Adaptation*

John H. Holland

Department of Computer and Communication Sciences
Logic of Computers Group
The University of Michigan
Ann Arbor, Michigan

I. General Theory

A. Introduction

In biology, adaptation is the basic process underlying evolution, acting at all levels of organization from enzymes to ecosystems. Defined more generally, adaptation designates any process whereby a structure is progressively modified to give better performance in its particular environment. So defined, adaptation has a critical role in fields as diverse as psychology ("learning"), economics ("optimal planning"), control theory ("adaptive control"), sampling ("efficient inference"), and in the computer sciences generally.

* Research reported in this article was supported in part by the National Science Foundation under grant DCR 71-01997.

Adaptive processes are fundamentally optimization processes made difficult because the structures being modified are complex and their performance is uncertain. Frequently, nonadditive interaction (i.e., "epistasis" or "nonlinearity") makes it impossible to determine the performance of a structure from a study of its isolated parts. Moreover, possibilities for improved performance must usually be exploited at the same time that the search for further improvements is pressed. While these difficulties pose a real problem for the analyst, we know that they are routinely handled by biological processes, *qua* processes.

The approach set forth here [and developed in detail in Holland (1975)] constructs a mathematical framework which makes it possible to extract and generalize critical factors of the biological processes. Two of the most important generalizations are: (1) the concept of a *schema* (formally a hyperplane in the space of structures) as a generalization of an interacting coadapted set of genes, and (2) the generalization of genetic operators such as crossing-over, inversion, and mutation. The schema concept makes it possible to dissect and analyze complex "nonlinear" or "epistatic" interactions, while the generalized genetic operators extend the analysis to studies of learning, optimal planning, etc. The possibility of *intrinsic parallelism*—the testing of many schemata by testing a single structure—is a direct offshoot of this approach.

The *genetic algorithms* which have been developed in the course of this study offer a completely new approach to optimization. It can be proved that genetic algorithms rank *each* schema ξ, having two or more instances in the data base of sampled structures, according to the cumulative average of the observed values of its instances $\hat{\rho}_\xi$. That is, from a sampling point of view, each schema is an event in the sample space of structures, and it is ranked by the algorithm according to its sample average. [This is a consequence of the theorem that *each* schema ξ, with two or more instances at time t in the data base $\mathcal{B}(t)$, changes its proportion (or relative rank) $P_\xi{}^t$ in the data base according to the recursion $P_\xi^{t+1} = (1 - \epsilon)\hat{\rho}_\xi P^t$, where ϵ is a small constant.] This ranking occurs automatically despite the fact that it is structures, *not* schemata, which are being tested. (In the serial versions of the algorithm, the structures are being tested one at a time.) There will be in excess of $M^2 \cdot 2^{l/2}$ schemata with two or more instances in a data base of size M, where l is the word size. Because a sizable fraction of these schemata are being reranked by *each* test—the *intrinsic parallelism* of the algorithm—the effective *speedup* is at least $M \cdot 2^{l/2}$, in comparison with typical algorithms for searching high-dimensional, multimodal functions.

In the course of this article we will touch upon several applications of the general theory, but the most detailed discussion of an application will be reserved for the second part of the article. There we will exploit the properties of genetic algorithms to produce a highly flexible cognitive system capable of adaptation and learning.

B. The Traditional Approach of Mathematical Genetics

Fisher's fundamental theorem of natural selection, since its publication in 1930, has been the cornerstone of most mathematical studies of evolution and adaptation. [See, for example, Moran (1962); Wallace (1968); or Crow and Kimura (1970).] Several working hypotheses of population genetics stem directly from this work:

1. Evolution is described in terms of pools of alleles and shifts in allele frequencies.
2. Genetic variance is used to estimate the rate of evolution of populations.
3. Mutation is treated as the major driving force of evolution and adaptation.
4. Populations are discussed in terms of handicaps imposed by various kinds of genetic load.

These concepts are excellent guides, to both research and experiment, when phenotypic traits are determined by noninteracting genes (e.g., the blossom colors of Mendel's garden peas) or when the traits are the result of random increments produced by a great number of genes (e.g., the overall weight of an organism). However, the concepts and the underlying theorems depend heavily upon assumptions of the additivity of fitness across genes. Later versions of the theorem are more general, but they still use least mean square estimators, free recombination, or some other hypothesis which reduces the interplay of epistasis and linkage to a perturbation on the additive effects. Thus, assumptions of additivity still play a vital role.

Unfortunately, when these assumptions do not hold, at least approximately, the predictions based upon them can be badly misleading. Moreover, the advent of molecular biology has made it amply clear that most genes are, in fact, members of strongly interacting groups of genes. Genes control cell function and structure via enzymes, which catalyze reaction rates so strongly that they are the major determinants of an organism's observed characteristics. Because they are catalysts, the enzymes interact in a very nonlinear way in controlling cell reactions. If a sequence of reactions depends upon several enzymes, the overall reaction does not proceed at all, for the cell's purposes, until all of the enzymes are present. More complicated reactions involving positive and negative feedback are common, particularly those in which the output of a reaction sequence is a catalyst or inhibitor for some intermediate step of the reaction. Thus, the genes, through the enzymes which express their instructions, interact in a strongly nonlinear fashion. In the standard terminology of genetics, strong *epistatic* interactions are involved.

Of course the existence and, sometimes, the importance of epistasis have long been recognized in the general literature of genetics. The term *coadaptation*, in recognition that epistasis generally affects allele frequencies, has been used to describe observable correlations between allele frequencies at different loci. More generally, genes are said to be coadapted if they make little contribution to fitness unless their alleles are present in particular combinations. A variety of descriptive techniques, such as the depiction of adaptive or epigenic landscapes, have been used to portray the complicated relations between multilocus allele frequencies and fitness, and there are partial catalogs of epistatic effects. However these efforts provide no quantitative information. When epistasis is strong or linkage close, the usual mathematical approaches also fail to give much useful quantitative information and, worse (as we shall see), they can be misleading even at the qualitative level.

Here we will take a different approach. It is based on a demonstration that each coadapted set of alleles having several instances in a population has its own intrinsic rate of increase—its own fitness. Moreover, this fitness is an observable that can be estimated by sampling carriers of the coadapted set. As one of its consequences, this result enables us to derive a version of Fisher's theorem for coadapted sets of alleles, without using least mean square approximations or other assumptions of additivity. By embedding this result in a general mathematical framework we can go further, obtaining a general theory which provides both theorems about natural systems and algorithms for artificial systems.

C. Coadapted Sets and Schemata

Consider the set $\mathcal{A}$ of all possible genotypes (combinations of alleles) formed from a (haploid) set of genes $\{\delta_1, \cdots, \delta_l\}$. Let $\delta_i(A)$ designate the allele at locus i in a particular genotype $A \in \mathcal{A}$. That is, $[\delta_1(A), \delta_2(A), \ldots, \delta_l(A)]$ gives the particular alleles present in $A \in \mathcal{A}$. Following the usual approach of mathematical genetics, we can assign a fitness $\mu(A)$ to each genotype A. (Strictly, there is an oversimplification here, albeit a common one. Fitness can only be assigned to phenotypes—the fitness of a genotype must be inferred therefrom.) It is worth noting that these definitions are quite general, extending far beyond the domain of genetics. $\mathcal{A}$ can be almost any set of "structures" (mixes of economic goods, control policies, strategies for games, computer programs, etc.), and the δ_i become "detectors" picking out certain attributes or components of these structures.

That is, δ_i is a function: $\mathcal{A} \to V_i$ where V_i is the set of attributes associated with detector i. If $V_i = \{0, 1\}$, then δ_i is a predicate, saying that A possesses property i just in case $\delta_i(A) = 1$. In this broader context we say that $A \in \mathcal{A}$ has the *representation* $[\delta_1(A), \delta_2(A), \ldots, \delta_l(A)]$ in terms of the given set of l detectors. The fitness measure μ: $\mathcal{A} \to \{r_1 \mid 0 \leq r_1 \leq r\} = [0, r]$ becomes, in the more general context, a performance measure (utility of a mix of economic goods, average error for a control policy, payoff for a strategy, etc.).

Our objective now is to designate subsets of $\mathcal{A}$ that have attributes (alleles) in common. (These are, potentially, coadapted sets.) To do this let the symbol $\square$ indicate that we "don't care" what attribute occurs at a given position (i.e., for a given detector). Thus $(v_{13}, \square, \square, \ldots, \square)$ designates the subset of all elements in $\mathcal{A}$ having the attribute $v_{13} \in V_1$. [Equivalently, $(v_{13}, \square, \ldots, \square)$ designates the set of all l-tuples in $\mathcal{A}$ beginning with the symbol v_{13}; hence, for $l = 3$, (v_{13}, v_{22}, v_{32}) and (v_{13}, v_{21}, v_{31}) belong to $(v_{13}, \square, \square)$, but (v_{12}, v_{22}, v_{32}) does not.] The set of all l-tuples involving combinations of "don't cares" and attributes is given by the augmented product set $\Xi = \Pi_{i=1}^{l} \{V_i \cup \{\square\}\}$. Then any l-tuple $\xi = (\Delta_{i_1}, \Delta_{i_2}, \ldots, \Delta_{i_l}) \in \Xi$ designates a subset of $\mathcal{A}$ as follows: $A \in \mathcal{A}$ belongs to the subset if and only if (i) whenever $\Delta_{i_j} = \square$, any attribute from V_j may occur at the jth position of A, and (ii) whenever $\Delta_{i_j} \in V_j$, the attribute Δ_{i_j} must occur at the jth position of A. (For example, $(v_{11}, v_{21}, v_{31}, v_{43})$ and $(v_{13}, v_{21}, v_{32}, v_{43})$ belong to $(\square, v_{21}, \square, v_{43})$ but $(v_{11}, v_{21}, v_{31}, v_{42})$ does not.) The set of l-tuples belonging to Ξ will be called the set of *schemata*; Ξ amounts to a decomposition of $\mathcal{A}$ into a large number of subsets based on the representation in terms of the l detectors $\{\delta_1: \mathcal{A} \to V_i, i = 1, \ldots, l\}$.

