6. Reproductive Plans and Genetic Operators

In the earlier informal discussion of genetics (sections 1.4 and 3.1) reproductive plans were introduced as the fundamental procedure of genetic adaptation. The present chapter lifts reproductive plans from the specific context of genetics to the general framework of chapter 2. This, at one stroke, makes reproductive plans suitable objects for rigorous study *and* yields a class of plans applicable to the full range of adaptive systems. Genetic plans, i.e., reproductive plans using generalized genetic operators, will be the prime focus; emphasis will be laid upon the operators' retention and use of relevant history as they exploit opportunities for improved performance.

Genetic plans can be applied to any domain of structures $\mathfrak{A}$ represented by strings (l-tuples). (To build a better intuition for this flexibility the reader may find it useful to consistently interpret the properties and theorems advanced here in the most familiar of the nongenetic illustrations of chapter 3.) We will see that *each* structure generated and tested by a genetic plan in effect tests a multitude of schemata and that the plan actually preserves and exploits this information. Genetic plans do this by generating sequences of structures in such a way that, once a few instances of *any* given schema ξ occur, one can count on the cumulative number of instances of ξ increasing at a rate closely related to μ_ξ. The generalized genetic operators act so as to test old schemata in new contexts, generate instances of schemata not previously tested, and so on (see sections 7.2–7.5), without disturbing the rates of increase. Genetic plans thus exhibit the intrinsic parallelism discussed at the ends of chapters 4 and 5.

Interpreted in genetics, the results of the next two chapters indicate that adaptation proceeds largely in terms of pools of coadapted sets of alleles rather than gene pools. As one important offshoot, this approach yields an extension of Fisher's (1930) classical result (on the rates of increase of alleles) to coadapted sets of alleles with epistatic interaction (see section 7.4). A typical interpretation for artificial systems can be obtained by looking again at the function $f(x)$ of Figure 10.

We see that the average value $\mu_{1\square\square\ldots\square}$ of all points in the schema $1\square\square\ldots\square$ (i.e., the area under the curve over the interval $\frac{1}{2} \leqq x < 1$ divided by $\frac{1}{2}$, the length of the interval) is approximately 1.5. Similarly, for $\square\square 0\square\ldots\square$ the value is approximately 1, for $1\square 0\square\ldots\square$ the value is approximately 2, etc. Thus instances of $1\square\square\ldots\square$ will accumulate at a higher rate than those of $\square\square 0\square\ldots\square$, and instances of $1\square 0\square\ldots\square$ will accumulate still more rapidly. The result is an ever greater clustering of test points (instances) in intervals (schemata) of above-average value (see Figure 13 and the example of section 7.3). In this way the genetic plan locates a global optimum of $f(x)$, exploiting false peaks (without entrapment) to rapidly increase the average value of points tested.

We will see that genetic plans act with a combination of simplicity and subtlety both pleasing to the eye and useful in application. They also act with robustness and efficiency, a fact that will be finally established in the next chapter. It should be emphasized that the plans (algorithms) set forth have a dual role. When the plan's parameter values (and functions) are determined from data about a particular natural process, the plan serves as an idealized model or hypothesis about that process. As such it is subject to the general observation-modification cycle applicable to physical theories in general. Because the model is already in algorithmic form, it is particularly suitable for simulations of the process. The other role occurs in relation to artificial (designed) processes. Here the plans serve as optimization procedures which can be fitted into the process to control its direction. In either role the theorems proved hereafter yield predictions which must come true if (for the natural systems) the basic model is verified or (for artificial systems) the algorithm is incorporated as a control.

1. GENERALIZED REPRODUCTIVE PLANS

To embed reproductive plans in the $(\mathfrak{I}, \mathcal{E}, \chi)$ framework of chapter 2 we must define a class of plans (algorithms) applicable to an arbitrary set of structures $\mathfrak{A}$. Moreover each plan must be a mapping of the form $\tau: I \times \mathfrak{A} \to \Omega$. It must use only the input from the environment, $I(t)$, and the structure tried at time t, $\mathfrak{A}(t)$, to determine a random variable over $\mathfrak{A}$, $\omega_t(\mathfrak{A}(t))$, which is in turn sampled to determine the next trial, $\mathfrak{A}(t+1)$. We will begin by defining a relatively narrow class of reproductive plans $\mathfrak{R}_1$. Later $\mathfrak{R}_1$ will be extended in ways which make some applications more natural, and we will see that the new algorithms are essentially no more powerful than those from $\mathfrak{R}_1$.

To begin let $\mathfrak{A}_1$ be the set of structures to be tested and, as in chapter 4, assume that the elements of $\mathfrak{A}_1$ are representations. (As long as each structure is

represented by a finite string of attributes, $\mathfrak{A}_1$ can be made countably infinite without affecting the presentation of $\mathfrak{R}_1$. This will be discussed in chapter 8.) Each plan in $\mathfrak{R}_1$ is an algorithm which acts at each instant t upon a small *set* of structures $\mathfrak{B}(t)$ from $\mathfrak{A}_1$ (interpretable, for instance, as a population or data base). The algorithm uses a single basic cycle to modify elements of the small set, one at a time, thereby producing a sequence of new structures for trial. In general terms, the basic steps of the cycle are:

1. Select one structure from $\mathfrak{B}(t)$ probabilistically, after assigning each structure a probability proportional to its observed performance.
2. Copy the selected structure, then apply operators to the copy to produce a new structure.
3. Select a second element from $\mathfrak{B}(t)$ at random (all elements equally likely) and replace it by the new structure produced in step 2.
4. Observe and record the performance of the new structure.
5. Return to step 1.

Note that the *number* of elements in $\mathfrak{B}(t)$ remains constant. (From the point of view of genetics, it is convenient to look upon the size of $\mathfrak{B}(t)$ as an upper bound on population size determined, say, by the "carrying capacity" of the environment.) The number of structures in $\mathfrak{B}(t)$ can be varied up to the maximum number by allowing null structures or vacancies.

With this outline as a guide, we can now go on to the rigorous definition of the algorithms in $\mathfrak{R}_1$. The following symbols and definitions will be used with the interpretations given:

$\mathfrak{A}_1$,	the set of basic structures being tested.
$\mathfrak{A}_1^M$,	the set of all M-tuples of structures corresponding to possible compositions of $\mathfrak{B}$.
$\mathfrak{B}(t)$,	the particular set of M structures $\{A_1(t), A_2(t), \ldots, A_M(t)\}$ available to the adaptive plan at time t.
$\mathfrak{I}_M = \{1, 2, \ldots, M\}$,	the first M positive integers, used as an index set for $\mathfrak{B}$.
Ω,	the set of stochastic operators for modifying structures.
$\mathfrak{I}_M \times \mathfrak{A}_1^M$,	compositions of $\mathfrak{B}$ with one structure selected (for modification by an operator); i.e. $(i, A_1(t), \ldots, A_M(t)) \in \mathfrak{I}_M \times \mathfrak{A}_1^M$ corresponds to $\mathfrak{B}(t)$ with the ith structure, $A_i(t)$ selected.

$\mathcal{P}$,	a set of probability distributions over $\mathfrak{A}_1$, one of which is selected by each application of a stochastic operator $\omega \in \Omega$.
$\rho: \mathfrak{A}_1 \rightarrow \Omega$,	assigns to each basic structure $A \in \mathfrak{A}_1$ the stochastic operator $\omega \in \Omega$ which is to be used to modify A.
$\omega: \mathfrak{I}_M \times \mathfrak{A}_1^M \rightarrow \mathcal{P}$,	an arbitrary operator from Ω which determines, from $\mathfrak{B}(t)$ and a selection $i(t)$, a distribution $P \in \mathcal{P}$ over $\mathfrak{A}_1$.

Once the set of structures $\mathfrak{A}_1$ has been given, along with an observation procedure which assigns a payoff $\mu_E(A)$ to each trial of a structure $A \in \mathfrak{A}_1$, a reproductive plan of type $\mathfrak{R}_1$ is determined by specifying the functions ρ and $\{\omega\}$. The algorithm proceeds as follows:

1 Set $t = 0$ and initialize $\mathfrak{B}$ by selecting M structures at random from $\mathfrak{A}_1$ to form $\mathfrak{B}(0) = \{A_h(0), h = 1, \ldots, M\}$.

2.1 Observe and store the performances $\{\mu_E(A_h(0)), h = 1, \ldots, M\}$.

2.2 Observe the performance of $A'(t)$ and replace $\mu_E(A_{j(t)}(t))$ by $\mu_E(A'(t))$.

3 Increment t by 1

4 Select one structure $A_{i(t)}(t)$ from $\mathfrak{B}(t)$ by taking one sample of $\mathfrak{B}(t)$ using the probabilities $\text{Prob}(A_h(t)) = \mu_E(A_h(t)) / \sum_{h'=1}^{M} \mu_E(A_{h'}(t))$, $h = 1, \ldots, M$.

5 Determine the operator $\omega_t \in \Omega$ to be applied to $A_{i(t)}$, $\omega_t = \rho(A_{i(t)}(t))$, and then use ω_t to determine a new structure $A'(t)$ by taking a sample of $\mathfrak{A}_1$ according to the probability distribution $P_t = \omega_t(i(t), A_1(t), \ldots, A_M(t)) \in \mathcal{P}$.

6 Assign probability $1/M$ to each number $1, \ldots, M$, select one number $1 \leqq j(t) \leqq M$ accordingly, and replace $A_{j(t)}(t)$ by $A'(t)$.

Algorithms of type $\mathfrak{R}_1$ are strictly sequential in the sense that one individual $A'(t)$ is tested at a time. $\mathfrak{B}(t)$ serves as a reservoir of information about the environment and as a basis for generating new trials. $\mathfrak{B}(t)$ remains constant in size because each new individual $A'(t)$ replaces an individual already in the population. Under the operators Ω of interest (particularly the generalized genetic operators), $A'(t)$ can be looked upon as the "offspring" of $A_{i(t)}(t)$, retaining many (but generally not all) of the attributes of $A_{i(t)}(t)$. Via the function ρ each structure in the population carries a specification of the operator appropriate to it (a kind of "species" designation). (The apparent generalization to stochastic selection of one of a *set* of

operators can actually be subsumed in the stochastic selection of offspring. See below.) The operators are computation procedures using random numbers; generally, they use at most one other member of the population, in addition to $A_{i(t)}(t)$, in the determination of $A'(t)$. (For instance, the operator may randomly select a "mate" for $A_{i(t)}(t)$.) The argument of each $\omega \in \Omega$ includes the whole population, because any structure in the population is a conceivable candidate for the second operand, even when ω is essentially a binary operator. (E.g., the probable outcomes of a "mating" will depend upon the range of "mates" available.)

