Complex Systems **5** (1991) 183–205

Equivalence Class Analysis of Genetic Algorithms

Nicholas J. Radcliffe*
Department of Physics, University of Edinburgh, Kings Buildings, Edinburgh, EH9 3JZ, Scotland

Abstract. The conventional understanding of genetic algorithms depends upon analysis by schemata and the notion of intrinsic parallelism. For this reason, only k-ary string representations have had any formal basis, and nonstandard representations and operators have been regarded largely as heuristics rather than as principled algorithms. This paper extends the analysis to general representations through identification of schemata as equivalence classes induced by implicit equivalence relations over the space of chromosomes.

1. Introduction

Intrinsic parallelism[1]—the phenomenon whereby each n-gene chromosome is an instance of 2^n schemata—has been the key theoretical tool for analyzing and understanding genetic algorithms. As conventionally understood, it provides powerful arguments for using binary genes to maximize the degree of intrinsic parallelism available.

Not all problems, however, find natural expression as binary—or indeed, k-ary—strings. Examples in this class include the much-studied Traveling Sales-rep Problem (TSP) [12, 13, 23], neural network shaping and training [14, 16, 3], and graph optimization [17, 18]. Of these, only the TSP has generated an alternative to standard schema analysis, in the form of Goldberg's o-schemata [8]. Nevertheless, nonstandard operators have been applied to all of these problems. Moreover, there has been much controversy over genetic algorithms using real-valued genes. Goldberg [11] has proposed his theory of virtual alphabets to explain the behavior of these under standard crossover, but a more general formulation that could take in a broader class of operators would still be valuable.

This paper extends the notion of intrinsic parallelism and the associated "Schema Theorem" to general non-string representations through the introduction of arbitrary equivalence relations. In doing so it provides a framework within which arbitrary genetic operators can usefully be analyzed.

*email: njr@castle.ed.ac.uk

[1] Also known as implicit parallelism.

The paper begins with a brief but careful review of standard genetic algorithms, reformulating slightly, making the connection between schemata and the equivalence relations that induce them, and introducing slightly unconventional notation to facilitate the transition to the more general formulation given later. This more general formulation involves the introduction in section 3 of general equivalence relations, and the Schema Theorem is expressed in terms of these. In section 4, interactions between the conventional k-ary representation and conventional operators are discussed. Schemata are generalized to *formae* in section 5, and "design principles" are suggested for the construction of useful equivalence relations, chromosomal representations, and crossover operators. In section 6, standard crossover operators are analyzed in the extended formalism, and deception is discussed in the context of general representations in section 7. In section 8 the theory is applied to the problem of real-valued genes. Results of experiments suggested by the theory are also given in section 8.

2. Traditional schemata

To search a space $\mathcal{S}$ with a genetic algorithm, the space is first mapped by a coding function ρ into a space of chromosomes $\mathcal{C}$, which the algorithm actually manipulates:

$$\rho : \mathcal{S} \longrightarrow \mathcal{C}.$$

Ideally ρ should be a bijection. A chromosome $\eta \in \mathcal{C}$ is usually taken to be a string of n genes $(\eta_1, \eta_2, \ldots, \eta_n)$ drawn from sets of alleles $(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n)$, so that the space of chromosomes is

$$\mathcal{C} \triangleq \mathcal{G}_1 \times \mathcal{G}_2 \times \cdots \times \mathcal{G}_n.$$

The conventional understanding of the way in which genetic algorithms search depends on the implicit introduction of certain equivalence relations on chromosomes. These equivalence relations identify chromosomes that share some genes. The set of all such equivalence relations for a chromosome with n genes can be represented by

$$\Psi \triangleq \{\, \square, \blacksquare \,\}^n,$$

where $\square$ is the "don't care" symbol that "matches" any allele, and $\blacksquare$ is used to indicate genes that must match for equivalence. Taking $n = 4$, a particular equivalence relation is then $(\square, \blacksquare, \square, \blacksquare)$, which is conveniently abbreviated to $\square\blacksquare\square\blacksquare$. Intuitively, the idea is that two chromosomes are equivalent under this equivalence relation if they have the same alleles wherever the definition has the $\blacksquare$ symbol. More carefully, calling each $\square$ or $\blacksquare$ symbol in the string describing an equivalence relation a *component*, given any equivalence relation $\sim \in \Psi$, with components $\sim_1, \sim_2, \ldots, \sim_n$ and given chromosomes $\eta, \zeta \in \mathcal{C}$

$$\eta \sim \zeta \iff (\forall\, i \in \mathbb{Z}_n \,(\sim_i = \blacksquare) \;:\; \eta_i = \zeta_i),$$

where $\mathbb{Z}_n = \{1, 2, \ldots, n\}$. That $\sim$ satisfies the conditions of symmetry, reflexivity, and transitivity, and is therefore an equivalence relation, follows immediately from this definition and the properties of $=$. These equivalence relations are in one-to-one correspondence with Walsh partitions, as described in Goldberg [9].

