Holland's equation (1.21).*

Proof: Taking the derivative we easily obtain

$$\frac{p(\mathbf{x},t)}{dt} = p(\mathbf{x},t)(f(\mathbf{x}) - \bar{f}(t)) \tag{1.22}$$

Let ξ define a schema, $\mathbf{x}_\xi$ the corresponding marginal distribution. Then

$$\begin{aligned} \frac{dP(\xi,t)}{dt} = \frac{dp(\mathbf{x}_\xi,t)}{dt} &= p(\mathbf{x}_\xi,t)\left(\frac{1}{p(\mathbf{x}_\xi,t)}\sum_{X|X_\xi=x_\xi} p(\mathbf{x},t)(f(\mathbf{x}) - \bar{f}(t))\right) \\ &= P(\xi,t)(\hat{\mu}_\xi(t) - \hat{\mu}(t)) \end{aligned}$$

Thus the Boltzmann distribution with the fixed *annealing schedule* $\beta(t) = t$ fulfills Holland's equation. *According to Holland's analysis* FDA *with this schedule should be an almost optimal algorithm!*

I hope this short discussion demonstrates that we now have a solid theory of genetic algorithms. But we are still far away from Holland's "logical theory of adaptive systems.t't'

14. Stochastic analysis and symbolic representations

We will use the stochastic analysis on more and more complex models. Finally we hope to analyze Holland's general model with the stochastic techniques presented above. Cellular automata can be seen as special cases of Holland's model. All automata perform in the same way, that is we have just one generator. Instead of a tree we have a one or two dimensional space. Selection can be modeled between neighboring automata. The reader has noticed that the stochastic analysis of cellular automata is already fairly difficult. This indicates that the analysis of Holland's model will be really difficult.

But in order to make progress in creating more intelligent machines, still another big step has to be done. From the discussions of our previous problems it becomes apparent that we have to *combine stochastic analysis with symbolic representations.* This problem was already stated by Wolfram Wolfram, 1994 in the context of cellular automata.

Problem 14 [Wolfram]: *What higher-level descriptions of information processing in cellular automata can be given?*

"One approach is statistical in nature. It consists in devising and describing attractors for the global evolution of cellular automata. All

initial configurations in a particular basin of attraction may be thought of as instances of some pattern, so that their evolution towards the same attractor may be considered as a recognition of the pattern ... The construction of attractors for more general problems is likely to be very difficult. An attempt in this direction might be made considering basis of attraction as sets of sequences corresponding to a particular formal language.

Another approach is to use symbolic representations for various attributes or components of cellular automaton configurations... perhaps data could be represented by an object like a graph, on which transformations can be performed in parallel... it seems likely that a radically new approach is needed."

The last statement seems to be correct. But to my knowledge Wolfram did not publish any proposal how to solve the problem.

15. Conclusion

In my opinion, the big problems in the theory of organisms and artificial automata have been recognized from the very beginning. In biology it was Darwin, in electronic computation von Neumann, Turing, Shannon. Some of the proposals for solving the challenging problems have been far too optimistic, other proposals have not been implemented because the implementation was too difficult. Therefore subsequent developments have lead to a fragmentation and specialization of research. This is true for biology as well as for computer science. Today evolutionary computation is divided into genetic algorithms, evolutionary algorithms, genetic programming, artificial life, and evolvable hardware – not to mention more specialized models like ant colony optimization, memetic algorithms, or classifier systems. But each model itself is too simple to solve the problems presented.

The challenging problems have faded away, less difficult problems and simpler models got into the center of attention. An exception is the problem of all problems: "Can we produce artificial intelligence comparable to or even surpassing human intelligence?" Researchers have often been too optimistic about the time scale to solve this problem. Whereas in the 60t's many researcher's predicted a solution in about 10 years, the time scale has now been increased to about 50 years! In my opinion, however, there will be no progress at all, unless not some of the sub-problems like the ones presented here will be solved.

Notes

1. In addition I recommend the essays of Stephen J. Gould.
2. McCulloch-Pitts had proven that their formal neural networks are equivalent to a Turing machine.
3. The discussion of the talk started with a remark of Bar-Hillel: “Dr. McCarthy’s paper belongs in the Journal of Half-Baked Ideas, the creation of which was recently proposed by Dr. I.J. Good.”
4. Whether this very precise value is justified by logical arguments is still a subject of hot discussions.

References

Burns, A.W. (1970). *Essays on Cellular Automata*. University of Illinois Press, Urbana.

Cherry, C. (1961). *Information Theory: Fourth London Symposium*. Butterworth, London.

Cover, Th. M. and Thomas, J.A. (1989). *Elements of Information Theory*. Wiley, New York.

Darwin, Ch. (1859). *The Origins of Species by Means of Natural Selection*. Penguin Classics, London.

Dawkins, R. (1989). *The Selfish Gene: Second Edition*. Oxford University Press, Oxford.