It should be noted that the *state* of the algorithm at the beginning of any cycle includes not only the population $\mathfrak{A}(t)$, but also the retained performances $\mu_E(A_h(t))$, $h = 1, \ldots, M$, of the structures in $\mathfrak{A}(t)$. Thus, in the general formalism of chapter 2,

$$\mathfrak{A} = \mathfrak{A}_1^M \times [0, r]^M$$

where $[0, r]$ is the interval of possible payoffs (performances), i.e. $[0, r]$ is the range of μ_E,

$$\mu_E : \mathfrak{A}_1 \rightarrow [0, r].$$

Accordingly,

$$\mathfrak{A}(t) = (A_1(t), \ldots, A_M(t), \mu_E(A_1(t)), \ldots, \mu_E(A_M(t))).$$

The *new* information $I(t)$, from the environment $E \in \mathcal{E}$ at each time t, is simply the payoff $\mu_E(A'(t))$ of the new structure $A'(t)$. Thus any adaptive plan $\tau \in \mathfrak{R}_1$ has the required form

$$\tau : I \times \mathfrak{A} \rightarrow \mathfrak{A}$$

since

$$\tau(\mu_E(A'(t)), [A_1(t), \ldots, A_M(t), \mu_E(A_1(t)), \ldots, \mu_E(A_M(t))]) = [A_1(t+1), \ldots, A_M(t+1), \mu_E(A_1(t+1)), \ldots, \mu_E(A_M(t+1))].$$

Informally, a reproductive plan is one under which the better an individual performs the more offspring it has. For plans $\tau \in \mathfrak{R}_1$ a precise counterpart of this property can be established with the help of the following

LEMMA 6.1: *If, at any time-step, p_1 is the probability that a structure A produces an "offspring" during that time-step and p_2 is the probability that A is deleted during that time-step, then the expected number of "offspring" of A is p_1/p_2.*

Proof: This is immediately established by noting that, when p_1 and p_2 are constant, the expected lifespan of A is $1/p_2$ and the expected number of offspring is simply the number of offspring expected during the expected lifespan, i.e., p_1/p_2. In more detail, the probability of A surviving for *exactly* T time-steps is $p(T) = (1 - p_2)^{T-1} \cdot p_2$, and the expected number of "offspring" during that interval is $\bar{u}_A(T) = p_1 T$. Thus, the expected number of "offspring" during A's lifespan is

$$\sum_{T=1}^{\infty} p(T)\bar{u}_A(T) = p_1 p_2 \sum_{T=1}^{\infty} T(1 - p_2)^{T-1}.$$

But $\sum_{T=1}^{\infty} T(1 - p_2)^{T-1}$ converges to $(1/p_2)^2$ (as may be easily verified by taking the derivative of both sides of the identity $(1/1 - x) = 1 + x + x^2 + \ldots$). Therefore

$$\sum_{T=1}^{\infty} p(T)\bar{u}_A(T) = p_1/p_2. \qquad \text{Q.E.D.}$$

For plans in $\mathcal{R}_1$ the interpretation of this lemma is direct: The probability of A_h being selected to produce an offspring A' during time t is $\mu_{ht}/\sum_{h'}\mu_{h't}$ where $\mu_{h't} = {}^{df}\cdot\mu_B(A_{h'}(t))$, while the probability of A_h being deleted at the end of that time-step is $1/M$. Hence, if $\sum_{h'}\mu_{h't}$ changes negligibly over A_h's lifespan, the expected number of offspring is

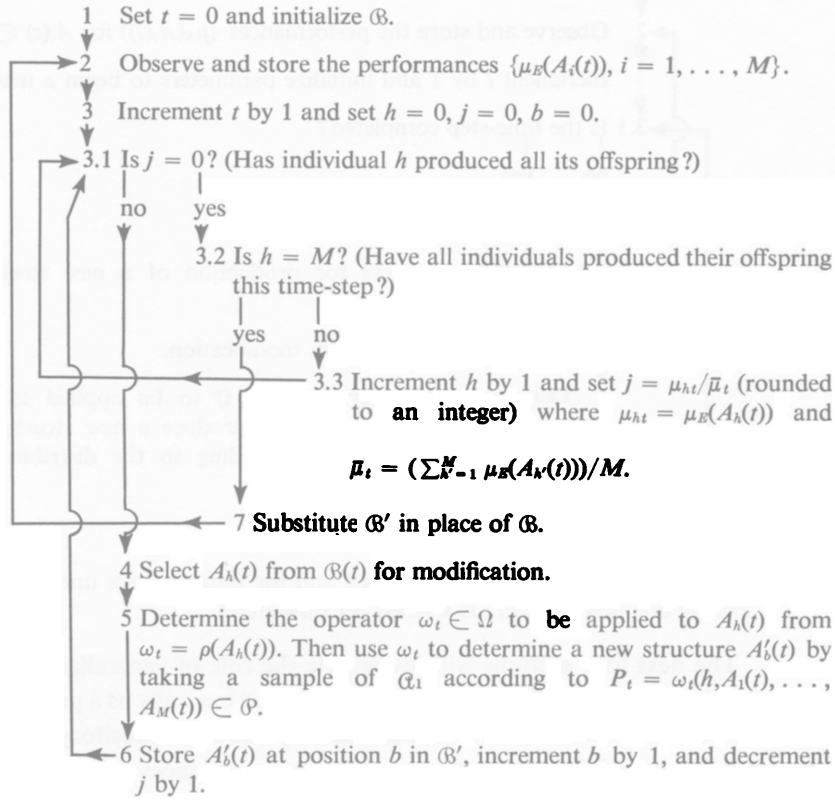
$$(\mu_{ht}/\textstyle\sum_{h'}\mu_{h't})/(1/M) = \mu_{ht}/(\sum_{h'}\mu_{h't}/M) = \mu_{ht}/\bar{\mu}_t.$$

$\mu_{ht}/\bar{\mu}_t$ can be looked upon as a "normalized" payoff, the "usefulness" of A_h being measured relative to the average performance of the other members in the population. With this arrangement the expected number of offspring of A_h is greater than 1 just in case A_h's performance is above average. Since $\bar{\mu}_t$ is *not* stationary for plans $\tau \in \mathcal{R}_1$, the probability p_1 does not in fact remain constant (though, over the expected lifespan of a structure, it will not often change greatly). If $\bar{\mu}_t$ increases (as it will generally with a good plan), then A_h will receive fewer offspring than predicted by the calculation of p_1 at the time A_h originated. That is, the performance of A_h looks less promising relative to the current average, so trials of A_h are curtailed. If $\bar{\mu}_t$ decreases, the opposite effect occurs. Still, the expected number of offspring varies in direct relation to A_h's relative performance, so that plans in $\mathcal{R}_1$ satisfy the (informal) characterization of reproductive plans.

A slight change in the form of the algorithms in $\mathcal{R}_1$ yields a class of algorithms $\mathcal{R}_d$ wherein a time-step is a "generation" during which *each* individual $A_h(t) \in \mathcal{B}(t)$ is replaced, deterministically instead of as an expectation, by $\mu_{ht}/\bar{\mu}_t$ offspring. Thus, for $\mathcal{R}_d$, $\mathcal{B}(t+1)$ consists of the set of *all* offspring of the individuals in $\mathcal{B}(t)$. (To keep the population level at M individuals a special kind of rounding

procedure must be used to handle the cases where $\mu_{ht}/\bar{\mu}_t$ involves a fraction so that the roundings of the fractions $\mu_{ht}/\bar{\mu}_t$ sum to zero, but this need not concern us here.)

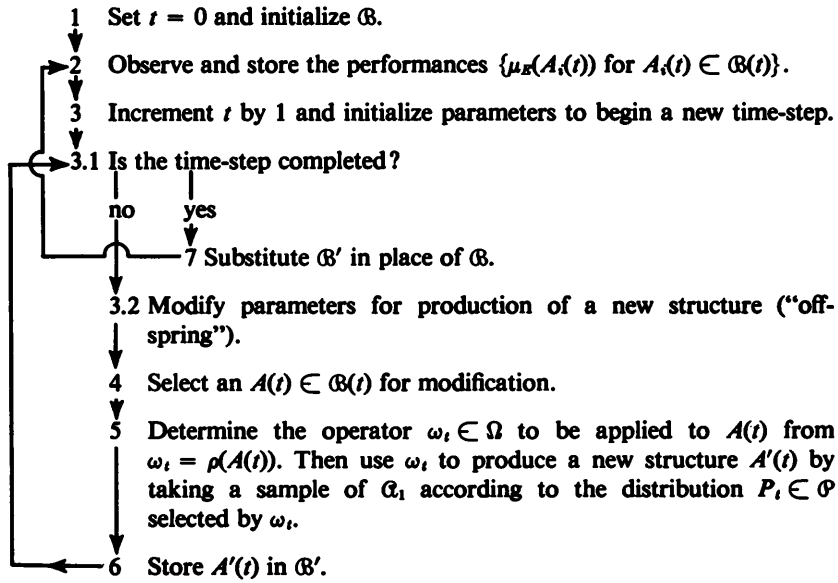
(h is the index of the individual currently producing offspring. j is a count [down] of the number of offspring produced by individual h, and b is a cumulative count of the number of offspring.)



Algorithms in the class $\mathcal{R}_d$ are closer to some of the "deterministic" models of mathematical genetics. It is easier, in some respects, to interpret the role of the population $\mathcal{B}(t)$ in these plans than it is for the strictly sequential, stochastic plans in $\mathcal{R}_1$. On the other hand the algorithms in $\mathcal{R}_1$ look more like the "one-point-at-a-time" algorithms of numerical analysis and computational mathematics. Though $\mathcal{R}_1$ and $\mathcal{R}_d$ behave similarly, it is useful to have both in mind, translating from one to the other as it aids understanding.