In practice the equivalence relations are rarely introduced explicitly, for the analysis depends only on the equivalence classes that they induce. In much the same way as for the equivalence relations, each equivalence class is conveniently expressed as a *schema*, a member of the set

$$\Xi \triangleq \mathcal{G}_1^\star \times \mathcal{G}_2^\star \times \cdots \times \mathcal{G}_n^\star,$$

where

$$\mathcal{G}_i^\star \triangleq \mathcal{G}_i \cup \{\square\}.$$

For example, $\xi = ab\square\square\cdots\square$ is the equivalence class of all those chromosomes η that have $\eta_1 = a$ and $\eta_2 = b$. Formally,

$$\eta \in \xi \iff (\forall i \in \mathbb{Z}_n\ (\xi_i \neq \square) :\ \eta_i = \xi_i)\,.$$

Plainly every chromosome is a member (or *instance*) of precisely 2^n schemata. (This can be seen by noting that replacing any subset of a chromosome's genes by $\square$ generates a schema that contains that chromosome, and that there are 2^n such subsets.)

Let the utility function that the genetic algorithm uses to guide its search be u. This associates with each chromosome a positive, real measure of its performance:

$$u : \mathcal{C} \longrightarrow \mathbb{R}^+.$$

It is useful to construct from u a measure

$$\mu : \Xi \longrightarrow \mathbb{R}^+,$$

which gives the utility of a schema as the average utility of all its members:

$$\mu(\xi) \triangleq \frac{1}{|\xi|} \sum_{\eta \in \xi} u(\eta),$$

where $|\xi|$ is the size of the equivalence class ξ. Noting that $\mathcal{C} \subset \Xi$, it is then immediately apparent that

$$\mu|_{\mathcal{C}} = u,$$

so that μ can be used to yield the utility of either a schema or a chromosome.

The introduction of a few more pieces of notation allows the statement of the "Schema Theorem," also known as the "Fundamental Theorem of Genetic Algorithms." The *defining positions* of a schema correspond to the $\blacksquare$ characters in the equivalence relation that induces it, so the defining positions

of $a\square b\square c$ are 1, 3, and 5. The *order* of a schema, $o(\xi)$, is equal to the number of defining positions it has, so $o(a\square b\square c) = 3$. The *defining length* of a schema, $\ell(\xi)$, is the maximum distance between any pair of defining positions, so $\ell(a\square b\square c) = 5 - 1 = 4$.

A fixed-size population $\mathfrak{B}(t)$ of chromosomes is maintained at time-step t. Each member of $\mathfrak{B}(t+1)$ is generated from one or more of the members of $\mathfrak{B}(t)$ by the application of the idealized genetic operators, typically crossover and mutation. A selection algorithm is employed to determine which chromosomes are to be used as parents. While many schemes are in use, the traditional approach is to use fitness-proportionate reproduction. The probability of picking $\eta \in \mathfrak{B}(t)$ as the principal[2] parent of any $\eta' \in \mathfrak{B}(t+1)$ is then taken to be:

$$P(\eta) = \frac{1}{|\mathfrak{B}(t)|}\frac{\mu(\eta)}{\bar{\mu}(t)}, \tag{2.1}$$

where

$$\bar{\mu}(t) \triangleq \sum_{\zeta \in \mathfrak{B}(t)} \mu(\zeta).$$

Finally, assume there is a set Ω of operators and that $\omega \in \Omega$ is applied with (independent) probability p_ω. Then let p_ω^ξ be the probability that a schema ξ will be disrupted by the application of this operator. That is, given an operator

$$\omega : \mathcal{C} \longrightarrow \mathcal{C},$$

p_ω^ξ is the probability that ξ does not contain the child whose parent it does contain:

$$p_\omega^\xi \triangleq P(\omega(\eta) \notin \xi \mid \eta \in \xi). \tag{2.2}$$