Griffiths, P. E. (2002). The philosophy of molecular and developmental biology. In *Blackwell Guide to Philosophy of Science*. Blackwell Publishers.

Holland, J.H. (1970a). Iterative circuit computers. In Burns, 1970, pages 277–296.

Holland, J.H. (1970b). Outline for a logical theory of adaptive systems. In Burns, 1970, pages 296–319.

Holland, J.H. (1973). Genetic algorithms and the optimal allocation of trials. *Siam Journal on Computing*, 2(2):88–105.

Holland, J.H. (1975/1992). *Adaptation in Natural and Artificial Systems*. Univ. of Michigan Press, Ann Arbor.

Jaynes, E.T. (1957). Information theory and statistical mechanics. *Phys. Rev*, 6:620–643.

Lauritzen, St. L. (1996). *Graphical Models*. Clarendon Press, Oxford.

Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Comm. of the ACM*, 38:924–948.

McCarthy, J. (1959). Programs with common sense. In *Mechanisation of Thought Processes*, pages 75–84. Her Majesty's Stationery Office, London.

McMullin, B. (2001). John von Neumann and the evolutionary growth of complexity: Looking backward, looking forward... *Artificial Life*, 6:347–361.

Minsky, M. (1961). Steps toward artificial intelligence. *Proc. of IRE*, 49:8–30.

Mühlenbein, H. (1991a). Darwin's continent cycle theory and its simulation by the Prisoner's Dilemma. *Complex Systems*, 5:459–478.

Mühlenbein, H. (1991b). Evolution in time and space - the parallel genetic algorithm. In Rawlins, G., editor, *Foundations of Genetic Algorithms*, pages 316–337. Morgan Kaufmann, San Mateo.

Mühlenbein, H. (1996). Algorithms, data and hypothese: Learning in open worlds. In G. Mahler, V.May, M.Schreiber, editor, *Molecular Electronics: Properties, Dynamics, and Applications*, pages 5–22. Marcel Dekker, New York.

Mühlenbein, H. and Høns, R. (2002). Stochastic analysis of cellular automata with application to the voter model. *Advances in Complex Systems*, 5(2-3):301–337.

Mühlenbein, H. and Mahnig, Th. (1999). FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376.

Mühlenbein, H. and Mahnig, Th. (2000). Evolutionary algorithms: From recombination to search distributions. In Kallel, L., Naudts, B., and Rogers, A., editors, *Theoretical Aspects of Evolutionary Computing*, Natural Computing, pages 137–176. Springer Verlag, Berlin.

Mühlenbein, H. and Mahnig, Th. (2002a). Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning. *Journal of Approximate Reasoning*, 31(3):157–192.

Mühlenbein, H. and Mahnig, Th. (2002b). Mathematical analysis of evolutionary algorithms. In Ribeiro, C. C. and Hansen, P., editors, *Essays and Surveys in Metaheuristics*, Operations Research/Computer Science Interface Series, pages 525–556. Kluwer Academic Publisher, Norwell.

Mühlenbein, H. and Mahnig, Th. (2003). Evolutionary algorithms and the Boltzmann distribution. In Jong, K. De, Poli, R., and Rowe, J., editors, *Foundations of Genetic Algorithms 7*. Morgan Kaufmann Publishers, San Francisco. to be published.

Mühlenbein, H., Mahnig, Th., and Ochoa, A. Rodriguez (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):215–247.

Opper, M. and Saad, D., editors (2001). *Advanced Mean Field Methods*, Cambridge. MIT Press.

Oyama, S. (2000). *Evolutions's Eye*. Duke University Press, Durham.

Parisi, D. and Ugolini, M. (2002). Living in enclaves. *Complexity*, 7:21–27.

Rapaport, A. (1970). Modern systems theory – an outlook for coping with change. *General Systems*, XV:15–25.

Shannon, C.E. (1953). Computers and automata. *Proc. of IRE*, 41:1234–1241.

Smith, J. Maynard and Szathmary, E. (1995). *The Major Transitions in Evolution*. W.H. Freeman, Oxford.

Turing, A.M. (1950). Computing machinery and intelligence. *Mind*, 59:433–460.

von Neumann, J. (1954). The general and logical theory of automata. In *The world of mathematics*, pages 2070–2101. Simon and Schuster, New York.

von Neumann, J. (1956). Probabilistic logics and the synthesis of reliable organs from unreliable components. In *Annals of Mathematics Studies 34*, pages 43–99. Princeton University Press.

Vose, M. (1999). *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge.

Whitehead, A. N. (1948). *Science and the Modern World*. Pelican Books, New York.

Wolfram, S. (1994). *Cellular Automata and Complexity*. Addison-Wesley, Reading.

Wright, S. (1937). The distribution of gene frequencies in populations. *Proc. Nat. Acad. Sci*, 24:253–259.