For both types of plan the operators brought into play in step 5 are critical

in determining just how past history is stored and exploited. The examination of specific operators can be expedited by subsuming $\mathfrak{R}_1$ and $\mathfrak{R}_d$ in a single overall diagram. Plans which satisfy this diagram and retain a recognizable variant of the "reproduction according to performance" procedures in $\mathfrak{R}_1$ or $\mathfrak{R}_d$ will be called plans of type $\mathfrak{R}$.



(In $\mathfrak{R}_1$ steps 6 and 7 are amalgamated and the tests in 3 are unnecessary because exactly one new structure is formed per time-step.)

The next four sections will investigate the role of generalized genetic operators in plans of type $\mathfrak{R}$. We will see that $\mathfrak{B}(t)$ is used basically as a pool of schemata. (Recall from chapter 4 that this means that $\mathfrak{B}(t)$ acts as a repository for somewhere between 2^l and $M \cdot 2^l$ schemata; i.e., it contains instances of this many distinct schemata.) Past history is recorded in terms of the ranking (number of instances) of each schema in $\mathfrak{B}(t)$, much as discussed at the end of chapter 5. From this point of view *crossing-over* acts to generate new instances of schemata already in the pool while simultaneously generating (instances of) new schemata (see section 6.2). In general a total of 2^l schemata will be affected by *each* crossing-over (see Lemma 6.2.1). *Inversion* (section 6.3) affects the pool of schemata by changing the linkage (association) of alleles (attributes) defining various schemata. In combination with reproduction, the net effect is to increase the linkage of schemata of high rank

(coadapted sets of alleles), making such schemata less subject to decomposition. *Mutation* (section 6.4) generally has a background role, supplying new alleles or new instances of lost alleles. All of this goes on without seriously disturbing the intrinsic rates of increase $\{\mu_\xi\}$ of most schemata instanced in $\mathfrak{B}(t)$. Chapter 7 establishes the robustness and intrinsic parallelism of these type $\mathfrak{R}$ plans for arbitrary string-representable domains $\mathfrak{A}$.

2. GENERALIZED GENETIC OPERATORS—CROSSING-OVER

When genetic operators are used with reproductive plans we get a surprisingly sophisticated set of adaptive plans. Like the rules of a well-constructed game (chess, go, poker), genetic operators are simply defined but subtle in their consequences.

Our first objective, as with reproductive plans, will be to lift genetic operators from their specific biological context to the general $(\mathfrak{I}, \mathfrak{E}, \chi)$ framework. With the help of this framework we can then define and investigate rigorously two critical advantages (first discussed in chapter 4) conferred by genetic operators:

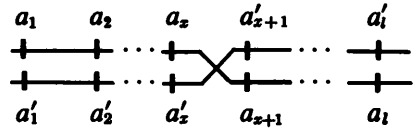
(i) intrinsic parallelism in the testing and exploitation of schemata, and
(ii) compact storage and use of the large amounts of information resulting from prior observations of schemata.

This contrasts with the common view of evolutionary processes as successive selection of the best of a sequence of variants produced by mutation—a process which we will see amounts to an enumeration of structures, with its attendant disadvantages.

The reader should be warned that the generalized operators presented in the next three sections are idealized to varying degrees. This has been done to emphasize the basic functions of the operators, at the cost of exploring the complex (and fascinating) biological mechanism underlying their execution. Even so an attempt has been made to keep the correspondence close enough to allow ready translation of the results to the original biological context.

Because it serves well as a paradigm for other genetic operators, we will look first at "crossing-over." 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 providing the structures with representations as in chapter 4. As before, for simplicity, $\mathfrak{A}$ will be taken to *be* the set of representations. Besides facilitating the generalization to arbitrary structures this emphasizes the effects of crossing-over on schemata. Crossing-over proceeds in three steps.

1. Two structures, $A = a_1a_2 \ldots a_l$ and $A' = a'_1a'_2 \ldots a'_l$, are selected (usually at random) from the current population $\mathfrak{B}(t)$. (a_i and a'_i are elements of the set of attribute values V. Hence, if A_0 is the basic structure prior to representation, $\delta_i(A_0) = a_i$. Again $a_1a_2 \ldots a_l$ abbreviates $(a_1, a_2, \ldots, a_l)$, etc.)
2. A number x is selected from $\{1, 2, \ldots, l-1\}$ (again at random).
3. Two new structures are formed from A and A' by exchanging the set of attributes to the right of position x, yielding $a_1 \ldots a_x a'_{x+1} \ldots a'_l$ and $a'_1 \ldots a'_x a_{x+1} \ldots a_l$.



(To incorporate crossing-over directly into plans of type $\mathfrak{R}$ one of the resultant structures is discarded.)

The quickest way to get a feeling for the role crossing-over plays in adaptation is to look at its effect upon schemata. To do this, consider $\mathfrak{B}(t)$ as a pool of schemata (following the suggestions of chapter 4) where the number $M_\xi(t)$ of instances of ξ in $\mathfrak{B}(t)$ reflects ξ's current "usefulness." The two direct effects of crossing-over on this pool are:

1. Generation of new instances of schemata already in the pool. E.g., $A = a_1a_2 \ldots a_l$ is an instance of the schema $a_1a_2 \square \ldots \square$ and, after crossing-over with $A' = a'_1a'_2 \ldots a'_l$, we have a new instance of $a_1a_2 \square \ldots \square$, namely $a_1a_2 \ldots a_xa'_{x+1} \ldots a'_l$ (assuming $a_i \neq a'_i$ for some $i \geqq x$). Each new instance of a schema ξ amounts to a new trial of the random variable corresponding to ξ. As such it increases the likelihood that the *observed* average performance μ_ξ of the instances of ξ closely approximates the expectation μ_ξ of the random variable ξ.
2. Generation of new schemata (i.e. schemata having neither A nor A' as an instance). E.g., after the crossing-over of A with A' the schema $\square \ldots \square a_xa'_{x+1} \square \ldots \square$ has an instance, though neither A nor A' are instances of it (if $a_x \neq a'_x$ or $a_{x+1} \neq a'_{x+1}$). Thus $\square \ldots \square a_xa'_{x+1} \square \ldots \square$ will receive its first trial with the instance $a_1a_2 \ldots a_xa'_{x+1} \ldots a'_l$, unless the schema has previously been introduced to the pool from another source.

Once again $f(x)$ of Figure 10 provides a simple illustration. New instances of a schema such as $1\square\square\ldots\square$ increase confidence that the observed average $\mu_{1\square\square\ldots\square}$ of evaluations of $f(x)$ for selected $x \in 1\square\square\ldots\square$ approaches $\mu_{1\square\square\ldots\square}$. At the same time an instance of some previously untried schema, say $110\square\ldots\square$, allows a plan of type $\mathfrak{R}$ to exploit the new schema (by giving it high rank) if it is above average.

In modifying the pool of schemata, crossing-over gains tremendous power from its intrinsic parallelism. *Each* crossing-over affects great numbers of schemata, as established by the following:

LEMMA 6.2.1: *Let* $A = a_1a_2\ldots a_l$ *and* $A' = a'_1a'_2\ldots a'_l$ *differ in attribute values at* x' *positions to the left of* $x + 1$ *and* x'' *positions to the right of* x. *Then either resultant of a single crossing-over of* A *with* A' *at* x *will be an instance of* $2^l - 2^{l-x'} - 2^{l-x''} + 2^{l-(x'+x'')}$ *"new" schemata (instanced by neither* A *nor* A'). *It will also be a new instance of* $2^{l-x'} + 2^{l-x''} - 2^{l-(x'+x'')}$ *schemata already instanced by* A *or* A' *(assuming* $x' \neq 0$ *and* $x'' \neq 0$).

Proof: After crossing-over, any schema which is defined at one or more of the x' positions on the left *and* at one or more of the x'' positions on the right will have neither A nor A' as an instance. On the left there are $2^{x'} - 1$ ways of combining one or more of the x' attribute values with "$\square$'s"; similarly there are $2^{x''} - 1$ ways on the right; at the other $l - (x' + x'')$ positions either an attribute value or a "$\square$" is allowable without restriction. Thus there are $(2^{x'} - 1)(2^{x''} - 1)(2^{l-(x'+x'')}) = 2^l - 2^{l-x'} - 2^{l-x''} + 2^{l-(x'+x'')}$ "new" schemata of which the resultant is an instance.

If $x' > 0$ and $x'' > 0$ the remainder of the 2^l schemata instanced by the resultant, i.e., $2^{l-x'} + 2^{l-x''} - 2^{l-(x'+x'')}$, will have a new instance (though they are not "new" schemata) since the resultant must differ by at least one attribute value from both A and A'. Q.E.D.

In other words *each* of the 2^l schemata instanced by the resultant arises from a potentially useful manipulation of schemata already in the pool (those instanced by A and A'). Note also that, even when l is only 20, a single operation is processing over a million schemata!

We can gain additional insight concerning crossing-over by considering its effect, over an extended interval, on the whole pool of schemata in $\mathfrak{B}(t)$. In the absence of reproduction and other operators, crossing-over generates a kind of diffusion from the pool to schemata not represented therein. More precisely,

repeated application of crossing-over to the individuals in $\mathcal{A}(t)$ yields a "steady state" wherein, at any instant (time-step), each schema ξ has a well-defined probability of occurrence $\lambda(\xi)$. It follows that the expected interval between occurrences of ξ will be just the reciprocal $1/\lambda(\xi)$ of this probability. Thus, if the proportions of schemata in $\mathcal{A}(t)$ are not far removed from steady-state values, $1/\lambda(\xi)$ is a reasonable measure of the expected time to an occurrence of ξ. Of course, no actively adapting system (natural or artificial) following a plan of type $\mathcal{A}$ will even begin to approach the steady state. Under such a plan, the steady state is continually "modulated" by changes in the number of instances of various ξ resulting from reproduction according to μ_ξ. In effect, with reproduction added, $1/\lambda(\xi)$ is a continually changing "background" testing rate, giving at any time a rough estimate of the expected time to first occurrence of ξ. These ideas, together with values for $\lambda(\xi)$ are established rigorously by

LEMMA 6.2.2: *Repeated crossing-over (with uniform random pairing of individuals and in the absence of other operators) in a population $\mathcal{A}(t)$ yields a "steady state" (i.e., a fixed point of the stochastic transformation) in which each schema ξ occurs with probability $\lambda(\xi) = \Pi_j P(_j\xi)$ where $P(_j\xi)$ is the overall proportion in $\mathcal{A}(t)$ of the allele occurring at the jth position of ξ (if a "□" occurs at the jth position take $P(_j\xi) = 1$).*

Proof: Let ${}^z\xi_1$ and ${}^z\xi_2$ be the resultants of a crossing-over of ξ_1 and ξ_2 at point x. Then a crossing-over of the resultants ${}^z\xi_1$ and ${}^z\xi_2$ at point x will bring back ξ_1 and ξ_2 (i.e., as may be determined directly from its definition, the crossover operator is self-dual).