The Schema Theorem then bounds the expected number of instances $N_\xi(t+1)$ of each schema ξ in the population $\mathfrak{B}(t+1)$ by

$$\left\langle N_\xi(t+1) \right\rangle \geq N_\xi(t)\frac{\hat{\mu}_\xi(t)}{\bar{\mu}(t)}\left[1 - \sum_{\omega \in \Omega} p_\omega p_\omega^\xi\right], \tag{2.3}$$

where $\hat{\mu}_\xi(t)$ is the sample average for utility of ξ over all chromosomes in the population $\mathfrak{B}(t)$ that it contains. It is, in fact, extremely easy both to prove this theorem and to fill in bounds for p_ω^ξ for the standard operators. The only subtlety concerns the treatment of recombination operators that introduce extra parents.

Assume initially that the operators are all unary (asexual) so that every child has precisely one parent. Then the term outside the brackets follows directly from selection of the parent on the basis of fitness (equation (2.1)),

[2]The crossover operator takes two parents, and the second is also usually selected with the probability given.

and the bracketed term reduces the bound to take account of the fact that each operator, when applied, can destroy membership of the schema. The second term in the bracket is called the *disruption rate.*

When treating binary (sexual) operators, p_ω^ξ must be interpreted as the probability that ω destroys membership of a schema given the probability distribution used to select the other parent.

For example, using conventional one-point crossover, if both parents are selected according to equation (2.1) then the probability of disrupting a schema ξ is bounded above by the probability that the cross point falls between the outermost defining positions. To see this, it is sufficient to note that picking both parents in this way results in a doubling of the expected number of offspring from each schema to $2N_\xi(t)\hat{\mu}_\xi(t)/\bar{\mu}(t)$, and that if the cross point falls outside the defining region one of the two possible children is guaranteed to instantiate the given schema. Assuming that the cross point is chosen uniformly along the length, this gives $p_X^\xi = \ell(\xi)/(n-1)$, where the subscript X denotes crossover.

Similarly, the probability of "losing" at least one defining position as a result of mutation is bounded above by $p_m o(\xi)$, where p_m is the point mutation rate. Substitution in equation (2.2) restores the familiar form of the Fundamental Theorem:

$$\left\langle N_\xi(t+1) \right\rangle \geq N_\xi(t)\frac{\hat{\mu}_\xi(t)}{\bar{\mu}(t)}\left[1 - p_X\frac{\ell(\xi)}{n-1} - p_m o(\xi)\right].$$

Holland [15], assuming that only one of the parents was chosen on the basis of fitness, showed a closely related result.

3. From schemata to equivalence

Schemata, fundamental as they have been to understanding genetic algorithms, are merely a mathematical tool for analyzing and designing their behavior. The population of a genetic algorithm consists of individual chromosomes, and it is the utility of these that is actually measured. Holland observed that each evaluation of a chromosome can be regarded as a statistical sampling event that yields information about the sample averages for utility of *each* of the 2^n schemata of which it is an instance—the phenomenon referred to as *intrinsic parallelism.* However, the significance of the hat in equation (2.3), indicating the *observed* utility $\hat{\mu}_\xi(t)$ of a schema rather than its true fitness μ_ξ, cannot be overstated: only provided that there are correlations between the performance of different members of the equivalence classes (schemata) can the information collected in the population accurately guide the further exploration of the space. This critical point is discussed in greater detail in Radcliffe [19].

This observation suggests that any representation is useful only insofar as correlations between different portions of the search space can be expressed in terms of schemata. Of course, there is freedom to analyze the algorithm in any way desired, through the introduction of such equivalence relations

and classes as may be useful, and the objective of this work is to suggest a framework within which nonstandard equivalence relations and equivalence classes may be exploited. The careful formulation of the schema theorem in equation (2.3) is equally valid if ξ is interpreted as an arbitrary subset of $\mathcal{C}$ provided only that the coefficients p_ω^ξ are calculated correctly according to equation (2.2). In particular, it applies to an arbitrary equivalence class of *any* equivalence relation $\sim$ on $\mathcal{C}$ (or equivalently, given a bijective coding function, on the real search space $\mathcal{S}$). A general method for bounding these coefficients is now discussed.