Assume that, during the Tth generation, genotype $A \in \mathcal{A}$ has a probability $P(A)$ of occurring. That is, $\mathcal{A}$ becomes a sample space under the distribution P over $\mathcal{A}$. The performance measure μ then becomes a random variable over $\mathcal{A}$, $A \in \mathcal{A}$ being tried with probability $P(A)$ and yielding payoff $\mu(A)$. More importantly, any schema $\xi \in \Xi$ designates an event on the sample space $\mathcal{A}$. Thus, the restriction $\mu \mid \xi$ of μ to the subset designated by ξ, is also a random variable, $A \in \xi$ being tried with probability $[P(A)]/[\sum_{A' \in \xi} P(A')]$ and yielding payoff $\mu(A)$. In what follows ξ will be used to designate both an element of Ξ and the corresponding random variable with sample space ξ, the particular usage being specified when it is not clear from context. As a random variable, ξ has a well-defined average μ_ξ (and variance $\sigma_\xi{}^2$) where, intuitively, μ_ξ is the payoff expected when an element of ξ is randomly selected under the marginal distribution $[P(A)]/[\sum_{A' \in \xi} P(A')]$.

Clearly, when $\mu_\xi > \bar{\mu}(T)$, where $\bar{\mu}(T)$ is the average performance expected under P, instances of ξ (i.e., $A \in \xi$) are likely to exhibit performance better than the current average $\bar{\mu}(T)$. This suggests a simple procedure (a bit too simple as it turns out) for exploiting combinations of attributes associated with better-than-average performance while further searching $\mathfrak{A}$: (i) try instances of various schemata until at least one schema ξ is located which exhibits a sample average $\hat{\mu}_\xi > \bar{\mu}(T)$; (ii) generate *new* instances of the (observed) above-average schema ξ, returning to step (i) from time to time (particularly when the increasing overall average $\bar{\mu}(T)$ comes close to $\hat{\mu}_\xi$) to locate new schemata $\{\xi'\}$ for which $\hat{\mu}_{\xi'} > \bar{\mu}(T)$. In effect, then, credit is apportioned to a combination of attributes in accord with the observed average performance of its instances. This procedure has some immediate advantages over a fixed random (or enumerative) search of $\mathfrak{A}$; it generates improvement with high probability while gathering new information by trying new $A \in \mathfrak{A}$; furthermore, the new trials of the above-average schema ξ increase confidence that the observed average $\hat{\mu}_\xi$ closely approximates μ_ξ. It is oversimple because each instance $A \in \mathfrak{A}$ tried yields information about a great many schemata other than ξ—information which is not used.

Given l detectors, a single structure $A \in \mathfrak{A}$ is an instance of 2^l distinct schemata, as can be easily affirmed by noting that A is an instance of any schema ξ defined by substituting $\square$'s for one or more of the l attribute values in A's representation. Thus a single trial A constitutes a trial of 2^l distinct random variables, yielding information about the expected payoff μ_ξ of each. (If l is only 20, this is still information about a million schemata!) Any procedure which uses even a fraction of this information to locate ξ for which $\mu_\xi > \bar{\mu}(T)$ has a substantial advantage over the one-at-a-time procedure just proposed.

Exploiting this tremendous flow of information poses a much more clearly defined challenge than the one set forth in Section I, B. Schemata have advanced our understanding, in this sense, but the new problem is difficult. The amount of storage required quickly exceeds all feasible bounds if one attempts to record the average payoff of the observed instances of each schema sampled. Moreover, the information will be employed effectively only if it is used to generate new $A \in \mathfrak{A}$ which, individually, test as many above-average schema as possible. The adaptive system is thus faced with a specific problem of compact storage, access, and effective use of information about extremely large numbers of schemata. The reproductive plans set forth in the next section provide a resolution of these difficulties.

Before going to the discussion of reproductive plans let us look at a concrete interpretation of schemata which has general implications. This interpretation is aimed at general questions of function optimization. Consider an arbitrary bounded function $f(x)$, $0 \leqq x < 1$, and assume that x

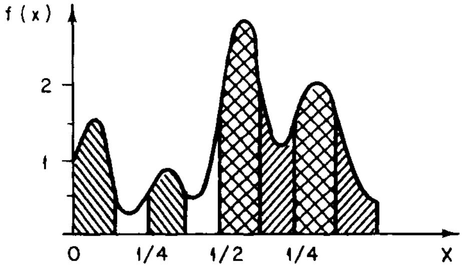


FIG. 1. Some schemata for a one-dimensional function. ▨—1□□□...□; ▧—□□0□...□; ▩—1□0□...□.

is specified to an accuracy of one part in a million, or, equivalently, that values of x are discretely represented by 20 bits. Define $\mathfrak{A}$ to be the set of 2^{20} discrete values of x represented with 20 detectors $\{\delta_j:\mathfrak{A} \rightarrow \{0, 1\}, j = 1, \ldots, 20\}$ where $\delta_j(x)$, $x \in \mathfrak{A}$, assigns to x the value of the jth bit in the binary expansion of x. The schema 1□□. . .□ then is just the right halfplane $\frac{1}{2} \leqq x < 1$, while the schema □□0□. . .□ is a set of four strips $\{0 \leqq x < \frac{1}{8}, \frac{1}{4} \leqq x < \frac{3}{8}, \frac{1}{2} \leqq x < \frac{5}{8}, \frac{3}{4} \leqq x < \frac{7}{8}\}$ and the schema 1□0□. . .□ is the intersection of the two previous schemata $\{\frac{1}{2} \leqq x < \frac{5}{8}, \frac{3}{4} \leqq x < \frac{7}{8}\}$ (see Fig. 1).

With this representation there are 3^{20} distinct schemata since any 20-tuple over the set {0, 1, □} defines a schema. (More technically, the schemata are simply hyperplanes, of dimension 20 or less, in the 20-dimensional space of detector-value combinations.) Note that there are many points, such as $x = \frac{13}{16} = 0.11010\ldots0$, which are instances of all three of the schemata just singled out. Note also that f has a well-defined average value f_ξ on each schema ξ (for any weighting of the values $f(x)$, as by a probability distribution). Clearly, for any x, knowledge of $f(x)$ is relevant to estimating f_ξ for any schema for which $x \in \xi$. Moreover, observations of f for relatively few x will enable f_ξ to be estimated for a great many $\xi \in \Xi$. Even a sequence of four observations, say $x(1) = 0.0100010\ldots0$, $x(2) = 0.110100\ldots0$, $x(3) = 0.1000010\ldots0$, $x(4) = 0.1111010\ldots0$, enables one to calculate three-point estimates for many schemata, e.g. (assuming all points are equally likely or equally weighted),

$$\hat{f}_{1\square\square\cdots\square} = [f(x(2)) + f(x(3)) + f(x(4))]/3$$

and

$$\hat{f}_{\square\square\square\square01\square\cdots\square} = [f(x(1)) + f(x(3)) + f(x(4))]/3$$

and two-point estimates for even more schemata, e.g.,

$$\hat{f}_{\square1\square\square01\square\cdots\square} = [f(x(1)) + f(x(4))]/2$$

and

$$\hat{f}_{11\square\cdots\square} = [f(x(2)) + f(x(4))]/2.$$

The picture is not much changed if f is a function of many variables $x_1, \ldots, x_d$. Using binary representations again, we now have $20d$ detectors (assuming the same accuracy as before), 3^{20d} schemata, and each point is an instance of 2^{20d} schemata. In the one-dimensional case the representation transformed the problem to one of sampling in a 20-dimensional space—already a space of high dimensionality—so the increase to a $20d$-dimensional space really involves no significant conceptual changes. Interestingly, each point $(x_1, \ldots, x_d)$ is now an instance of 2^{20d} schemata rather than 2^{20} schemata, an exponential (dth power) increase. Thus, for a given number of points tried, we can expect an exponential (dth power) increase in the number of schemata for which f_ξ can be estimated with a given confidence. As a consequence, if the information about the schemata can be stored and used to generate relevant new trials, high dimensionality of the argument space $\{0 \leqq x_j < 1, j = 1, \ldots, d\}$ imposes no particular barrier.

It is also interesting in this context to compare two different representations for the same underlying space. Six detectors with a range of ten values can yield approximately the same number of distinct representations as twenty detectors with a range of two values, since $10^6 \simeq 2^{20} = 1.05 \times 10^6$ (cf. decimal encoding vs binary encoding). However the numbers of schemata in the two cases are vastly different: $11^6 = 1.77 \times 10^6$ vs $3^{20} = 3.48 \times 10^9$. Moreover in the first case each $A \in \mathcal{A}$ is an instance of only $2^6 = 64$ schemata, whereas in the second case each $A \in \mathcal{A}$ is an instance $2^{20} = 1.05 \times 10^6$ schemata. This suggests that, for adaptive plans which can use the increased information flow (such as the reproductive plans), many detectors deciding among few attributes are preferable to few detectors with a range of many attributes. In genetics this would correspond to chromosomes with many loci and few alleles per locus (the usual case) rather than few loci and many alleles per locus.

D. Reproductive Plans

The fundamental mechanism of genetic adaptation is reproduction according to performance. (Indeed the definition of fitness is based upon this coupling of performance and reproduction.) Setting this in a more general framework, let us define a *reproductive plan* as one which makes the probability $P(A)$ of generating new variants (samples) of a structure $A \in \mathcal{A}$ depend upon A's performance $\mu(A)$. Using this formulation we can design a reproductive plan for any domain of structures $\mathcal{A}$ having a meaningful notion of performance $\mu : \mathcal{A} \to [0, r]$. We will see that such a plan seeks the global optimum of μ. Moreover, during the search, it exploits any local optima (false peaks) it encounters to improve its current average performance $\bar{\mu}(T)$.