Letting $P(\xi)$ designate the proportion of (instances of) ξ in $\mathcal{A}(t)$, we have $P(\xi_1)P(\xi_2)$ as the probability that ξ_1 will be paired with ξ_2 for crossing-over (under uniform random pairing). Thus the probability that ${}^z\xi_1$, ${}^z\xi_2$ arise from a crossing-over of ξ_1, ξ_2 at x is $P(\xi_1)P(\xi_2)P_x$, where P_x is the probability that crossover takes place at x.

Similarly the probability of a reversion (ξ_1, ξ_2 arising from ${}^z\xi_1$, ${}^z\xi_2$ by crossover at x) is $P({}^z\xi_1)P({}^z\xi_2)P_x$.

Considering only the effects of crossing-over at x on the pairs ξ_1, ξ_2 and ${}^z\xi_1$, ${}^z\xi_2$, there will be no changes in their probabilities of occurrence if

$$P(\xi_1)P(\xi_2)P_x = P({}^z\xi_1)P({}^z\xi_2)P_x.$$

If (and only if) such an equation holds for *every* x and *every* ordered quadruple (ξ_1, ξ_2, ${}^z\xi_1$, ${}^z\xi_2$) will there be no change in the probability of occurrence of any schema.

To balance all of these equations simultaneously, note first that the set of alleles $\{{}_j\xi_1, {}_j\xi_2\}$ is identical to the set of alleles $\{{}^z_j\xi_1, {}^z_j\xi_2\}$ since, after crossing-over, the same alleles are still present at the jth positions (though to the right of x they will have been interchanged). Hence

$$P({}_j\xi_1)P({}_j\xi_2) = P({}^z_j\xi_1)P({}^z_j\xi_2).$$

Thus, if $P(\xi) = \Pi_j P({}_j\xi)$ for each ξ, as defined in the statement of the lemma, we have for any x, ξ_1, ξ_2, ${}^z\xi_1$, ${}^z\xi_2$,

$$\begin{aligned} P(\xi_1)P(\xi_2) &= (\Pi_j P({}_j\xi_1))(\Pi_j P({}_j\xi_2)) = \Pi_j P({}_j\xi_1)P({}_j\xi_2) \\ &= \Pi_j P({}^z_j\xi_1)P({}^z_j\xi_2) = (\Pi_j P({}^z_j\xi_1))(\Pi_j P({}^z_j\xi_2)) \\ &= P({}^z\xi_1)P({}^z\xi_2). \end{aligned}$$

In other words, each of the equations will be balanced if the schemata occur with probabilities $\lambda(\xi) = \Pi_j P({}_j\xi)$; it is also clear that any departure from these probabilities will unbalance the equations in such a way as to result in changes in some of the probabilities of occurrence. Thus, the assignment $\lambda(\xi)$ is the unique "steady state" (fixed point) of the crossover operator. Q.E.D.

We can see from the proof of this lemma that a kind of "pressure" toward the steady state

$$\Delta = P(\xi_1)P(\xi_2) - P({}^z\xi_1)P({}^z\xi_2)$$

can be defined for each quadruple ξ_1, ξ_2, ${}^z\xi_1$, ${}^z\xi_2$. If $\Delta \neq 0$ for any quadruple then probabilities of occurrence will start changing and there will be a diffusion toward the resultants ${}^z\xi_1$, ${}^z\xi_2$ ($\Delta > 0$) or the precursors ξ_1, ξ_2 ($\Delta < 0$). For example, if $P(\xi_1) > \lambda(\xi_1)$ while the other components remain at their steady-state values, there will be a "movement to the right"—a tendency to increase the probabilities of the result. The following heuristic argument gives some idea of the rate of approach to steady state from such departures:

A given individual has probability $2/M$ of being involved in a crossover when $\mathfrak{A}(t)$ contains M individuals (since two individuals are involved in each application of the crossover operator). Thus in N trials a given individual can expect to undergo $2N/M$ crossing-overs. When N is in the vicinity of $lM/2$, where l is the length of individual representations, each individual in the population can be expected to have undergone independent crossing-over at almost every position. As a result even extreme departures from steady state should be much reduced in $lM/2$ trials.

The reduction to steady state does not, however, proceed uniformly with respect to all schemata because the crossover operator induces a *linkage* phenomenon. Simply stated, linkage arises because a schema is less likely to be affected by crossover if its defining positions are close together. In more detail, let ξ's defining positions (those not having a "□") be $i_1 < i_2 < \ldots < i_k$ and let the *length* of ξ be defined as $l(\xi) = (i_k - i_1)$. Then the probability of the crossover falling somewhere in ξ, once an instance of ξ has been selected for crossing-over, is just $l(\xi)/(l-1)$. E.g., if $A = a_1a_2a_3a_4a_5 \ldots a_l$ is selected for crossing-over, the probability of the crossover point x falling within $\xi = \square\, a_2\, \square\square\, a_5\, \square \ldots \square$ is $3/(l-1)$. Clearly the smaller the length of a schema, the less likely it is to be affected by crossing-over. Thus, the smaller the length of ξ, the more slowly will a departure from $\lambda(\xi)$ be reduced.

Alleles defining a schema ξ of small length $l(\xi)$ which exhibits above-average performance will be tried ever more frequently as a unit under an adaptive plan of type $\mathfrak{R}$. I.e., the alleles will be associated and tried accordingly. More modifications and tests of such schemata will be tried, and many of these trials will be of a variety of combinations with other similarly favored schemata defined at other positions. In effect such schemata serve as provisional structural elements or primitives. This observation is made precise by the following simple but important

THEOREM 6.2.3: *Consider a reproductive plan of type $\mathfrak{R}$ using only the* simple crossover operator*—defined as a crossover operator with both precursors, and the single crossover point, determined by uniform random selection. Then the expected proportion of* each *schema represented in $\mathfrak{B}(t)$ changes in one generation from $P(\xi, t)$ to*

$$P(\xi, t+1) \geq [1 - P_C \cdot (l(\xi)/(l-1))(1 - P(\xi, t))](\mu_\xi(t)/\mu(t))P(\xi, t),$$

where P_C is the proportion of individuals undergoing crossover during a generation and $\mu(t)$ is the observed average performance of $\mathfrak{B}(t)$. (The unit of time here—a generation—is the expected *time for an individual to produce its offspring.)*

Proof: During one generation each individual $A \in \mathfrak{B}(t)$ can be expected to produce $\mu_E(A)/\mu(t)$ offspring under a reproductive plan of type $\mathfrak{R}$. The total expected offspring of the set of instances $\mathfrak{B}_\xi(t)$ of ξ in $\mathfrak{B}(t)$ is thus given by

$$M'_\xi(t) = (\textstyle\sum_{A \in \mathfrak{B}_\xi(t)} \mu_E(A))/\mu(t) = \mu_\xi(t) M_\xi(t)/\mu(t).$$

If P_C is the proportion of $\mathfrak{B}(t)$ selected to undergo crossover and $l(\xi)$ is the length of ξ, then a proportion $P_C l(\xi)/(l-1)$ of the $M'_\xi(t)$ offspring will have a crossover falling within the defining positions of ξ. When an instance of ξ is crossed with

another instance of ξ the result will also be an instance of ξ; otherwise the resultant may not be an instance of ξ. Since the probability of ξ crossing with ξ is $P(\xi, t)$ no more than a proportion $(1 - P(\xi, t))P_C l(\xi)/(l - 1)$ of the modified offspring of ξ can be expected to be instances of schemata other than ξ; the remainder $[1 - (1 - P(\xi, t))P_C l(\xi)/(l - 1)]$ will be instances of ξ.

That is,

$$\begin{aligned} P(\xi, t + 1) &= M_\xi(t + 1)/M \\ &\geq [1 - (1 - P(\xi, t))P_C l(\xi)/(l - 1)]M'_\xi(t)/M \\ &= [1 - P_C \cdot (l(\xi)/(l - 1))(1 - P(\xi, t))](\mu_\xi(t)/\mu(t))P(\xi, t). \end{aligned}$$

(It should be noted that crossing-over applied to precursors which are not instances of ξ may yield a resultant which is an instance of ξ. Thus $M_\xi(t + 1)$ may be enlarged, by a small amount usually, from sources outside $B_\xi(t)$; this of course only strengthens the above bound.) Q.E.D.

From this result we see that the proportion of (instances of) a schema ξ will increase as long as

$$[1 - P_C \cdot (l(\xi)/(l - 1))(1 - P(\xi, t))](\mu_\xi(t)/\mu(t)) \geq 1$$

or, using the fact that $1/(1 - c) \geq 1 + c$ for $c \leq 1$,

$$\mu_\xi(t) \geq [1 + P_C \cdot (l(\xi)/(l - 1))(1 - P(\xi, t))]\mu(t).$$

Since the worst case occurs when $P_C = 1$ (every individual in $\mathcal{A}(t)$ subjected to crossing-over) and $P(\xi, t)$ is small, we see that ξ will always increase its representation if

$$\mu_\xi(t) \geq [1 + (l(\xi)/(l - 1))]\mu(t).$$

Since $1/l \leq l(\xi)/(l - 1) \leq 1$, short schemata need perform only slightly above average to increase, while the longest schemata (if they occur in small proportion) may have to exhibit a performance twice the population average to increase.