A fairly general recombination operator X has the functional form

$$X : \mathcal{C} \times \mathcal{C} \times \mathcal{A}_X \longrightarrow \mathcal{C},$$

where $\mathcal{A}_X$ is a set of *control parameters* that determine which of the typically many possible crosses between two chromosomes occurs. For example, in the case of one-point crossover [15] $\mathcal{A}_X = \mathbb{Z}_{n-1}$, the set of possible cross points. Both two-point crossover [5] and Goldberg's partially-mapped crossover [12] use the control set $\mathcal{A}_X = \mathbb{Z}_{n-1}^2$, the set of all pairs of cross points, while uniform crossover (see, for example, [22]) has $\mathcal{A}_X = \{0,1\}^n$, the set of all n-bit binary masks. In the case of a few crossover operators (such as the Grefenstette's "Heuristic" Crossover [13]) the control set—if it is meaningful to talk of one at all—depends on the two chromosomes being crossed, but such operators are beyond the scope of this paper.

Given this structure, an often useful upper bound on the coefficient p_ω^ξ of equation (2.2), with $\omega = X$, can be calculated as follows. Let $\mathcal{A}_X^\xi$ be defined by

$$\mathcal{A}_X^\xi \triangleq \{\, a \in \mathcal{A}_X \mid \forall\, \eta \in \xi\ \forall\, \zeta \in \mathcal{C} \ :\ X(\eta, \zeta, a) \in \xi \,\}.$$

This is the set of parameter settings for which membership of ξ is passed to the child from the principal parent (η), regardless of the partner (ζ) chosen. Then p_X^ξ can be bounded by

$$p_X^\xi \leq \left(1 - w^\xi \frac{|\mathcal{A}_X^\xi|}{|\mathcal{A}_X|}\right), \tag{3.1}$$

where w^ξ is a weight to take account of the possibility that control parameters from $\mathcal{A}_X$ are not all selected with equal probability. In most cases (including all the crossover operators listed above) the choice is conventionally unbiased so that $w^\xi \equiv 1$. This bound (3.1) is, in effect, the one used by Holland to derive the Schema Theorem, and is typically used in deriving variations for other operators.

A similar approach can be taken for mutation operators. Conventional point mutation can be viewed as a collection of n operators

$$M_i : \mathcal{C} \times \mathcal{A}_i \longrightarrow \mathcal{C},$$

with $\mathcal{A}_i \equiv \mathcal{G}_i$, the allele sets. Then

$$M_i(\eta_1\eta_2 \ldots \eta_n, a) = \eta_1\eta_2 \ldots \eta_{i-1} a \eta_{i+1} \ldots \eta_n.$$

The coefficients p_i^{ξ} are then given by

$$p_i^{\xi} = \begin{cases} 0, & \text{if } \xi_i = \square, \\ (|\mathcal{G}_i| - 1)/|\mathcal{G}_i|, & \text{otherwise.} \end{cases}$$

If each gene is drawn from a set of k alleles, this yields

$$\sum_{i=1}^{n} p_i^{\xi} = o(\xi)\left(\frac{k-1}{k}\right).$$

4. Representations

There is little theory surrounding good representations for genetic algorithms. Holland [15] suggested subjecting the representation itself to adaptation, but the author is aware of no implementation in which this approach is adopted outside the domain of classifier systems. Schaefer's Argot Strategy [20] does alter the representation during the course of the search, but not in the manner suggested by Holland, nor in a way that is amenable to this analysis. Walsh function analysis is also sometimes used for postmortem analysis of why a genetic algorithm fails [10]. Goldberg [8], however, suggested the following two principles for good representations:

> *The Principle of Meaningful Building Blocks:*
> The user should select a [representation] so that short, low-order schemata are relevant to the underlying problem and relatively unrelated to schemata over other [defining] positions.
>
> *The Principle of Minimal Alphabets:*
> The user should select the smallest alphabet that permits a natural expression of the problem.

The analysis presented here focuses on the interaction between the chromosomal representation, *some* set of equivalence relations Ψ over the chromosomes, and the genetic operators used. Goldberg's principles are formulated with respect to conventional chromosomal representations (n-tuples of genes drawn from sets of alleles) analyzed with conventional schemata.

His first principle requires three things. First, it emphasizes the point made in the previous section that as many equivalence classes (schemata) as possible should contain chromosomes that have correlated performance. Second, by seeking to reduce the defining length and order of good schemata, it attempts to minimize the likelihood of disruption by the genetic operators. Finally, it tries to ensure that recombination of (instances of) different schemata works in a useful manner. The second principle attempts to maximize the degree of intrinsic parallelism available to the algorithm by ensuring that each chromosome is an instance of many schemata.