In the remainder of this article we will concentrate on a particular reproductive plan as a representative of the class. The plan operates in terms of a basic cycle which is repeated to provide successive generations $\mathcal{B}(1)$, $\mathcal{B}(2), \ldots, \mathcal{B}(T), \ldots$ of structures to be tested. Informally, the steps of the cycle are:

1. Reproduce the structures $A \in \mathcal{B}(T)$ in proportion to their observed performances $\mu(A)$.
2. Modify each of the reproduced structures, using genetic operators (crossing-over, inversion, etc.) to produce the new generation $\mathcal{B}(T + 1)$.
3. Return to step (1).

Let us begin by considering the first step of the cycle, the reproductive step, letting $\mathcal{B}(T)$ consist of M structures (genotypes) $\{A_1(T), A_2(T), \ldots, A_M(T)\}$. Corresponding to each of these structures is its observed performance $\mu[A_i(T)]$, which will be abbreviated to $\mu_i(T)$. Thus, at time T, there is a set of observations $I(t) = \{\mu_1(T), \mu_2(T), \ldots, \mu_M(T)\}$. These observations, along with $\mathcal{B}(T)$, constitute the "data base" of the plan at time T, in the sense that the plan's total information about its past history and current condition is contained in $I(T) \cup \mathcal{B}(T)$.

It is useful to keep the size of $\mathcal{B}(T)$ constant over successive generations. (In biological terms the size of $\mathcal{B}(T)$ would constitute a maximum population size, set by the "carrying capacity" of the environment. If, at time T, the population size were less than M, then the remaining positions in the "data base" would remain empty—i.e., they would be occupied by null structures.) To accomplish this we must reconcile individual "reproduction according to performance" with the requirement that the overall average reproduction rate be 1 so that successive generations have the same size. We do this by assigning the reproduction rate $\rho_i(T) = \mu_i(T)/[\sum_j \mu_j(T)/M] = \mu_i(T)/\bar{\mu}(T)$ to the individual $A_i(T)$. Clearly, if $\mu_h(T) = c\mu_i(T)$ then $A_h(T)$ has c times as may offspring as $A_i(T)$, while the overall average reproduction rate is

$$\begin{aligned}(1/M)\ &\textstyle\sum_i \rho_i(T)\\ &= (1/M)[1/\bar{\mu}(T)] \textstyle\sum_i \mu_i(T)\\ &= [1/\textstyle\sum_j \mu_j(T)] \textstyle\sum_i \mu_i(T)\\ &= 1\end{aligned}$$

as required.

Now, let us see what happens during the reproductive step to a schema ξ with one or more instances in $\mathcal{B}(T)$. Let there be $M_\xi(T)$ instances of ξ

in $\mathcal{B}(T)$ and let this set of instances be designated $\xi(T)$. Each $A_i(T) \in \xi(T)$ will of course produce $\rho_i(T)$ offspring. Thus there will be

$$\sum_{i|\xi} \rho_i(T) = \sum_{i|\xi} \left(\frac{\mu_i(T)}{\bar{\mu}(T)}\right)$$

total offspring to instances of ξ in $\mathcal{B}(T)$ where $i \mid \xi$ means those indices i such that $A_i(T) \in \xi(T)$. Stated another way, the average rate of increase of instances of ξ at T is

$$\rho_\xi(T) = [\sum_{i|\xi} \mu_i(T)]/[\bar{\mu}(T) \cdot M_\xi(T)]$$

Or, what is the same, the number of offspring is given by

$$\rho_\xi(T) M_\xi(T) = \frac{\mu_\xi(T)}{\bar{\mu}(T)} M_\xi(T)$$

where $\mu_\xi(T)$ is the average performance of the instances of ξ at T. Thus, during the reproductive step, *each* schema with instances in $\mathcal{B}(T)$ increases at its *own* intrinsic rate—a rate proportional to the average performance of its instances.

Here we have the first hint of intrinsic parallelism. Each schema ξ increases its proportion in successive generations at a rate determined by the average performance μ_ξ of *its* instances, independently of what is happening to other schemata. There will be somewhere between 2^l and $M \cdot 2^l$ schemata being handled this way even though there are only M structures in the data base $\mathcal{B}(T)$. Moreover, since μ_ξ determines the rate of increase of each ξ, the number of instances of ξ in $\mathcal{B}(T)$ soon reflects its rank (average contribution to performance) relative to the other schemata. In effect, each schema ξ in $\mathcal{B}(T)$ is ranked on a scale from 0 to M according to the number of its instances in $\mathcal{B}(T)$, and this ranking reflects the potential usefulness of ξ's instances. In this way the M elements of $\mathcal{B}(T)$ compactly store the relative rankings of a tremendous number of schemata, reflecting the adaptive plan's past experience. Finally, since the random destruction of a few elements of $\mathcal{B}(T)$ can affect the relative rankings by at most a few "notches" on the 0 to M scale, $\mathcal{B}(T)$ stores this information in a very reliable way. (This distributed form of storage, though different in detail, is reminiscent of holography in its ability to sustain all sorts of local damage without losing the overall pattern.)

It only remains to show that the application of the genetic operators in the second step does not seriously perturb the intrinsic parallelism. At first sight, the need for genetic operators might be unclear; the schemata are already being treated in the desired way by step (1). However, large though $M \cdot 2^l$ is, it will in general be a small fraction of the total 3^l schemata. (For $l = 30$, $M = 1000$, we have $M \cdot 2^l \simeq 10^{12}$ schemata, while $3^l \simeq 2 \cdot 10^{14}$.) Thus, it is important to be able to generate instances of schemata not already tried, preferably in an intrinsically parallel fashion. This is the function *par excellence,* of the genetic operator *crossing-over.*

Because crossing-over serves well as a paradigm for other genetic operators we will concentrate on it here, referring the reader to "Adaptation in Natural and Artificial Systems" (Holland, 1975) for detailed discussions of some of the other operators. In biological systems, crossing-over is a process yielding recombination of alleles via exchange of segments between pairs of chromosomes. We can lift this process to the level of a general operator on structures, by considering the structures in terms of their representations $[\delta_1(A), \delta_2(A), \ldots, \delta_l(A)]$.

Crossing-over proceeds in three steps:

1. Two structures, $A = (a_1, a_2, \ldots, a_l)$ and $A' = (a_1', a_2', \ldots, a_l')$, are selected at random from the set of elements resulting from the reproductive step.

2. A number x is selected at random from $\{1, 2, \ldots, l - 1\}$.

3. Two new structures are formed from A and A' by exchanging sets of alleles to the right of position x, yielding $A'' = (a_1, \ldots, a_x, a_{x+1}', \ldots, a_l')$ and $A''' = (a_1', \ldots, a_x', a_{x+1}, \ldots, a_l)$.

The two direct effects of crossing-over on the pool of schemata in $\mathcal{B}(T)$ are:

1. Generation of "new" schemata (schemata having neither A nor A' as an instance). For example, if $a_x \neq a_x'$ or $a_{x+1} \neq a_{x+1}'$, the schema $\square\ldots\square a_x a_{x+1}'\square\ldots\square$ has an instance *after* the crossing-over, though neither A nor A' is an instance of it.

2. Generation of new instances of schemata already in $\mathcal{B}(T)$. For example, if $a_i \neq a_i'$ for some $i \geqq x$, the schema $a_1 a_2\square\ldots\square$ has an instance after crossing-over which is distinct from A. Each new instance of a schema ξ amounts to a new trial of the random variable corresponding to ξ. As such it increases confidence in the observed average performance μ_ξ of the instances of ξ.

Each crossing-over affects great numbers of schemata. If A differs from A' at x' loci to the left of $x + 1$, and at x'' loci to the right of x, then as a result of a crossover at x there will be $2^l - 2^{l-x'} - 2^{l-x''} + 2^{l-(x'+x'')}$ "new" schemata introduced, and $2^{l-x'} + 2^{l-x''} - 2^{l-(x'+x'')}$ new instances of schemata already instanced by A or A'. [See Lemma 6.2.1 of Holland (1975).] If $l = 30$, this means that 10^9 schemata are usefully manipulated by *each* crossing-over:

Now we must investigate the effect of crossing-over on the reproduction rate μ_ξ of the reproductive step. Consider the schema $\xi = \square\ldots\square a_i \ldots a_j\square\ldots\square$. Let A be an instance of ξ and let A' be an instance of the complement of ξ, i.e., $A' \notin \xi$. If A and A' are crossed and the crossover point falls between i and j, then *neither* of the resultants will be instances of ξ (except in special cases). Thus, crossing-over *can* destroy some instances of a schema, thereby reducing the effective reproduction rate over the full cycle of the reproductive plan.

Since the crossover point is chosen at random, the probability that it will fall between i and j is simply $(j - i)/(l - 1)$. (If the crossover point falls outside the region between i and j, the result is still an instance of the schema.)

If we assume every crossover falling within the limits of a schema ξ is destructive [a condition only true if there are not many instances of ξ in $\mathcal{B}(T)$], then we can set an upper bound on the reduction of μ_ξ resulting from crossing-over,

$$M_\xi(T+1) \geqq \frac{\mu_\xi}{\bar{\mu}}\left[1 - \left(\frac{l_\xi}{l-1}\right)\right] P_C M_\xi(T)$$

where l_ξ is the difference between the rightmost and leftmost loci exhibiting a non-□ in the definition of ξ, and P_C is the probability that a given structure will undergo crossing-over in a given generation. We see that crossing-over is always a small perturbation for schemata ξ for which $l_\xi/(l - 1)$ is small. If l_ξ involves many alleles or widely separated alleles, then μ_ξ must be proportionately large for the number of instances of ξ to increase in successive generations. For instance if $l_\xi = (l - 1)/2$ then μ_ξ must be greater than $2\bar{\mu}$ for ξ to increase (assuming $P_C = 1$, i.e., every structure undergoes crossing-over in every generation).