Theorem 6.2.3 provides the first evidence of the intrinsic parallelism of genetic plans. *Each* schema represented in the population $\mathcal{A}(t)$ increases or decreases according to the above formulation *independently of what is happening to other schemata* in the population. The proportion of each schema is essentially determined by its average performance in relation to the population average. Thus we see the evolution of a ranking of schemata based on observed performance, as suggested at the end of chapter 4 and amplified in section 5.4. Crossing-over serves

this adaptive process by continually introducing new schemata for trial, while testing extant schemata in new contexts—all this without much disturbing the ranking process (except for the longer schemata). Moreover, crossing-over makes it possible for the *schemata* represented in $\mathcal{B}(t)$ to move automatically to appropriate rankings through the application of the genetic plan to individual *structures* from $\mathcal{A}$. As a result this very large number of rankings is compactly stored in a selected, relatively small population of individuals (exploiting the possibility suggested at the end of chapter 4).

By extending the pressure analogy introduced just before Theorem 6.2.3 we can gain a global view of the interaction of reproduction and crossover. Whenever some schema ξ exhibits better-than-average performance, reproduction introduces "pressures" $\Delta > 0$, disturbing the steady state which would result from the action of the crossover operator alone. The disturbances both shift the steady-state values $\lambda(\xi')$ for large numbers of schemata, because of changes in the proportions $P(_j\xi)$ of the alleles $_j\xi$, $1 \leqq j \leqq l$, and also introduce local transitory departures because $P(\xi) > \lambda(\xi)$. Because all schemata are being affected simultaneously, and because reproduction affects them according to observed performance, we have a diffusion "outward" from schemata currently represented in $\mathcal{B}(t)$, a diffusion which proceeds rapidly in the vicinity of schemata exhibiting above-average performance. This is closely analogous to a gas diffusing from some central location through a medium of varying porosity, where above-average porosity is the analogue of above-average performance. The gas will exhibit a quickened rate of diffusion wherever it encounters a region of higher porosity, rapidly saturating the whole region. All the while it slowly but steadily infuses enclaves of low porosity. In effect, high porosity is exploited wherever it occurs, without prejudicing eventual penetration into regions of lower porosity. As a result the overall rate of penetration is much more determined by regions of high porosity and their proximity to each other than by average porosity.

Restated in terms of schemata, regions of higher porosity correspond to sets of schemata of above-average performance which can be produced from each other by relatively few crossovers. Thus, following the analogy, local optima in performance are thoroughly explored in an intrinsically parallel fashion. At the same time the genetic plan does not get entrapped by settling on some local optimum when further improvements are possible. Instead all observed regions of high performance are exploited without significantly slowing the overall search for better optima. Here we begin to see in a more precise context the powers of generalized genetic plans, powers first suggested in the specific context of section 1.4.

One final point: Plans of type $\mathcal{R}$ measure a schema's performance relative

to the current average performance of the population. Thus, as time elapses, schemata must meet progressively higher criteria to attain (or retain) a high ranking. (This is, again, somewhat analogous to the slowed rate of occupation of a gas as it occupies successively larger volumes, higher porosity being required for the same occupation rate.) As a result, older schemata associated with local optima steadily lose ranking as better optima are located (unless the older schemata are components of the new schemata), so that capacity is not wasted on superseded regions.

The overall results of this section can be illustrated by elaborating the comment (on page 99) about $f(x)$ of Figure 10. Using 6 bits of accuracy ($l = 6$), assume $A_1 = .001100$, $A_2 = .000100$, $A_3 = .101000$, $A_4 = .110011$, and $A_5 = .011100$ have been chosen at random to form $\mathcal{B}(0)$. (The size of $\mathcal{B}(0)$, $M = 5$, is of course much too small to be realistic even for an algorithm for artificial systems, but it is adequate to illustrate the effects of crossing-over.) Looking at Figure 10 we see that $\mu_1 = f(A_1) = f(.001100) \cong \frac{1}{2}$. Similarly, $\mu_2 = f(A_2) \cong 1\frac{1}{2}$, $\mu_3 \cong 2$, $\mu_4 \cong 1\frac{3}{4}$, and $\mu_5 \cong \frac{1}{2}$. For these points $\bar{\mu} \cong \frac{5}{4}$. Accordingly A_1 will produce $\mu_1/\bar{\mu} \cong (\frac{1}{2})/(\frac{5}{4}) = \frac{2}{5}$ offspring—i.e., A_1 has about 2 chances out of 5 of being reproduced. Similarly A_2 will have $\mu_2/\bar{\mu} \cong \frac{2}{5}$ offspring; and so on. Figure 12 shows a typical outcome for a plan of type $\mathcal{R}$ using only reproduction and simple crossover on $\mathcal{B}(0)$. (Thus, for the reproduction of A_1, a trial was made of a random variable yielding 1 with probability $\frac{2}{5}$ and 0 with probability $\frac{3}{5}$—the outcome of the trial was 0.) The crossing-

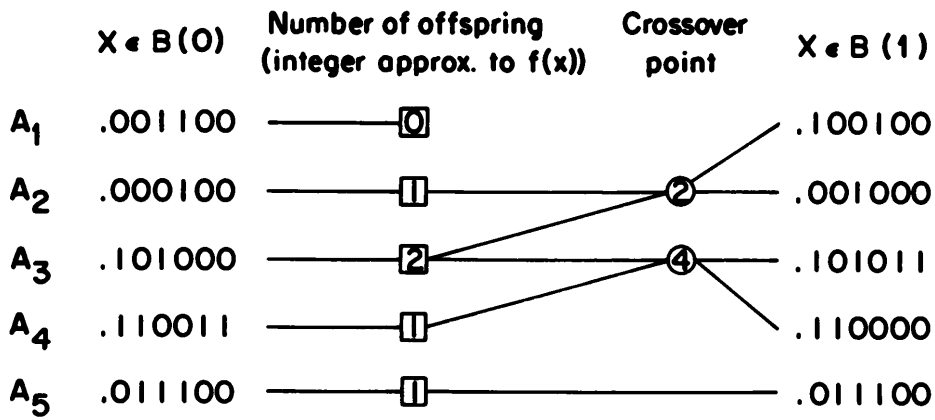


Fig. 12. Some effects of a type $\mathcal{R}$ plan on a one-dimensional function

over of A_2 with one of the replicates of A_3 at intersection ② serves both to generate another (different) instance of $1\square\square\ldots\square$, and to generate a first instance of $1\ \square\ 0\ \square\ldots\square$. Clearly such a crossover becomes increasingly likely as instances of $1\square\square\ldots\square$ and $\square\square 0\square\ldots\square$ proliferate. (Points from these schemata are likely to exhibit above-average values and hence will have more offspring on the average.) Similar effects will be happening to all other schemata instanced in $\mathcal{R}(0)$, $\mathcal{R}(1)$, etc. Figure 13 displays a more elaborate example of these effects.

3. GENERALIZED GENETIC OPERATORS—INVERSION

Crossover, by inducing a linkage between alleles, offers the possibility of an adaptable net of associations between alleles. By changing the length of a schema we modify the probability of its being affected by crossover; instances of a shorter schema are less likely to have the defining alleles separated by crossover. In consequence, under a plan of type $\mathcal{R}$, instances of the shorter schema proliferate more rapidly. The long-term effect is a selective increase in the linkage of various schemata exhibiting above-average performance. The corresponding alleles (attributes) are more frequently found in association (on the same string) in successive generations. Since schemata are defined for any string-representable domain $\mathcal{A}$, such an adaptable network of associations can be induced for any such domain by introducing an appropriate operator for changing linkage.

The linkage between the alleles defining a schema can be altered only by changing the length of the schema. That is, the positions of the alleles defining the schema (particularly the end-points) must be modifiable. However, up to this point, the functional meaning of an allele has been determined by its position. The allele a_i at the ith position of the representation of the structure A is the value $\delta_i(A)$ of the ith detector when A is its argument. Thus, if linkage is to be changed, an allele must have the same functional interpretation in any position (as is the case generally in genetics). This in turn requires a change in the method of representation.

The simplest way to change the method of representation formally is to assign each allele an index indicating the detector with which it is associated. That is, each allele is now taken to be a pair (i, a) indicating that $a = \delta_i(A)$. It follows that a structure A can be represented by any permutation of

$$((1, \delta_1(A)), (2, \delta_2(A)), \ldots, (l, \delta_l(A))).$$

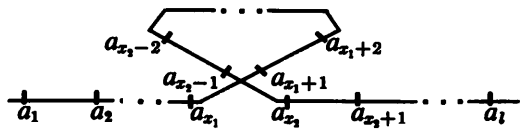
For example,

$$((3, \delta_3(A)), (2, \delta_2(A)), (1, \delta_1(A)), (4, \delta_4(A)), \ldots, (l, \delta_l(a)))$$

would still represent A. Moreover, the schema $(1, \delta_1(A))\,\square\square\,(4, \delta_4(A))\,\square \ldots \square$ designates the same subset of $\mathfrak{A}$ as the schema $\square\square(1, \delta_1(A))(4, \delta_4(A))\,\square \ldots \square$, though the latter is more tightly linked than the former. To define this enlarged set of representations precisely, let V_i be redefined to be the set of pairs $V_i' = \{(i, v),$ for all $v \in V_i\}$ and let $\sigma^\dagger$ indicate the *set* of all permutations of the string (or l-tuple) σ. Then $\mathfrak{A}^\dagger = (\Pi_{i=1}^{l} V_i')^\dagger$ is the enlarged set of all representations of elements in $\mathfrak{A}$. The set of schemata is enlarged accordingly to $\Xi = (\Pi_{i=1}^{l} \{V_i' \cup \{\square\}\})^\dagger$.