5. Formae

The above considerations (and others) lead to the following proposals for constructing useful equivalence relations, good representations, and suitable sets of operators. These principles are not all necessary for an effective genetic algorithm and are certainly not sufficient for it, but might be expected to characterize good representations. To emphasize the link between these equivalence classes and schemata, the former will be referred to as *formae*,[3] and the number of formae induced by an equivalence relation will be referred to as the *precision* of the relation and the formae it induces.[4] From this point on, Ξ will be interpreted as the set of all formae induced by the equivalence relations in Ψ.

Two formae ξ and ξ' will be said to be *compatible* if it is possible for a chromosome to be an instance of both ξ and ξ'.[5] Denoting this by $\xi \bowtie \xi'$, a more careful statement is

$$\xi \bowtie \xi' \iff \xi \cap \xi' \neq \emptyset \ .$$

Design principles

1. (Minimal redundancy) *The representation should have minimal redundancy; such redundancy as exists should be capable of being expressed in terms of the equivalence relations used.*
Ideally, each member of $\mathcal{S}$ should be represented by only one chromosome in $\mathcal{C}$. This is highly desirable in order to minimize the size of the search space. If redundant solutions are present, but are related by one of the equivalence relations used, then the genetic algorithm should effectively be able to "fold out" the redundancy (see principle 4); otherwise it is doomed to treat redundant solutions as unrelated.

2. (Correlation within formae) *Some of the equivalence relations, including some of low precision, must relate chromosomes with correlated performance.*
This ensures that useful information can be gathered about the performance of a forma by sampling its instances. Such information is used to guide the search. The emphasis is placed on low-precision formae because these will generally be less likely to be disrupted by the application of genetic operators, and are also more likely to be compatible with one another.

3. (Closure) *The intersection of any pair of compatible formae should itself be a forma.*

[3] Although Holland chose the neuter form for the Latin noun schema, there is no option but to choose the feminine form of its synonym, forma.

[4] In the case of schemata and genes with k alleles, the precision is k^o, where o is the order of a schema.

[5] The term *competitive schemata* has sometimes been used to describe those that here would be called *incompatible*.

This ensures that solutions can be specified with different degrees of accuracy and allows the search to be refined gradually. Clearly the precision of formae thus constructed will be at least as high as that of the higher precision of the intersecting formae.

4. (Respect) *Crossing two instances of any forma should produce another instance of that forma.*
Formally, it should be the case that

$$\forall \xi \in \Xi \, \forall \eta \in \xi \, \forall \zeta \in \xi \, \forall a \in \mathcal{A}_X : \; X(\eta, \zeta, a) \in \xi,$$

where X is the crossover operator. In this case the crossover operator will be said to *respect* the equivalence relations (and their formae). This ensures that the algorithm can converge on good formae, and implies, for example, that $X(\eta, \eta, a) = \eta$ (assuming that equivalence relations of maximum precision specify chromosomes completely). It also effectively reduces the disruption rate in the Schema Theorem, though a more accurate value for p_X^ξ than that given in equation (3.1) is needed to see this. Informally, respect requires that any properties that parents share, and that are capable of expression in terms of the formae, be passed on to all their children.

5. (Proper assortment) *Given instances of two compatible formae, it should be possible to cross them to produce a child that is an instance of both formae.*
Formally,

$$\forall \xi \in \Xi \, \forall \xi' \in \Xi \, (\xi \bowtie \xi') \, \forall \eta \in \xi \, \forall \eta' \in \xi' \, \exists a \in \mathcal{A}_X : X(\eta, \eta', a) \in \xi \cap \xi'. \tag{5.1}$$

This relates to Goldberg's "meaningful building blocks," of which he writes ([8], page 373)

> Effective processing by genetic algorithms occurs when *building blocks*—relatively short, low order schemata with above average fitness values—combine to form optima or near-optima.

A crossover operator that obeys equation (5.1) seems much more likely to be capable of recombining "building blocks" usefully, and any crossover operator that obeys this principle will be said *properly to assort* formae. Informally, proper assortment requires crossover to be capable of mixing compatible properties from the two parents.