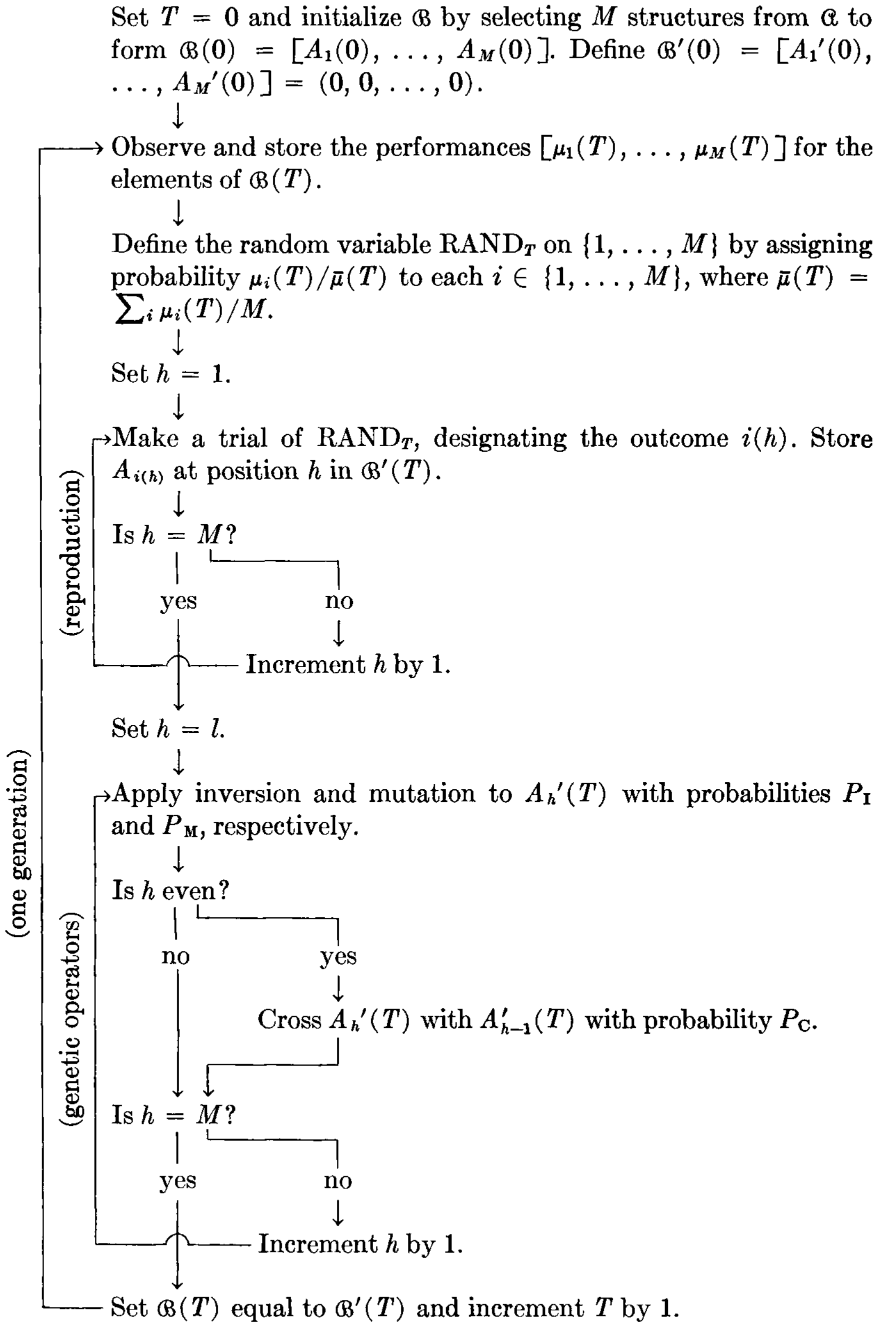
Similar results can be derived for other genetic operators (inversion, mutation, etc.) indicating small perturbations when l_ξ, the "length" of the schema, is small relative to the overall length l of the representation. For crossing-over, inversion, and mutation, with probabilities P_C, P_I, and P_M, respectively, the result can be written

$$M_\xi(T+1) \geqq \frac{\mu_\xi}{\bar{\mu}}\left[1 - \frac{l_\xi}{l-1}\,(P_C + 2P_I) - P_M^{n(\xi)}\right] M_\xi(T)$$

where $n(\xi)$ is the *number* of non-box loci in the definition of ξ. [See Lemma 7.2 in Holland (1975).]

From this we see that the full cycle of the reproductive plan, by simple operations on the "data base" $\mathcal{B}(T)$, produces sophisticated, intrinsically parallel tests of the space of schemata. Local optima (ξ for which $\mu_\xi > \bar{\mu}$), instead of diverting or trapping the plan, are exploited to improve the average performance of the plan, while the search goes on. Moreover, large numbers of schemata, for which l_ξ is not too large relative to l, have their relative rankings compactly stored in $\mathcal{B}(T)$.

A more detailed outline of the reproductive plan just discussed follows.



Set $T = 0$ and initialize $\mathcal{B}$ by selecting M structures from $\mathcal{A}$ to form $\mathcal{B}(0) = [A_1(0), \ldots, A_M(0)]$. Define $\mathcal{B}'(0) = [A_1'(0), \ldots, A_M'(0)] = (0, 0, \ldots, 0)$.

Observe and store the performances $[\mu_1(T), \ldots, \mu_M(T)]$ for the elements of $\mathcal{B}(T)$.

Define the random variable RAND_T on $\{1, \ldots, M\}$ by assigning probability $\mu_i(T)/\bar{\mu}(T)$ to each $i \in \{1, \ldots, M\}$, where $\bar{\mu}(T) = \sum_i \mu_i(T)/M$.

Set $h = 1$.

Make a trial of RAND_T, designating the outcome $i(h)$. Store $A_{i(h)}$ at position h in $\mathcal{B}'(T)$.

Is $h = M$?

Increment h by 1.

Set $h = l$.

Apply inversion and mutation to $A_h'(T)$ with probabilities P_{I} and P_{M}, respectively.

Is h even?

Cross $A_h'(T)$ with $A_{h-1}'(T)$ with probability P_{C}.

Is $h = M$?

Increment h by 1.

Set $\mathcal{B}(T)$ equal to $\mathcal{B}'(T)$ and increment T by 1.

E. Fisher's Theorem Generalized to Coadapted Sets

The results of the last section weigh strongly against the (still widely held) view that biological adaptation proceeds primarily by the selection of advantageous mutant genes. In addition, the results directly contradict the closely related view (in mathematical genetics) that alleles are replaced independently of each other, increasing or decreasing according to their individual average excesses. The results suggest, rather, that the adaptive process works largely in terms of pools of schemata (potentially coadapted sets of genes). Because the pool of schemata corresponding to a population is so much larger than the pool of genes (2^l vs $2l$), selection has broader scope with many more pathways to improvement and the great advantage of intrinsic parallelism.

The foregoing statements, and the results, take on a more familiar form if they are recast using the "average excess (of fitness) of a coadapted set." To accomplish this the average excesses must be defined in terms of the observable μ_ξ.

Let $\mu_\xi'(T) \overset{\text{df.}}{=} M_\xi(T+1)/M_\xi(T)$, the effective rate of increase of the schema ξ at time T. By the previous section

$$\mu_\xi'(T) \geqq \rho_\xi(T) \left[1 - \frac{l_\xi}{l-1} (P_\mathrm{C} + 2P_\mathrm{I}) - P_\mathrm{M}^{n(\xi)}\right]$$

If we allow the reproduction rate (fitness) ρ_ξ to be derived from an arbitrary function

$$\mu : \mathfrak{A} \rightarrow [0, r]$$

rather than using $\mu/\bar{\mu}$, the resulting theorem will hold for an arbitrary changing (growing or decreasing) population. [That is $\mu_\xi(T) = \sum_{A \in \xi(T)} \mu(A)/M_\xi(T)$, with no requirement that $\bar{\mu}(T) = 1$.] Following this line we have

$$\mu_\xi'(T) \geqq [\textstyle\sum_{A \in \xi(T)} \mu(A)]\,(1 - \epsilon_\xi)/M_\xi(T)$$

where

$$1 - \epsilon_\xi = 1 - \frac{l_\xi}{l-1} (P_\mathrm{C} + 2P_\mathrm{I}) - P_\mathrm{M}^{n(\xi)}$$

In order to derive a Fisher-like formula suitable for either discrete or continuous formulations we note that

$$M_\xi(T + \Delta T) - M_\xi(T) \simeq [M_\xi(T+1) - M_\xi(T)]\Delta T$$

where the equation becomes exact as $\Delta T \rightarrow 0$. Using the definition of μ_ξ' we get

$$M_\xi(T + \Delta T) - M_\xi(T) \simeq (\mu_\xi' - 1) M_\xi(T) \Delta T$$

The probability $P(\xi, T)$ that $A \in \mathcal{B}(T)$ belongs to ξ is simply

$$P(\xi, T) = M_\xi(T)/M(T)$$

and the rate of change of $P(\xi, T)$ is

$$\Delta P(\xi, T) = \frac{M_\xi(T + \Delta T)}{M(T + \Delta T)} - \frac{M_\xi(T)}{M(T)}$$

taking account of the fact that the size of the population, $M(T)$, can change with time. Substituting from above we have

$$\Delta P(\xi, T) \simeq \frac{[(\mu_\xi' - 1)\Delta T + 1]M_\xi(T)}{[(\bar{\mu} - 1)\Delta T + 1]M(T)} - \frac{M_\xi(T)}{M(T)}$$

$$= \left(\frac{[(\mu_\xi' - 1)\Delta T + 1]}{[(\bar{\mu} - 1)\Delta T + 1]} - 1\right)\frac{M_\xi(T)}{M(T)}$$

$$= \frac{(\mu_\xi' - \bar{\mu})\Delta T}{(\bar{\mu} - 1)\Delta T + 1} P(\xi, T)$$