The object now is to find an operator which when used with crossover and reproduction will tend to replace an above-average schema ξ with a permutation $\xi' \in \xi^\dagger$ of shorter length $l(\xi') < l(\xi)$. The genetic operator which fits this specification is *inversion*. It works by producing a crossover within a single structure as follows:

1. A structure $A = a_1 a_2 \ldots a_l$ is selected (usually at random) from the current population $\mathfrak{B}(t)$ (where each a_i, $i = 1, \ldots, l$, now represents a pair $(j, v) \in V_j$).
2. Two numbers, x_1' and x_2', are selected from $\{0, 1, 2, \ldots, l + 1\}$ (again at random) and are used to define $x_1 = \min\{x_1', x_2'\}$ and $x_2 = \max\{x_1', x_2'\}$.
3. A new structure is formed from A by inverting the segment which lies to the right of position x_1 and to the left of position x_2, yielding $a_1 \ldots a_{x_1} a_{x_2-1} a_{x_2-2} \ldots a_{x_1+1} a_{x_2} \ldots a_l$.



It is clear that a single inversion can bring previously widely separated alleles into close proximity, viz., a_{x_1} and a_{x_2-1} in the description. It is also clear that any possible permutation of the representation can be produced by an appropriate sequence of inversions. (More technically, the inversions $(x_1 = 0, x_2 = 2)$, $(x_1 = 1, x_2 = 3)$, $\ldots$, $(x_1 = l - 1, x_2 = l + 1)$ are sufficient to generate the group of all permutations of order l.) The effect of the inversion operator upon (the instances of) a schema ξ is to randomly produce permutations ξ' of ξ with varying lengths. Though inversion alters the linkage of schemata, it does *not* alter the subsets of $\mathfrak{A}$ which they designate. Every permutation ξ' of ξ designates the same subset in the set of (original) structures $\mathfrak{A}$ (since the same set of detector values occurs in both ξ and ξ'). The lengths of many schemata are affected simultaneously by a single inversion, so this operator too exhibits intrinsic parallelism. As with crossover, schemata of shorter lengths are less frequently affected by the inversion operator.

Let us define the *simple inversion operator* as an inversion with both the structure selected for inversion and the two points x_1 and x_2 determined by uniform random selection. To see the combined effect of simple inversion, simple crossover, and reproduction we need only refer to Theorem 6.2.3. The theorem guarantees that, if inversion has produced a permutation ξ' of ξ where $l(\xi') < l(\xi)$, then the proportion of ξ' in $\mathcal{A}(t)$ increases more rapidly than the proportion of ξ. For example, if $P_C = 1$ and $P(\xi, t), P(\xi', t) \ll 1$ we can expect

$$P(\xi', t+1) = (((l - 1 - l(\xi'))/(l - 1 - l(\xi)))(P(\xi', t)/P(\xi, t))P(\xi, t+1)$$

since $\mu_\xi = \mu_{\xi'}$. Or, after T generations

$$P(\xi', t+T) = (((l - 1 - l(\xi'))/(l - 1 - l(\xi)))^T(P(\xi', t)/P(\xi, t))P(\xi, t+T).$$

As a result, any time inversion yields a shorter permutation ξ' of a schema ξ of above-average performance, that permutation will rapidly predominate. Because the rate of reproduction of a schema is dependent upon its length, there is a constant "pressure" toward tighter linkage of the defining alleles of schemata. Because only schemata exhibiting above-average performance occupy substantial fractions of $\mathcal{A}(t)$, the "pressure" is only important for such schemata. Inversion, by repeatedly varying the linkage, gives this pressure a chance to act.

A great many schemata are affected by each inversion, but tightly linked schemata are much less likely to be affected than loosely linked ones, so that variations are primarily in the loosely linked schemata. That is, changes in linkage are concentrated in the loosely linked (long) schemata of above-average performance, where changes are desirable. More precisely, if P_I is the proportion of the population undergoing inversion in a given generation, then the probability of a schema ξ of length $l(\xi)$ being affected is

$$2P_I \cdot (l(\xi)/(l - 1)) \cdot (1 - l(\xi)/(l - 1)) = 2P_I[l(\xi)/(l - 1) - (l(\xi)/(l - 1))^2],$$

where the second factor comes from the fact that an inversion wholly inside a schema does not affect its length. Hence, if $l(\xi) = b \cdot l(\xi') < l/4$, $b > 1$, for two schemata ξ and ξ', ξ is almost b times as likely to have its length altered.

One new restriction must be made upon the crossover operator when it is used in combination with inversion. Because of inversion, two l-tuples in $\mathcal{A}(t)$ will not always have the alleles for a given detector at the same position. Crossing-over can thus produce resultants with two (or more) alleles for a given detector, or resultants with no alleles for a given detector. For example, crossing

$$((1, a_1), (2, a_2), (3, a_3)) \quad \text{with} \quad ((1, a_1'), (3, a_3'), (2, a_2'))$$

at $x = 2$ yields $((1, a_1), (2, a_2), (2, a_2'))$ as one of the resultants. The simplest way to remedy this is to permit crossing-over only between *homologous* representations, where two representations are defined to be homologous if the detector indices (first number of each pair in the representation) are in the same order. For example, $((1, a_1), (3, a_3), (2, a_2))$ is homologous to $((1, a_1'), (3, a_3'), (2, a_2'))$, even if $a_j \neq a_j'$ for some or all j, while $((1, a_1), (2, a_2), (3, a_3))$ is not homologous to either of the foregoing. This remedy requires that the probability of inversion P_I be small so that there will exist substantial homologous subpopulations for the crossover operator to act upon. A second alternative (with a biological precedent) would be to temporarily make the second of the l-tuples chosen for crossover homologous to the first by reordering it, returning it to the population in its original order after the resultants of the crossing-over are formed. Under this alternative inversion can be unrestricted, i.e., P_I can be as large as desired.

Summing up: Inversion, in combination with reproduction and crossover, selectively increases the linkage (decreases the length) of schemata exhibiting above-average performance, and it does this in an intrinsically parallel fashion.

4. GENERALIZED GENETIC OPERATORS—MUTATION

Though mutation is one of the most familiar of the genetic operators, its role in adaptation is frequently misinterpreted. In genetics mutation is a process wherein one allele of a gene is randomly replaced by (or modified to) another to yield a new structure. Generally there is a small probability of mutation at each gene in the structure. In the formal framework this means that, each structure $A = a_1 a_2 \ldots a_l$ in the population $\mathfrak{B}(t)$, is operated upon as follows:

1. The positions $x_1, x_2, \ldots, x_h$ to undergo mutation are determined (by a random process where each position has a small probability of undergoing mutation, independently of what happens at other positions).
2. A new structure $A' = a_1 \ldots a_{x_1-1} a_{x_1}' a_{x_1+1} \ldots a_{x_2-1} a_{x_2}' a_{x_2+1} \ldots a_h$ is formed where a_{x_i}' is drawn at random from the range V_i of the detector δ_i corresponding to position x_1, each element in V_i being an equilikely candidate; $a_{x_2}', \ldots, a_{x_h}'$ are determined in the same way.

If ${}^1P_M$ is the probability of mutation at each position, then the probability of h mutations in a single representation is given by the Poisson distribution with parameter ${}^1P_M$.

If successive populations are produced by mutation alone (without reproduction), the result is a random sequence of structures drawn from $\mathfrak{A}$. The process is evidently enumerative (see section 1.5) since the order in which structures are generated is unaffected by the observed performances of the structures. Even a reproductive plan of type $\mathfrak{R}$ using only the mutation operator is little more than an enumerative plan retaining the best structure encountered to each point in time. That is, if ${}^1P_M$ is small enough, reproduction will assure that structures with above-average performance predominate in successive generations thus retaining the better structures generated by the mutation operator. There is actually a bit of history dependence since, with ${}^1P_M$ small, the most likely structures resulting from mutation will differ by one or two alleles from the current "best" structures. Thus, the sequence of tests is not entirely random, though the dependence on observations is very unsophisticated compared to that generated by crossing-over.

Since enumerative plans are, at best, useful in very limited situations, it would seem that mutation's primary role is *not* one of generating new structures for trial—a role very efficiently filled by crossing-over. It might be objected that crossing-over cannot generate all possible combinations of alleles unless the population $\mathfrak{B}(t)$ contains at least one copy of every allele. However this is not a burdensome requirement. If k is the maximum number of alleles for any detector, then as few as k strings will suffice to provide a copy of each allele. (E.g., if $V_i = \{0, 1\}$, $i = 1, \ldots, l$, then the two l-tuples $00\ldots0$ and $11\ldots1$ suffice.) There is nevertheless a difficulty which is remedied by mutation. In a population that is small relative to $\mathfrak{A}$, there is always the possibility that the last copy of some

will be eliminated during the deletion phase of a plan of type $\mathfrak{R}$. Alleles which occur in structures of below-average performance are particularly susceptible; yet at some later stage these same alleles may be required in a combination necessary for further improvement. Stated another way, the lost allele may be necessary for the adaptive plan to escape a false peak. Once an allele is lost from a population, the crossover operator has no way of reintroducing it. Here, then, is a role uniquely filled by mutation, because it assures that no allele permanently disappears from the population.

Mutation introduces an additional source of loss for schemata undergoing reproduction. If the probability of mutation at each position is less than or equal to ${}^1P_M$, then a schema ξ defined on $l^0(\xi)$ positions can expect to undergo one or more mutations with probability

$$1 - (1 - {}^1P_M)^{l^0(\xi)}$$

which is approximately equal to $l^0(\xi) \cdot {}^1P_M$ when ${}^1P_M$ is small relative to $1/l$. Thus, adding mutation to the list of operators in Theorem 6.2.3, we get

COROLLARY 6.4.1: *Under a reproductive plan of type* $\mathfrak{R}$ *using the simple crossover operator and mutation, the expected proportion of each schema represented in* $\mathfrak{R}(t)$ *changes in one generation from* $P(\xi, t)$ *to*

$$P(\xi, t+1) \geqq [1 - P_C \cdot (l(\xi)/(l-1))(1 - P(\xi, t))] \cdot (1 - {}^1P_M)^{l^0(\xi)} \left(\frac{\mu_\xi(t)}{\mu(t)}\right) P(\xi, t).$$

Unlike the case for crossing-over, mutation is a constant source of loss for a schema ξ, *with* ${}^1P_M$ *fixed, even when* $P(\xi, t) = 1$. *In effect it is a "disturbance" introduced to prevent entrapment on a false peak.*

Summing up: Mutation is a "background" operator, assuring that the crossover operator has a full range of alleles so that the adaptive plan is not trapped on local optima. (Of course if there are many possible alleles—e.g., if we consider a great many variants of the nucleotide sequences defining a given gene—then even a large population will not contain all variants. Then mutation serves an enumerative function, producing alleles not previously tried.)