6. (Ergodicity) *It should be possible, through a finite sequence of applications of the genetic operators, to access any point in the search space $\mathcal{S}$ given any population $\mathfrak{B}(t)$.*
This provides the *raison d'être* for the mutation operator.

6. Crossover and formae

It is instructive to examine the way standard crossover operators interact with schemata (the "standard" formae) to see whether they respect and properly assort them in the sense of principles 4 and 5. The crossover operators that have traditionally been used are one- and two-point crossover. More recently, attention has focused on multi-point crossover and the so-called "uniform" crossover operator. Eshelman, *et al.* [7] have also discussed "shuffle" crossover operators. Recall that uniform crossover makes an independent random choice as to which of the parents the allele at each locus is drawn from, and shuffle crossover shuffles the (effective) order of the genes before crossing over, removing "positional" bias in the sense of Eshelman, *et al.* [7]. All of these operators respect schemata, for it is plain that under all of them a child will be an instance of any schema containing both its parents. Only the uniform and shuffle crossover operators, however, properly assort schemata.

To see this, consider the chromosomes and schemata $1010 \in 1\square1\square$ and $0101 \in \square1\square1$. Plainly the two given schemata are compatible, with intersection 1111, but neither one- nor two-point crossover can cross them to produce 1111 in a single step.[6] It should be clear that this kind of problem will arise for n-point crossover with any *fixed* n. Both uniform and shuffle crossovers, however, can recombine the two chromosomes as required (albeit with low probability) and it should be apparent that they always respect schemata.

7. Deception

Deception, like most work on genetic algorithms, has only hitherto been considered in the context of classical schemata, and has been rigourously defined by Goldberg [10]. If, however, more general formae are considered, then it becomes necessary to consider deception in terms of the formae under consideration.

Recall that, classically, a function-coding combination is said to be *partially deceptive* if some low-order schemata lead away from the optimum, and is *fully deceptive* if all lower-order schemata lead away from the optimum. This indicates that defining positions on a schema that are "wrong" (carry a different allele from the optimal chromosome at that locus) lead to higher utilities for the schema.

This definition cannot immediately be carried over to the case of general formae because it is not meaningful to talk of "genes," "defining positions,"

[6] Of course, Holland [15] advocated using inversion with one-point crossover. The aim of this was to bring co-adapted sets of alleles closer together on the chromosome, and in these circumstances proper assortment is probably not relevant. Since inversion is rarely used, however, this case is not considered in detail here. For a discussions of inversion see Holland [15], Goldberg [8], and Radcliffe [19].The fact that uniform crossover is more disruptive to short schemata of a given order than is one-point crossover becomes a consideration only if the layout of the genes on the chromosome is believed accurately to reflect the degree of linkage between the properties they code.

and so forth for an arbitrary forma. The following definitions, however, seem to capture the spirit of deception, which in the context of formae will be termed f-deception. Assume that there is a unique global optimum represented by $\eta^\star \in \mathcal{C}$, that is,

$$\forall \eta \in \mathcal{C} \setminus \{\eta^\star\} : \mu(\eta) < \mu(\eta^\star).$$

Let the formae induced by any relation $\sim \in \Psi$ of precision k be $\xi_\sim^{(1)}, \xi_\sim^{(2)}, \ldots, \xi_\sim^{(k-1)}, \xi_\sim^\star$, where $\eta^\star \in \xi_\sim^\star$. Then a representation will be said to be *partially f-deceptive* with respect to Ψ if

$$\exists \sim \in \Psi : \max_i \mu\left(\xi_\sim^{(i)}\right) > \mu(\xi_\sim^\star).$$

In other words, a representation is partially f-deceptive (with respect to the equivalence relations in Ψ) if the global optimum is not in the equivalence class (forma) of highest utility for all of the equivalence relations.

In the same spirit, let $\Psi^\star$ be the set of equivalence relations of precision lower than the size of the search space (i.e., those relations that do not induce only singleton formae). *Full f-deception* can then be defined as follows:

$$\forall \sim \in \Psi^\star : \max_i \mu\left(\xi_\sim^{(i)}\right) > \mu(\xi_\sim^\star).$$

This says that for every low-precision equivalence relation the optimum $\eta^\star$ falls outside the equivalence class of top utility.