It follows that

$$\frac{\Delta P(\xi, T)}{\Delta T} \simeq \frac{(\mu_\xi' - \bar{\mu})}{(\bar{\mu} - 1)\Delta T + 1} P(\xi, T)$$

If we take the limit as $\Delta T \to 0$, going to a continuous time scale, we have

$$\frac{dP(\xi, t)}{dt} = (\mu_\xi' - \bar{\mu})P(\xi, t)$$

$(\mu_\xi' - \bar{\mu})$ is just the average excess associated with schema ξ. This equation, *restricted to alleles*, is just Fisher's (1930) classic result; however, it holds for arbitrary schemata. Using the result $\mu_\xi' \geqq \mu_\xi(1 - \epsilon_\xi)$ we have

$$\frac{dP(\xi, t)}{dt} \geqq [\mu_\xi(1 - \epsilon_\xi) - \bar{\mu}]P(\xi, t)$$

giving the rate of increase in terms of the average performance defined earlier. On a discrete time scale $T = 1, 2, 3, \ldots, \Delta T = (T + 1) - T = 1$ and the above becomes

$$\Delta P(\xi, T) \gtrsim \frac{[\mu_\xi(1 - \epsilon_\xi) - \bar{\mu}]}{\bar{\mu}} P(\xi, T)$$

The overall result gives us a way of predicting the rate of increase of a set of alleles with epistatic interactions from a sample average $\hat{\mu}_\xi$ of the fitnesses of carriers of the set of alleles.

At this point we can compare predictions in terms of schemata with predictions made from the classic approach assuming noninteracting genes (additivity). Consider the two-locus case with three alleles per locus and the following table of fitnesses:

A	$\mu(A)$
UX	0.8
UY	0.3
UZ	1.6
VX	1.1
VY	1.4
VZ	0.8
WX	1.4
WY	1.3
WZ	0.3

Then, assuming all combinations are equally likely, we can derive the following table for various μ_ξ

ξ	$\mu(\xi)$
$U\square$	0.9
$V\square$	1.1
$W\square$	1.0
$\square X$	1.1
$\square Y$	1.0
$\square Z$	0.9

Under independent selection

$$P(a_1a_2) = P(a_1\square)P(\square a_2)$$

so that

$$\begin{aligned}\frac{dP(a_1a_2)}{dt} &= \frac{d}{dt}\left[P(a_1\square)P(\square a_2)\right]\\ &= P(\square a_2)\,\frac{dP(a_1\square)}{dt} + P(a_1\square)\,\frac{dP(\square a_2)}{dt}\\ &= P(a_1a_2)\left[\frac{1}{P(a_1\square)}\,\frac{dP(a_1\square)}{dt} + \frac{1}{P}\,\frac{dP(\square a_2)}{dt}\right]\\ &= P(a_1a_2)\left(\left[\mu(a_1\square) - \bar{\mu}\right] + \left[\mu(\square a_2) - \bar{\mu}\right]\right)\end{aligned}$$

using the fact that

$$\frac{1}{P(\xi)} \cdot \frac{dP(\xi)}{dt} = [\mu(\xi) - \bar{\mu}].$$

Thus, assuming independent selection, we predict that *UZ decreases* at the rate

$$P(UZ)([\mu(U\square) - 1] + [\mu(\square Z) - 1]) = -0.2\, P(UZ)$$

while *VX increases* at the rate

$$P(VX)([\mu(V\square) - 1] + [\mu(\square X) - 1]) = +0.2\, P(VX)$$

These results clearly contradict the table of fitnesses where UZ with fitness 1.6 is much more favorable than VX with fitness 1.1.

On the other hand the formulation in terms of schemata yields rates

$$P(UZ)([\mu(UZ) - 1]) = +0.6\, P(UZ)$$

and

$$P(VX)([\mu(VX) - 1]) = +0.1\, P(VX)$$

as required.

It is important to note that the same differences in prediction would occur if we were considering the two loci in a much larger context (say full chromosomes). In effect we would then be considering the schemata $\square \ldots \square a_1 a_2 \square \ldots \square$ along with the "component" schemata $\square \ldots \square a_1 \square \square \ldots \square$ and $\square \ldots \square \square a_2 \square \ldots \square$. In this larger context assumptions of independent selection would still yield exactly the same misleading predictions. When epistasis is important, the reproductive plans and the corresponding theorems about schemata provide a better hypothesis than the hypothesis of independent selection or the use of least mean squares estimates of the fitness of sets of alleles.

II. An Adaptive Cognitive System

A. Introduction

To illustrate the application of this theory outside the domain of genetics, let us consider the construction of a cognitive system that adapts to its environment under sensory guidance. The version presented here is a relatively simple version—a more complete version will appear in J. Holland and J. Reitman (in preparation). In broadest outline, the system consists of four parts: a set of resource reservoirs, a detector array, a classifier array, and an effector array (see Fig. 2).

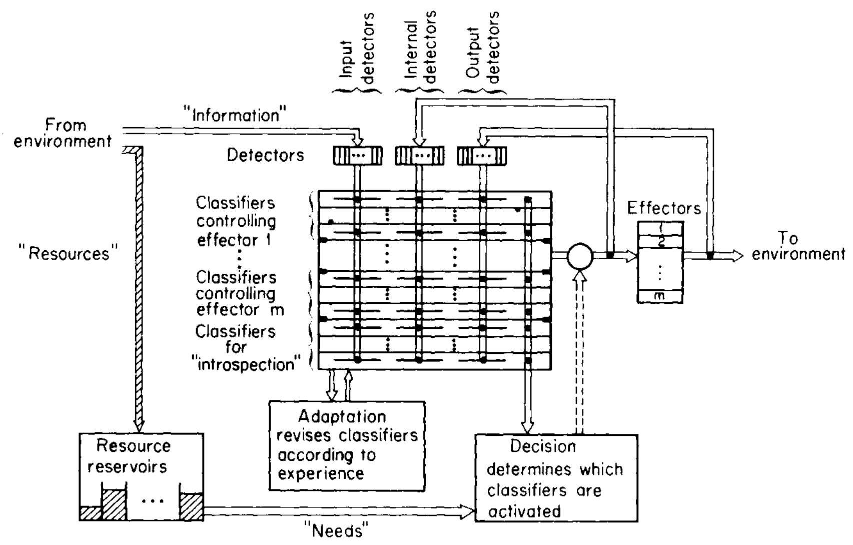

Input detectors
Internal detectors
Output detectors
From environment
"Information"
Detectors
"Resources"
Classifiers controlling effector 1
Classifiers controlling effector m
Classifiers for "introspection"
Effectors
1
2
m
To environment
Resource reservoirs
Adaptation revises classifiers according to experience
Decision determines which classifiers are activated
"Needs"

FIG. 2. Major sections of the cognitive system.

The resource reservoirs determine the ultimate goals of the system. They are being continually depleted as time elapses and must be replenished by actions of the system upon the environment. Stated another way, the current needs of the system are determined by how empty its reservoirs are.

In the detector array it is assumed for simplicity that each detector is either turned on (1) or off (0) at each time step by the overall situation (environmental + internal) presented at that time.

Each classifier consists of two parts: an input side and an output side. The input side is sensitive to a range of detector configurations—a subset of the set of all detector configurations—called a *taxon*. The output side specifies the condition of a given effector in the effector array. The effector is to be set to this condition (activated or deactivated) whenever the classifier is activated. In addition the classifier keeps a running average, called the *performance figure*, of the needs satisfied when it has taken an active part in determining part of an effector sequence. (More about this later.)

Each effector in the effector array specifies a way in which the system can affect the environment. More formally, the effector specifies a transformation of the environment. Each effector can be activated or deactivated independently of the others.

The condition (active or inactive) of each effector at each time step is determined by a competition among the classifiers associated with that effector. The winner sets the condition of the effector. In determining the winner, each classifier in the set is scored on two counts:

1. The classifier's taxon is scored on how closely it specifies the current detector configuration.
2. The classifier's performance figure is scored on how closely it matches current needs.

The product of these two scores determines the classifier's standing in its competition with the other classifiers controlling the condition of the effector. Each classifier in the set is assigned a probability proportional to its standing (so that the sum of the probabilities over the set of competitors is 1). Then a classifier is chosen at random according to this distribution (i.e., the higher the classifier's standing the more likely it is to be chosen). The selected classifier is designated the winner and is said to be *activated.* Because of the product, any classifier that poorly specifies the detector configuration produced by the input or that yields an inappropriate performance relative to current needs is unlikely to win the competition.

At intervals, some action (setting of the effector array) of the cognitive system will produce an input of resources to the system, increasing the levels of the reservoirs. This input will be called a *payoff.* The interval between payoffs will be called an *epoch.* Each classifier activated during an epoch will (usually) receive some credit for the payoff at the end of the epoch (but see below). Its performance figure will be modified accordingly and it will be deactivated.

B. Behavioral Mechanisms

There are two basic steps in making this account of the behavioral mechanisms precise. First, the notion of taxon and its match to an arbitrary detector configuration must be defined. Second, the algorithm for deriving a classifier's performance figure must be defined, along with a way of determining its match to an arbitrary need configuration.