5. FURTHER INCREASES IN POWER

The next chapter will establish that the three genetic operators just described are adequate for a robust and general purpose set of adaptive plans, with one important reservation which will be discussed at the end of this section. However, there are additional operators which can make significant contributions to efficiency in more complex situations. Chief among these is the dominance-change operator which (among other things) helps to control losses resulting from mutation. Because losses resulting from mutation, for given ${}^1P_M$, do not diminish as schema ξ gains high rank, a constant "load" is placed on the adaptive plan by the random movements away from optimal configurations. For this reason it is desirable to keep the mutation rate ${}^1P_M$ as low as possible consistent with mutation's role of supplying missing alleles. In particular, if the rate of disappearance of alleles can be lowered without affecting the efficiency of the adaptive plan, then the mutation rate can be proportionally lower. Since the main cause of disappearance of alleles is sustained below-average performance, the rate of loss can be reduced by shielding such

alleles from continued testing against the environment. Dominance provides just such shielding.

To introduce dominance, we must extend the method of representation once again. Pairs of alleles will be used for each detector, so that a representation involves a pair of homologous l-tuples. The object is to let some of the extra alleles be carried along with the others in an unexpressed form, forming a kind of reservoir of protected alleles. Precisely, then, the set of representations will be extended to the set of all permutations of homologous pairs $\mathfrak{A}_2^\dagger = (\Pi_{i=1}^{l}(V_i')^2)^\dagger$. Since there is now a *pair* of alleles at each position there is no longer a direct correspondence between the detector values for a structure A and the representation of A.

Let (A', A'') be a homologous pair of l-tuples drawn from $\mathfrak{A}_2^\dagger$ and let ${}_i(A', A'') = {}^{df}((h, v'), (h, v''))$ where $v', v'' \in V_h$, designate the pair of alleles occurring at the ith position of the l-tuples. The most direct way to relate this pair of l-tuples to a structure is to designate either v' or else v'' as the value of detector h, ignoring the other allele. The allele so designated will be called *dominant*, the other *recessive*. For each position i, this designation should be completely determined by information available in the pair (A', A''). Formally, for each i there should be a dominance map $d_i: \mathfrak{A}_2^\dagger \to \mathfrak{A}$ such that, for each $(A', A'') \in \mathfrak{A}_2^\dagger$, $d_i({}_i(A', A''))$ is either the first allele or the second allele of ${}_i(A', A'')$. It should be emphasized that in this general form, the determination of the dominant allele in ${}_i(A', A'')$ may depend upon the whole context (i.e., the other alleles in (A', A'')). (This corresponds closely with Fisher's [1937, Chapter III] theory of dominance.) A simpler approach makes the determination dependent only upon the pair ${}_i(A', A'')$ itself. Thus, for each h,

$$d_h': V_h^2 \to V_h \text{ such that } d_h'(v', v'') \in \{v', v''\}$$

and

$$d_i: \mathfrak{A}_2^\dagger \to \mathfrak{A} \text{ such that for } {}_i(A', A'') = ((h, v'), (h, v'')),\ d_i({}_i(A', A'')) = d_h'(v', v'').$$

Accordingly (A', A'') represents the structure $A \in \mathfrak{A}$ for which

$$\delta_h(A) = d_{i(h)}(A', A'')$$

where $i(h)$ is the index of the pair of alleles in (A', A'') for detector h.

A particularly interesting example of the simpler dominance map, useful for binary (two allele) codings (see chapter 4), can be constructed as follows. Let $V_h = \{1, 1_0, 0\}$, where 1_0 is to be recessive whenever it is paired with 0, and let the mapping $d_h: V_h^2 \to V_h$ be given by the following table:

v', v''	$d_h(v', v'')$
0 0	0
0 1_0	0
0 1	1
1_0 0	0
1_0 1_0	1
1_0 1	1
1 0	1
1 1_0	1
1 1	1

Then, for example, the representation

$$A' = ((1, 0), (3, 1_0), (2, 0), (4, 1))$$
$$A'' = ((1, 1_0), (3, 0), (2, 1), (4, 0))$$

maps to the (unpermuted) representation

$$((1, 0), (2, 1), (3, 0), (4, 1)) = 0101.$$

That is, (A', A'') represents the structure $A \in \mathfrak{A}$ for which

$$\delta_1(A) = 0, \delta_2(A) = 1, \delta_3(A) = 0, \delta_4(A) = 1.$$

In order to examine the effect of dominance on genetic plans, the simple crossover operator must be extended to this new type of representation (inversion takes place, as before, on the individual l-tuples in the homologous pairs). To cross the homologous pair (A', A'') with the pair (A''', A''''), the procedure will be to cross A' with A''' with probability P_C, and then select one of the resultants at random. Similarly, A'' is crossed with A'''' and, again, one of the resultants is selected at random. The two selected resultants are then paired to yield one of the outcomes of the extended operation; the other two resultants are paired to yield the other outcome (if it is to be used).

To see the effect of dominance on the mutation rate, let us consider the case of two alleles v_1, v_0 at position i, where v_1 is dominant and v_0 is recessive. There are four distinct pairs of these alleles which can occur at position i in (A', A''), namely $\{(v_1, v_1), (v_1, v_0), (v_0, v_1), (v_0, v_0)\}$, and only one of these maps to v_0 under the dominance map. That is, in the pairs (v_1, v_0) and (v_0, v_1) the allele v_0 is shielded or stored without test (because the representation maps to one where only allele v_1 is present).

Stated another way, allele v_0 is only expressed or tested when it occurs in the pair (v_0, v_0). Let us assume that, on the average, the adaptive plan is to provide at least one occurrence of each allele in every T generations. That is, $P(v_0, t) \geqq 1/MT$ must be assured. In the absence of dominance (using the earlier single l-tuple representation), let the reproduction rate of v_0 (corrected for operator losses) be $(1 - \epsilon(v_0))$ exclusive of additions resulting from mutation. Then

$$P(v_0, t + 1) = (1 - \epsilon(v_0))P(V_0, t) + {}^1P_M(1 - P(v_0, t)) - {}^1P_M P(v_0, t).$$

To keep $P(v_0, t) \geqq 1/MT$ for all t, ${}^1P_M$ must be at least large enough to maintain the steady state $P(v_0, t) = P(v_0, t + 1) = 1/MT$. That is,

$$1/MT = (1 - \epsilon(v_0))/MT + {}^1P_M(1 - 2/MT)$$

or

$${}^1P_M = \epsilon(v_0)/((1 - 2/MT) \cdot MT).$$

If MT is at all large (as it will be for all cases of interest) this reduces to

$${}^1P_M \geqq \epsilon(v_0)/MT$$

as a close approximation to the mutation rate required without dominance. (In the extreme case that alleles v_0 are deleted whenever they are tested, ${}^1P_M = 1/MT$.) With dominance, the allele v_0 is subject to selection only when the pair (v_0, v_0) occurs. Under crossover, as extended to homologous pairs, the pair (v_0, v_0) occurs with probability $P^2(v_0, t)$. The loss from selection then is

$$2\epsilon(v_0)P^2(v_0, t)M$$

the factor 2 occurring because 2 copies of v_0 are lost each time the pair (v_0, v_0) is deleted. Again the gains from mutation are

$${}^1P_M \cdot (1 - P(v_0, t)2M - {}^1P_M P(v_0, t))2M$$

where the factor 2 occurs because the M homologous *pairs* are $2M$ l-tuples. Thus

$$P(v_0, t + 1) = P(v_0, t) - 2\epsilon(v_0)P^2(v_0, t) + {}^1P_M \cdot 2(1 - 2P(v_0, t))$$

for the homologous pairs with dominance. Setting $P(v_0, t) = P(v_0, t + 1) = 1/MT$ as before, and solving, we get

$${}^1P_M = \epsilon(v_0)/((1 - 2/MT)(MT)^2).$$

We have thus established

LEMMA 6.5.1: *To assure that, at all times, each allele a occurs with probability* $P(a, t) \geqq 1/MT$, *the mutation rate* ${}^1P_M$ *must be* $\geqq 1/MT$ *in the absence of dominance, but only* $\geqq (1/MT)^2$ *with dominance.*

For example, to sustain an average density of at least 10^{-3} for every allele, the mutation rate would have to be 10^{-3} without dominance, but only 10^{-6} with dominance.

It should be noted that, with dominance, $P(v_0, t)$ is no longer the expected *testing* rate. Although dominance allows the constant mutation load to be reduced, while maintaining a given proportion of disfavored alleles as a reserve, the testing rate of the reserved alleles is only $P^2(v_0, t)$ not $P(v_0, t)$. This reservoir is only released through a change in dominance.

Dominance change in the general case $d_i: \mathcal{A}_2^l \to \mathcal{A}$, $i = 1, \ldots, l$, occurs simply through a change in context, so that dominance is directly subject to adaptation by selection of appropriate contexts. In the more restricted case $d_h: V_h^2 \to V_h$ a special operator is required. The example using $V_h = \{1, 1_0, 0\}$ will serve to illustrate the process. The basic idea will be to replace some or all occurrences of 1 by 1_0, and vice versa, in an l-tuple. Thus the previous recessives become dominant and vice versa, this change being transmitted to all progeny of the l-tuple. A simple way to do this is to designate a special inversion operator which not only inverts a segment but carries out the replacement in the inverted segment. (In genetics, there is a distant analogue in the effects produced by changes of context when a region is inverted, but it should not be taken literally.) Thus for the dominance-change inversion operator, step 3, p.107 of the inversion operator is followed by

4. In the inverted segment each occurrence of 1 is replaced by 1_0 and vice versa.

With this operator the defining alleles of an arbitrary schema can be "put in reserve" in a single operation to be "released" later, again in a single operation.