Let us start with the notion of a taxon. Each taxon will be taken to be a specific hyperplane *in the space of the detector configurations.* Thus, if we use # to indicate an irrelevant detector, the set of all taxa is simply

$$\Gamma = \{0, 1, \#\}^l$$

That is, as in the case of schemata, a taxon is named by an l-tuple $(\lambda_1, \lambda_2, \ldots, \lambda_l)$ where

$\lambda_j = 1$ (or 0) indicates that detector j must be *on* (or *off*) for the overall detector configuration to be a member of the taxon.

$\lambda_j = \#$ indicates that detector j can have either value without affecting the configuration's membership in the taxon.

Given a detector configuration $(i_1, i_2, \ldots, i_l)$, the *matching score* of a given taxon is 0 if, for some j, $\lambda_j = 1$ and $i_j = 0$ or $\lambda_j = 0$ and $i_j = 1$. That is, a mismatch at some relevant position gives the taxon the lowest possible score. Otherwise the score is $l - n(\#)$ where $n(\#)$ is a count of the number of irrelevant positions. That is, the more closely the taxon specifies the configuration, the higher its score.

The second basic step is specification of an algorithm for generating the performance figure of each classifier. The basic idea is that the performance figure should reflect the average payoff *rate* achieved during epochs when the classifier is activated. (The payoff *rate* achieved during any given epoch is the payoff received at the end of the epoch divided by the length of the epoch.) In effect the performance figure amounts to a prediction (or estimate) of what is to be expected when (or if) the classifier is activated. Given the performance figure π_γ at the beginning of an epoch in which the classifier γ is activated, the object is to revise that figure to conform to the changed average resulting from the payoff U at the end of the epoch. If the classifier has been activated ν_γ times in the past and if the length of the epoch is E time steps then the new performance figure π_γ' (after payoff) is

$$\pi_\gamma' = \frac{\nu_\gamma}{\nu_\gamma + 1}\pi_\gamma + \frac{1}{\nu_\gamma + 1}\left(\frac{U}{E}\right)$$

(Notice that π is an r-tuple, since U is.) Assuming that the reservoirs are depleted at a constant rate between payoffs, and setting the unit of measurement for the reservoir so that exactly one unit is used per time step we have

$$E = L_1(t_j) - L_1(t_{j+1} - 1)$$

when $L_1(t)$ is the level of the reservoir at time t and t_j is the first time step of the jth epoch. Rewriting the expression for π' using this notation,

$$\pi_\gamma(t_{j+1}) = \left(\frac{\nu_\gamma}{\nu_\gamma + 1}\right)\pi_\gamma(t_j) + \left(\frac{1}{\nu_\gamma + 1}\right)\frac{U(t_{j+1})}{L_1(t_j) - L_1(t_{j+1} - 1)}$$

and, of course,

$$\nu_\gamma(t_{j+1}) = \nu_\gamma(t_j) + 1$$

This would be adequate for revising the performance figure were it not for the fact that we want the performance figure to be a *consistent predictor* of performance, as well as a measure of past performance. In this context consistent prediction means that a *succession* of activated classifiers associated with a given effector would all make the same prediction. That is, their performance figures (expected payoffs) should be approximately the same.

The *error* involved, if successive performance figures are not the same, is defined as $\epsilon_\gamma = [(\pi_{\gamma'}(t + t_1) - \pi_\gamma(t))]^-$, where t and $t + t_1$ are their respective activation times, and $[(w_1, \ldots, w_r)]^-$ is defined as the sum of the *negative* w_j, recalling that π_γ and hence the difference $w = (\pi_{\gamma'} - \pi_\gamma)$ is an r-tuple. That is, ϵ_γ is the sum of the *negative* component differences.

Two principles underlie the use of the error number:

1. The later of two activated classifiers is taken to make the more accurate prediction (being closer to the end of the epoch).
2. Only errors of negative sign (the earlier activated classifier "overestimates") will be attended to. (A classifier that underestimates is less likely to win the competition for activation, and the consequences of gaining more payoff than expected are less serious than the reverse.)

Each successive negative error in a succession of activated classifiers will cause an *attenuation* β of the payoff credited to *all* classifiers activated earlier. That is, when the performance figure of a classifier is revised at the end of an epoch, the payoff U credited to it will be reduced according to the number and size of intervening prediction errors. More precisely, let

$$\beta_1 = 1 - e^{c/\epsilon}$$

where $c > 0$ (and usually c is of the order of $\frac{1}{2}\, L_{\max}$, half the maximum reservoir capacity). For negative ϵ's this function generates a typical "s curve," slowly departing from 1, then dropping almost linearly and ultimately approaching 0 as an asymptote. The *cumulative attenuation* β at time t is given by the recursion

$$\begin{aligned}\beta_\gamma(t+1) &= \beta_{1\gamma}\cdot\beta_\gamma(t) && \text{if } \epsilon_\gamma \text{ is negative}\\ &= 1 && \text{if the classifier was deactivated at time } t \text{ (i.e., } \beta_\gamma \text{ is ``reset'')}\\ &= \beta_\gamma(t) && \text{otherwise}\end{aligned}$$

Now, instead of using the payoff $U(t_{j+1})$ in the calculation of the performance figure π_γ, we use the attenuated payoff $\beta_\gamma(t_{j+1})\,U(t_{j+1})$.

Finally, the *appropriateness* α_γ of π_γ to the needs $N(t_j) = L_{\max} - L(t_j)$ is measured in terms of the needs left *unfulfilled* by γ's prediction, $[\pi_\gamma - N]^-$. Actually, by raising each component to a power $c_1 > 1$, yielding $[\pi_\gamma{}^{c_1} - N^{c_1}]^-$, the "demand" can be made to increase exponentially as the reservoir empties. Since $[\pi_\gamma{}^{c_1} - N^{c_1}]^-$ is always negative, we obtain as an expression for appropriateness

$$\alpha_\gamma = [\pi_\gamma{}^{c_1} - N^{c_1}]^- + c_\alpha$$

where c_α is added so that α_γ is always positive (e.g., $c_\alpha = \max\{-[\pi_\gamma{}^{c_1} -$

$N^{c_1}]^-\})$. That is, the more positive α_γ, the more appropriate classifier γ's predicted performance is to fulfilling the system's current needs.

Summarizing to this point:

The response of the system at each time step is determined by the particular configuration of activated effectors. Whether a given effector is activated or not depends upon a competition among the classifiers associated with that effector. The winner is determined by a probability that is proportional to the product of two scores:

1. The preciseness with which the classifier's taxon specifies the current detector configuration (which is a representation of the overall state, environment plus interior, confronting the system).
2. The appropriateness to the system's needs (reservoir depletions) of the classifier's predicted payoff (as measured by the average credit it has received for past payoffs).

There is one final adjustment to be made. A problem arises when the payoff attained at time t_j does not come close to the prediction of some activated classifier γ. In this case, it would be inappropriate to deactivate γ if there is some chance that γ's goal (prediction) can still be attained. More precisely, γ should not be deactivated if, taking into account γ's cumulative attenuation β_γ, a better-than-zero contribution to $\pi_\gamma(t_{j+1})$ is still predicted. Such a contribution is predicted if $[1/(\nu_\gamma + 1)]\beta_\gamma\pi_\gamma > 1$. Thus, at the end of each epoch, π_γ will be revised but γ will not be deactivated unless $[1/(\nu_\gamma + 1)]\beta_\gamma\pi_\gamma < 1$. (Note that β_γ will continue to decrease, with subsequent errors ϵ_γ, since it is not reset to 1 until γ is deactivated.)

Formal Summary: The state of a classifier γ at any time t is given by a quintuple

$$(\lambda_\gamma, \pi_\gamma, \nu_\gamma, \beta_\gamma, \eta_\gamma)$$

where

$\lambda_\gamma \in \Gamma = \{0, 1, \#\}^r$ specifies the taxon employed by γ

π_γ is the performance figure of γ revised at the end of each epoch t_{j+1} according to the recursion

$$\pi_\gamma(t_{j+1}) = \left(\frac{\nu_\gamma}{\nu_\gamma + 1}\right)\pi_\gamma(t_j) + \left(\frac{1}{\nu_\gamma + 1}\right)\frac{\beta_\gamma \cdot U(t_{j+1})}{L_1(t_j) - L_1(t_{j+1})}$$

ν_γ is the number of activations of γ over its history, revised at the end of each epoch so that

$$\begin{aligned}\nu_\gamma(t_{j+1}) &= \nu_\gamma(t_j) + 1 \qquad \text{if } \gamma \text{ deactivated}\\ &= \nu_\gamma(t_j) \qquad \text{otherwise}\end{aligned}$$

β_γ is the attenuation factor, updated by the recursion

$$\begin{aligned} \beta_\gamma(t+1) &= \beta_{1\gamma}\beta_\gamma(t) && \text{if the prediction error } \epsilon_\gamma \text{ is negative, } \beta_{1\gamma} \text{ being an ``s function'' of the prediction error} \\ &= 1 && \text{if the classifier is deactivated at } t \\ &= \beta_\gamma(t) && \text{otherwise} \end{aligned}$$

$$\begin{aligned} \eta_\gamma &= 1 && \text{if } \gamma \text{ has been activated at some prior time in the epoch} \\ &= 0 && \text{otherwise} \end{aligned}$$

Before closing this section let us look once more at the overall system and its capabilities (see Fig. 3).

The detector array presents to the classifiers a representation of the current situation. One subset of detectors is triggered by sensors which constitute the system's interface with its environment. (For example, there could be a planar array of sensors corresponding to retina, another array sensitive to audio frequencies, etc.) A second subset consists of detectors, each of which is turned on by the activation of a particular classifier. That is, if classifier γ is activated at time step t, then the corresponding detector is

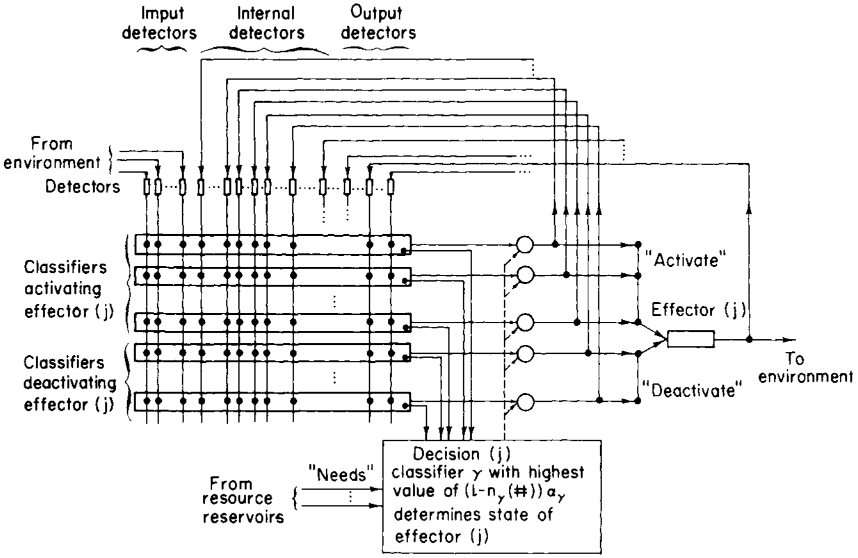


FIG. 3. Detailed diagram of a block of classifiers. Note: There is a block of classifiers (for "introspection") with no associated effector.

"on" (1) at time step $t + 1$. [This allows the system to introspect. In fact, an appropriate set of classifiers using this set of detectors can carry out any computable function up to limits imposed by the number of classifiers (the size of memory).] Finally there will be a subset of detectors, each of which is turned on by the activation of a particular effector.

Each effector has an associated *set* of classifiers competing to control its state. Because there are detectors sensitive to the state of other effectors, a given effector's activity can be made dependent (through the taxon of the controlling classifiers) on the activity of other effectors. (Thus, for example, useful coordinated sequences of effector actions can be favored.) The adaptive algorithm, to be discussed next, repeatedly modifies the set of classifiers associated with each effector, steadily increasing the average performance figure.

C. Adaptation under Sensory Guidance

At this point we have completed the description of the cognitive system's mechanisms for generating behavior and are now ready to look at the way in which the cognitive system adapts to its environment (learns) under sensory guidance. The pivotal task here is one of selecting classifier taxa with the highest level of generality compatible with consistent performance. The system must steer between two extremes: Too much generality in a taxon will yield inconsistent performance because the system gives the same response to environmental situations requiring differing responses. Too little generality in the taxon wastes system capacity on situations that recur infrequently or not at all.

Under the procedures so far outlined, a too general taxon will have a low associated performance figure because its predictions are not borne out with any consistency (i.e., high attenuation lowers its credit for any payoff achieved). We have not, however, made provision for the too specific taxon which is infrequently activated. We want to make its deletion from memory increasingly likely (presumably replacing it with a more general taxon which includes it as a special case). To provide the adaptive algorithm with the necessary information, each classifier will be assigned an *age* a_γ which increases steadily when the classifier is not used. Specifically, the age of any classifier is determined by the recursion

$$a_\gamma(t + 1) = a_\gamma(t) + 1$$

except that whenever γ is active and an appropriate payoff occurs, the age is reduced according to the formula

$$a_\gamma(t + 1) = a_\gamma(t)/[1 + \alpha_\gamma(t)/c_\alpha]$$

(That is, γ's age is halved if the payoff is maximally appropriate.) The adaptive plan must use the performance figures and ages of classifiers, which reflect the system's experience, to exert a steady selective pressure toward appropriate levels of generality.

Genetic plans, because of their intrinsic parallelism, are prime candidates for directing this selection. Before we can discuss the meaning of intrinsic parallelism in the present context, we must specify just how genetic plans can be applied to the classifiers in memory to form new, more appropriate classifiers.

The first step is the determination of a replication rate for each classifier. It is the replication rate that regulates the proportion of new classifiers, which are variations of a given classifier. The replication rate is, essentially, a product of the two factors mentioned above—the performance figure and the frequency of usage (inversely related to age). Thus, only frequently used taxa with high performance figures have a high replication rate; if *either* of these factors is low the replication rate is also low. Stated another way, the replication rate measures a taxon's success in avoiding the extremes of too much or too little generality.

The genetic plan, operating with these replication rates, exerts the required pressure toward optimal generality. In detail: At regular intervals (or at the end of each epoch) a subset of (nonactivated) classifiers is replicated for modification by the genetic operators. Each classifier to be replicated is selected from its subset (of classifiers controlling the same effector) with a probability proportional to its performance figure. The replicates are then modified by genetic operators and returned to memory, displacing extant classifiers in the process. The probability that a classifier already in memory will be so deleted is proportional to its age. In this way the size of memory is kept constant, and the net effect of the genetic plan is that of deleting old, infrequently used classifiers to make room for variants of highly rated classifiers. Replication in proportion to the performance figure combined with deletion according to age gives each classifier the desired replication rate. [Lemma 6.1 of Holland (1975) gives a rigorous version of this statement.]

A genetic algorithm exhibits intrinsic parallelism because, as it generates new classifiers, it continually favors *all* combinations of alleles which have, in the past, been associated with classifiers exhibiting above-average replication rates. From a statistical viewpoint, the genetic algorithm generates its trials (i.e., new classifiers) in one space, but is being guided by sample averages of events associated with a much more numerous space, the space of all combinations of alleles. A more careful version of this statement is based upon the comparison of taxa in terms of the alleles they have in common. Specifically, if a pair of classifiers have the same alleles at loci

$i_1, i_2, \ldots, i_c$ in the taxon then, by definition, they belong to a particular hyperplane on those dimensions. That is, this hyperplane *is* the set of all taxa which have the given alleles at each of the given loci. All of its members are similar to that extent. Each hyperplane is formally named (as indicated in Section I, C) by using a □ to indicate those loci at which it is irrelevant which alleles occur, while the names of the alleles which must occur are inserted at the other loci. Thus 1□#0□□. . .□ designates the hyperplane consisting of every taxon having allele 1 at locus 1, allele # at locus 3, and allele 0 at locus 4.

It should be clear that, at any given time, memory will contain instances (that is, elements) of a great many hyperplanes. In fact, any given classifier belongs to 2^l hyperplanes (where l is the number of loci used to define the taxon). Overall, then, memory will contain instances of around $(M/2)\cdot 2^l$ hyperplanes, where M is the number of classifiers in memory. With $l = 100$ detectors and $M = 1000$ classifiers in memory, this is on the order of 10^{32} hyperplanes! Moreover with *each* of these hyperplanes we can form an estimate of its usefulness in deriving new classifiers by associating with it the average replication rate of its instances in memory. From the viewpoint of statistics, a hyperplane is an event in the sample space of taxa, and the average value of the replication rate over this event is a measure of the usefulness of samples drawn from it. If this average is high, then it is a reasonable inference (subject to further confirmation) that other taxa drawn from the same hyperplane will have high replication rates. This suggests that the probability of generating classifiers which belong to this hyperplane should be increased, both to increase overall performance and to add confirmation to the inference.

Intrinsic parallelism, in these terms, occurs when a succession of trials of classifiers serves to vary appropriately the sampling rates of the large numbers of hyperplanes with instances in memory. Specifically, genetic algorithms act so that the proportion $P_\xi{}^T$ of classifiers drawn from a given hyperplane ξ at replication time T changes approximately according to the recursion

$$P_\xi^{T+1} = (\hat{\mu}_\xi/\hat{\mu})P_\xi{}^T$$

where $\hat{\mu}_\xi$ is the average observed replication rate of the instances of ξ in memory at T, and $\hat{\mu}$ is the average replication rate of all classifiers in memory at T. [The rigorous counterpart of this statement is given by Lemma 7.2 of Holland (1975), which parallels the results of Sections I, D and I, E here.] In words: If the hyperplane ξ has an observed average replication rate $\hat{\mu}_\xi$ above the average $\hat{\mu}$ for the whole of memory, then ξ will be sampled at an exponentially increasing rate (until the overall average $\hat{\mu}$ is brought up to $\hat{\mu}_\xi$). Because the genetic algorithms are stochastic, this is only the *expected* sampling rate; moreover, because of the nature of the genetic operators, this expectation generally holds only for hyperplanes with two or more instances in memory. Still a very large number of hyperplanes have their individual sampling rates adjusted according to the average performance of their instances, and at no great computational expense. In the example of 100 detectors and 1000 classifiers cited earlier, at least 10^7 hyperplanes would be sampled at rates determined by the above recursion, even though only a few hundred classifiers are tried between replication times! The claims made for this cognitive model stem directly from the tremendous gain in flexibility and quickness in learning so achieved.

One detail remains to be handled—the initial state assigned to a newly generated classifier. With the addition of age a_γ, each classifier is now specified by a sextuple

$$(\lambda_\gamma, \pi_\gamma, \nu_\gamma, \beta_\gamma, \eta_\gamma, a_\gamma).$$

λ_γ, the taxon, is specified by the genetic algorithm. π_γ, the performance figure, will be taken to be the maximum performance figure of the parents of γ since, if it is overoptimistic, it will quickly be reduced at subsequent payoffs. ν_γ, the number of prior activations, is set to zero; β_γ, the cumulative attenuation, is set to one; η_γ, the activation indicator is, of course, zero; and a_γ is set to the average age of the classifiers controlling the same effector as γ (so that the new classifier has no particular advantage with respect to deletion).