Dominance provides a reserved status not only for alleles but, more importantly, for schemata. A useful schema ξ_1 defined on many positions may be the result of an extensive search. As such it represents a considerable fragment of the adaptive plan's history, embodying important adaptations. When it is superseded by a schema ξ_2 exhibiting better performance, it is important that ξ_1 not be discarded until it is established that ξ_2 is useful over the same range of contexts as ξ_1. ξ_2's performance advantage may be temporary or restricted in some way, or ξ_1 may be useful again in some context engendered by ξ_2. In any case it is useful to retain

ξ_1 for a period comparable to the time it took to establish it. Dominance makes this possible.

Summing up: Under dominance, a given minimal rate of occurrence of alleles can be maintained with a mutation rate which is the square of the rate required in the absence of dominance. Moreover, with the dominance-change operator the combination of alleles defining a schema can be "reserved" (as recessives) or "released" (as dominants) in a single operation.

When the performance function depends upon many more or less independent factors, there is another pair of operators, *segregation* and *translocation* which can make a significant contribution to efficiency. In such situations it is useful to make provision for distinct and independent sets of associations (linkages) between genes. This again calls for an extension in the method of representation. Let each element in $\mathfrak{A}$ be represented by a *set* of homologous pairs of n-tuples, and let crossover be restricted to homologous n-tuples. After two elements of $\mathfrak{A}$, A and A', are chosen for crossover and after *all* homologous pairs have been crossed (as detailed under the discussion of dominance change) then from *each* pair of resultants one is chosen at random to yield the offspring's n-tuples. Each offspring thereby consists of the same number of homologous pairs of n-tuples as its progenitors. The genetic counterpart of this random selection of resultants is known as segregation. Clearly, under segregation, there is no linkage between alleles on separate nonhomologous n-tuples, while alleles on homologous n-tuples are linked as before. With this representation it is natural to provide an operator which will shift genes from one linkage set to another (so that, for example, schemata that are useful in one context of associations can be tested in another). The easiest way to accomplish this is to introduce an exceptional crossover operator, the *translocation* operator, which produces crossing-over between randomly chosen nonhomologous pairs.

Another genetic operation provides a means of adaptively modifying the effective mutation rate for different closely linked sets of alleles. The operator involved is *intrachromosomal duplication* (see Britten 1968); it acts by providing multiple copies of alleles on the same n-tuple. To interpret this operation, n-tuples with multiple copies of the alleles for a given gene must be mapped into the set of original structures. This can be done most directly by extending the concept of dominance to multiple copies of alleles. With this provision, if there are k_a copies of a given allele a, the probability of one or more mutations of allele a is k_a times greater than if there were but one copy. That is, the probability of occurrence, via mutation, of allele $a' \neq a$ is increased k_a times. Thus, increases and decreases in the number of copies of an allele have the effect of modifying the (local) mutation

rate. In genetics the decreases are provided by *deletion*. The easiest generalization of these operators is an operator which doubles (or halves) the number of copies at a randomly chosen (set of adjacent) location(s).

Though the operators just described are useful, they are not necessary. Moreover they do *not* compensate the major shortcoming of genetic plans which use just the first three operators described. That shortcoming is the complete dependence of such plans upon the detectors determining the representation. If the set of detectors $\{\delta_i\}$ is inadequate, in any way, the plan must operate within that constraint. However, if the plan could add or modify detectors at need, it could circumvent the difficulty. This implies making the detectors themselves subject to adaptation. When we note that each detector can be specified by an appropriate subroutine (string of instructions) for a general purpose computer, a way of making this extension suggests itself. By keeping the number of basic instructions from which the subroutines are constructed small, we can treat them as alleles. $\mathfrak{A}$ can then be extended to include all strings of basic instructions. In this way $\mathfrak{A}$ contains a representation of any possible detector, set of detectors or, in fact, any effectively describable way of processing information. Moreover, under this extension, favored schemata correspond to useful coordinated sets of instructions (such as detectors). Genetic plans applied to $\mathfrak{A}$, so extended, can thus develop whatever functions or representations they need. This problem and the suggested approach are complex enough to merit a chapter, chapter 8.

The Jacob-Monod (1961) "operon" model of the functioning of the chromosome has an interesting relation to the extension of $\mathfrak{A}$ just suggested. In the extension, we can think of each element of $\mathfrak{A}$ as a program processing inputs from the environment to produce outputs affecting that environment (cf. chapter 3.4 where transformations $\{\eta_i\}$ are the outputs). The performance of the element is thus directly determined by the relevance or fitness of the program. The "operon" model treats the chromosome as a similar information processing device. Each gene can either be active (cf. the execution of an instruction) or inactive. When active the gene is participating in the production of signals (enzymes) which modulate the ongoing activity of the cell. It thereby determines the cell's modes of action and critical aspects of its structure. The genes are collected in groups—operons—such that all genes in the group are either simultaneously active or inactive, as determined by one control gene in the group called an "operator gene" (or more recently, a "receptor gene" in Britten and Davidson 1969; see Fig. 14). The remainder of the cell is treated as the chromosome's environment. The action of the "receptor gene" is conditional upon the presence of signals (proteins) from the cell (usually through the mediation of other genes—"repressor" or "sensor"

genes). In this way one operon can cause the cell to produce signals which (with controlled delays) turn on other operons. This provision for action conditional upon previous (conditional) actions gives the chromosome tremendous information-processing power. In fact, as will be shown in chapter 8, any effectively describable information-processing program can be produced in this way.

6. INTERPRETATIONS

For the geneticist, the picture of the process of adaptation which emerges from the mathematical treatment thus far exhibits certain familiar landmarks:

> Natural selection directs evolution not by accepting or rejecting mutations as they occur, but by sorting new adaptive combinations out of a gene pool of variability which has been built up through the combined action of mutation, gene recombination, and selection over many generations. For the most part Darwin's concept of *descent with modification* fits in with our modern concept of interaction between evolutionary processes, because each new adaptive combination is a modification of an adaptation to a previous environment. (p. 31)
>
> *Inversions* and *translocations* of chromosomal segments, when present in the heterozygous condition, can increase genetic linkage and so bind together adaptive gene combinations. . . . The importance of such increased linkage is due to the number of diverse genes which must contribute to any adaptive mechanism in a higher plant or animal. (p. 57)
>
> Stebbins in *Processes of Organic Evolution*

> Not only do we claim in this case [of inversions found in *D. pseudoobscura* and *D. persimilis*] that the precise pairing of the chromosomes in the species hybrids shows that the chromosomal material has had a common source, but we also claim that the sequence of rearrangements [produced by inversions] that occurred in the chromosome reconstructs for us the precise pattern of change that led up to and then beyond the point of speciation.
>
> Wallace in *Chromosomes, Giant Molecules, and Evolution* (p. 49)

At the same time the emphasis on gene interaction poses a series of difficult problems:

> Intricate adaptations, involving a great complexity of genetic substitutions to render them efficient would only be established, or even maintained in the species, by the agency of selective forces, the intensity of which may be thought of broadly, as proportional to their complexity.
>
> Fisher in *Evolution as a Process*, ed. Huxley et al. (p. 117)

> The interaction of genes is more and more recognized as one of the great evolutionary factors. The longer a genotype is maintained in evolution, the stronger will its developmental homeostasis, its canalizations, its system of internal feed backs become. . . . one of the real puzzles of evolution is how to break up such a perfectly co-adapted system in such a way so as not to induce extinction . . .
>
> Mayr in *Mathematical Challenges to the Neo-Darwinian Interpretation of Evolution*, ed. Moorhead & Kaplan (p. 53)

> The other and I think more interesting problem, which we have hardly begun to solve, is the question: How many changes of information are necessary to explain evolution?
>
> Waddington in *Mathematical Challenges to the Neo-Darwinian Interpretation of Evolution*, ed. Moorhead & Kaplan (p. 96)

And, even though the centennial for the *Origin of Species* has passed, speciation still lacks a general mathematical explanation. Moreover, the question of "enough time" plagues the neo-Darwinian almost as much as it did his predecessors. It is a question which weighs heavily if it is assumed that coadapted sets of alleles occur only by the spread of mutant alleles to the point that relevant combinations are likely (see Eden's [1967] comments).

In the present context each of these questions can be rephrased in terms of the processing of schemata by genetic operators. This allows us to probe the origin and development of coadapted sets of alleles much more deeply, particularly the way in which different genetic mechanisms enable exploitation of useful epistatic effects. In the next chapter, we will be able to extend Corollary 6.4.1 to demonstrate the simultaneous rapid spread of *sets* of alleles, as sets, whenever they are associated with above-average performance (because of epistasis or otherwise). Theorem 7.4 establishes the efficiency of this process for epistatic interactions of arbitrary complexity (i.e., for any fitness function $\mu: \mathfrak{A} \to \mathfrak{U}$, however complex). Section 7.4 gives a specific example of the process in genetic terms and exhibits a version of Fisher's (1930) theorem applicable to arbitrary coadapted sets. Finally, in section 9.3, the formalism is extended to give an approach to speciation. This extension suggests reasons for competitive exclusion within a niche, coupled with a proliferation of (hierarchically organized) species when there are many niches.

For the nongeneticist, the illustration at the end of section 6.2 should convey some of the flavor of algorithms of type $\mathfrak{R}$ as optimization procedures. It is easy enough to extend that illustration to cover inversion and mutation. For example, under the revised representation of section 6.3 each bit δ is paired with a number j designating its significance (i.e. (j, δ) designates the bit $\delta \cdot 2^{-j}$). Thus bits

of different orders can be set adjacent to each other in a string without changing their significance. In consequence, under the combined effect of inversion and reproduction, bits defining various regions of above-average values for $f(x)$ will be ever more tightly linked. This in turn increases the rate of exploration of intersections and refinements of these regions. Filling in the remaining details to complete the extension of the illustration is a straightforward exercise. Section 7.3 in the next chapter provides a detailed example of the response of an algorithm of type $\mathfrak{R}$ to nonlinearities. Theorem 7.4 of that chapter, coupled with the comments on dimensionality in chapter 4 (p. 71) shows that, whatever the form of f (i.e., for any f mapping a bounded d dimensional space into the reals), an algorithm of type $\mathfrak{R}$ optimizes expeditiously. Moreover, the algorithm does this while rapidly increasing the average value of the points it tests (though they may be scattered through many different hyperplanes), thus making the algorithm useful for "on-line" control. Sections 9.1 and 9.3 provide more detailed summaries of these advantages.