Where it is desirable, the cognitive system can be made particularly sensitive to surprising situations (situations characterized by a large prediction error). One simply enters the corresponding detector configuration in the appropriate set of classifiers, as a taxon λ without any #'s, deleting an aged classifier to make room for it. This new classifier is then assigned a performance figure π corresponding to the payoff actually achieved at the end of that epoch, and all other parameters are assigned as above. "Curiosity" can be implemented by introducing a reservoir which only receives increments when surprising situations are encountered and which only acts as a need when the other reservoirs are reasonably full.

Summarizing: The set (or population) of classifiers associated with each effector is continually being adapted to the system's experience by means of the genetic algorithm. Because of this algorithm, the set of classifiers actually stores the relative rankings of a tremendous number of components (schemata) from which taxa can be constructed. Components associated with above-average performance receive above-average rankings and are tested in new combinations with probabilities proportional to these rankings. When the performance figure of any given classifier is revised a great many rankings are updated because of the genetic algorithm's *intrinsic parallelism* with respect to the rankings. Moreover, new components are continually being generated and tested—all of this while the overall average performance figure is being optimized. Overall there is a steady, intrinsically parallel selection pressure for classifiers as general as possible within the constraint that each predict, with reasonable consistency, the average payoff to be expected if it is activated. It is worth noting that the resulting set of classifiers is at once the system's memory and its information-processing repertoire, making the system a highly distributed processor. Even after massive damage (deletions, etc.) this processor can quickly regenerate its repertoire, because it is the *number* of instances of a schema in memory which reflects its relative rank.

The interaction of this procedure for adaptation with the controlling equations for the classifiers yields a cognitive system of unusual flexibility. Experience continually revises classifications of detector configurations and the predicted appropriateness (in terms of payoff) of responding according to a given classification. The net effect is a progressive integration of responses, so that each input leads to a response building on preceding responses and so that the whole response sequence consistently approaches the same goal. "Internal" detectors turned on by previous activations of classifiers and effectors make this possible. By paying attention to the internal state of the system they allow "introspection" and sophisticated computations based on prior actions and outcomes. Intrinsic parallelism assures that each experience has extensive effects on the revision of classifiers and predictions, simultaneously modifying the trial probabilities (rankings) of large numbers of components (schemata).

With appropriate selections of detectors, the system can perform appropriately in a wide variety of situations. Two tasks representative of this range are:

1. *Game-playing and searches.* Here the environment consists of a graph with labeled nodes ("board configurations" in the case of board games) and a payoff function over the nodes (assigning the payoff to "terminal" configurations in the case of games). Responses choose among edges leading out from a node (legal moves for games). The detector array must provide a distinct configuration of detector values for each node. (For example, for board games, there could be a detector for each board square, reporting the piece there). In order to move appropriately the system must accomplish a two-fold task. It must learn appropriate classifications for the nodes [such as a count of "pieces ahead," cf. the parameters in Samuel's (1959) checker player]. And it must associate appropriate responses (moves) with these classifications. Classifiers associated with a null response can be developed as internal processors to carry out sophisti-

cated computations such as counts of pieces. [For example, a particular classifier, say λ_1, can be turned on when the system is supposed to be paying attention to square j of a board game. If some other classifier, say λ_2, signals the end of computations with respect to square j, then the detector values corresponding to "λ_1 on" and "λ_2 on" can be a part of the taxon for the classifier which is to be on when the system is paying attention to square $j + 1$. The overall procedures are similar to those discussed in connection with the broadcast language in Holland (1975).] For simply defined graphs with a payoff function, several interesting "convergence" theorems can be proved. (See Table I.)

2. *Sophisticated language processing.* It is easy to provide classifiers having taxa that correspond to semantic units (i.e., taxa that more or less regularly distinguish entities, qualities, actions, etc. of interest in the environment). Then (through construction or learning) the system can acquire classifiers that serve as "semantic formulas" [as in Wilks's (1975) Preference Semantics]. That is, these classifiers are sensitive (via the "internal" detectors) to preferred combinations of the "semantic unit" classifiers. The competition for activation among these "semantic formula classifiers" corresponds closely to a selection of the best description of the current situation in terms of the formulas' preferences. Higher levels (templates, stereotypes, etc.) can be handled in similar ways. The amount of structure given the cognitive system initially, in terms of initially specified classifiers corresponding to specific semantic units, formulas, etc., varies inversely with interest in the system's ability to learn these structures. Because the classifier "language" (with "internal" detectors) is a complete programming language, as large a fragment as desired can be given the system initially.

Given some general receptors (a vision raster or other artificial retina) and some general built-in response sequences (initially specified interacting classifiers), the cognitive system should be able to *learn* a broad and flexible behavioral repertoire. This, of course, depends upon a suitable "teaching" environment and a suitable reinforcement schedule. However, the intrinsic parallelism of the adaptive plan gives the system the possibility of accomplishing a great deal in reasonable amounts of time. If, in addition, the system is supplied with some *extrinsic* parallelism so that all the classifiers can be tested simultaneously, learning in "physiological real time" seems possible.

In developing this application of adaptive plans, Hebb's characterization of behavior as "adaptation to the environment under sensory guidance" has been taken quite seriously. The object has been to produce a conceptually general and rigorously defined cognitive system capable of *learning*

TABLE I

FRAGMENT OF A GAME-PLAYING VERSION OF THE COGNITIVE SYSTEM

DETECTORS

Label		Property detected
δ_u	$= 0$	if there is no piece on square q
	$= 1$	if there is a black piece on square q
	$= 2$	if there is a white piece on square q
δ_{u+1}	$= 0$	if there is no piece on square $q + 1$
	$= 1$	if there is a black piece on square $q + 1$
	$= 2$	if there is a white piece on square $q + 1$
δ_v	$= 0$	if classifier γ_h was not active on the previous time step
	$= 1$	if classifier γ_h was active on the previous time step
δ_{v+1}	$= 0$	if classifier γ_{h+1} was not active on the previous time step
	$= 1$	if classifier γ_{h+1} was active on the previous time step
δ_w	$= 0$	if classifier γ_i was not active on the previous time step
	$= 1$	if classifier γ_i was active on the previous time step
δ_{w+1}	$= 0$	if classifier γ_{i+1} was not active on the previous time step
	$= 1$	if classifier γ_{i+1} was active on the previous time step

CLASSIFIERS

Label	Taxon	"Meaning"
	$\delta_1 \cdots \delta_{u-1}\delta_u\delta_{u+1} \cdots \delta_{v-1}\delta_v\delta_{v+1} \cdots \delta_{w+1}\delta_w\delta_{w+1} \cdots \delta_m$	
γ_h	#. . .###. . .1##. . .###. . .#	Scan square q
γ_{h+1}	#. . .###. . .#1#. . .###. . .#	Scan square $q + 1$
γ_i	#. . .#1#. . .#1#. . .1##. . .#	Cumulative total of black pieces is b after scan of square q
γ_{i+1}	#. . .##1. . .##1. . .#1#. . .#	Cumulative total of black pieces is $b + 1$ after scan of square $q + 1$

appropriate behavior in complex environments. (It is my feeling that a cognitive system can exhibit versatile "intelligence" only if learning is an *integral* part of the overall system. A system that repeats the same mistake *ad nauseum,* when that mistake is simply defined in terms of the system's capabilities, can hardly be called intelligent.) Learning in complex environments has been a refractory subject with rigor and quantification hard to come by. The framework laid out here is a new approach. Though there is much to be done, the proposed approach has some interesting qualifications. It draws on several disciplines and can integrate distinctive insights from each. In the computer sciences it makes contact with studies ranging from those of Samuel (1959) (game-playing) through cluster algorithms for pattern recognition [see Sokal (1974)] to those of Wilks (1975) (semantics). The cognitive systems are rigorously defined by means of recursive equations and systems of this type can be easily programmed for simulation. Thus, questions about the model's capabilities can be subjected to test, rather than resting on the outcome of intuitive arguments. Because the learning procedure is a genetic algorithm, the body of theorems about adaptive plans (Section I) can be brought to bear. This opens the possibility of *proving* bounds on the expected time for the system to acquire optimal response sequences for idealized environmental state graphs with assigned payoffs. With a little care, the particular cognitive system selected for this analysis can be designed so that it can also handle much more complex environments. By simulating and running it in a progression of more complex environments (up to and including real world situations), comparisons can be made between the theory for idealized situations and actual behavior in a complex situation. Finally the model has counterparts of and speaks to several parts of psychology ranging from Hubel and Wiesel's (1968) considerations concerning detectors, through Hebbian (Hebb, 1958) cell-assembly notions *vis-à-vis* the classifiers, to behavioral comparisons with recent versions of the beta models of Luce (1959).

In the long run, by offering an unusual and demanding context for genetic plans, the models should substantially increase our general understanding of adaptation.

REFERENCES

Crow, J. F., and Kimura, M. (1970). "An Introduction to Population Genetics Theory." Harper & Row, New York.

Fisher, R. A. (1930). "The Genetical Theory of Natural Selection." Oxford Univ. Press (Clarendon), London and New York.

Hebb, D. O. (1958). "A Textbook of Psychology." Saunders, Philadelphia, Pennsylvania.

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

Holland, J. H., and Reitman, J. S. In preparation.

Hubel, D. H., and Wiesel, T. N. (1968). *J. Physiol.* (*London*) **195**, 215–243.

Luce, R. D. (1959). "Individual Choice Behavior." Wiley, New York.

Moran, P. A. P. (1962). "The Statistical Processes of Evolutionary Theory." Oxford Univ. Press (Clarendon), London and New York.

Samuel, A. L. (1959). *IBM J. Res. Dev.* **3**, 210–229.

Sokal, R. R. (1974). *Science* **185**, 1115–1123.

Wallace, B. (1968). "Topics in Population Genetics." Norton, New York.

Wilks, Y. (1975). *Commun. ACM* **18**, 264–274.