8. Real-valued problems

Conventional wisdom holds that real-valued problems are best tackled using binary representations because this allows the maximum level of intrinsic parallelism to be achieved. (Recall that each chromosome is an instance of 2^n schemata, and that n is maximized for binary genes.) In practice, however, this intrinsic parallelism can be exploited only when schemata relate solutions with correlated performance. To emphasize this critical point, notice that if the size of the search space $\mathcal{S}$ is s, there are $s!$ possible bijective coding functions

$$\rho : \mathcal{S} \longrightarrow \mathcal{C},$$

almost all of which effectively destroy patterns over the search space. To see this, imagine randomly selecting a mapping from these $s!$ choices, and notice that this is exactly equivalent to choosing a (unique) random chromosome from $\mathcal{C}$ to represent each structure in $\mathcal{S}$. Under these circumstances it should be clear that gathering information about the performance of *any* subset of the chromosomes provides *no* information about the performance of the remaining structures—those represented by the untested chromosomes. Nevertheless, the Schema Theorem (equation (2.3)) will be obeyed for every one of the $s!$ representations.

In such circumstances, the search could not be effective except by chance simply because almost none of the schemata would relate chromosomes with correlated performance. In other words, schemata are not useful formae in this context. (Holland's observations ([15], page 142) about "enriched schemata" appear initially to refute this claim, but on closer analysis do not. This is discussed in detail in Radcliffe [19] (pages 17–18, Compressed Edition).)

In effect the results and arguments presented thus far in this paper can be seen as a critique of the idea that there is a single, all-embracing representation and set of operators that can reasonably be expected to tackle all or most search problems effectively. The focus here is on finding sets of formae that characterize the regularities in the particular problem or class of problems under consideration, and developing operators that manipulate these to good effect. Thus, for example, rather than seeing a function-representation pair as deceptive, deception (or f-deception) is seen as characterizing a mismatch between the set of formae used (together with the operators used to manipulate them) and the regularities in the space being searched.

To explore these ideas further, the next sections discuss general binary representations for real-valued problems and two types of regularities for which it might be desirable to develop formae to characterize. The ideas are made more concrete by applying them to a familiar set of functions.

8.1 Binary representations

The great strength of binary representations lies in their versatility: different schemata relate chromosomes on quite different bases. Indeed, their robustness is demonstrated by the wide variety of problems that have been tackled successfully using binary representations. For example, consider the natural coding for a real number in the range $[\alpha, \beta]$, with N divisions,

$$\rho(x) = \left\lfloor N \frac{x - \alpha + \delta}{\beta - \alpha} \right\rfloor,$$

where $\delta = 1/2N$. The schema $1\square\square\cdots\square$ then specifies the upper half-space $x > (\alpha + \beta)/2$, whereas the schema $\square\square\cdots\square 1$ specifies alternate strips of width 2δ across the space, capturing some possible periodicities.

The problem lies in the fact that the schemata are far from uniform over the space. Suppose that $\alpha = 0$ and $\beta = 1.5$ with 16 divisions. Then $\rho(0) = 0000$, $\rho(0.7) = 0111$, $\rho(.8) = 1000$, and $\rho(1.5) = 1111$. It is possible to specify the interval $[0.8, 1.5]$ exactly using the low-order schema $1\square\square\square$, whereas there is no schema of *any* order that specifies any range that crosses the "Hamming cliff" between 0.7 and 0.8. Caruna and Schaffer [4] advocate using Gray coding to avoid this (and other) problems, but the interpretation of schemata is then even less obvious, and serious problems with such schemes have been pointed out by Goldberg [11]. Nor is locality the only problem: while some periodicities that are multiples of powers of two in the discretization size are easily characterized using schemata, periodicities that are multiples of powers

of three, for example, are incapable of being so represented. The problem seems to lie in the fact that relatively few of the schemata seem to be induced by equivalence relations that group together "useful" sets of points.

Whether more useful equivalence relations can be developed depends very much on how much insight can be gained into likely kinds of structure in the problem. For function optimization over intervals in $\mathbb{R}^n$, locality and periodicity seem like obvious—though not universal—starting points.

8.2 Locality

Simplifying to functions of one real variable ($\mathcal{S} \subset \mathbb{R}$), suitable equivalence relations for capturing locality are intervals specified by a position and a radius. Let $\mathrm{B}[c,r)$ be the half-open interval $c - r \leq x < c + r$, $r \in \mathbb{R}^+$, and let $\mathrm{B}[c,0) = \{c\}$. Then the equivalence relation specified by position p and radius r is

$$\eta \sim \zeta \iff (\exists\, k \in \mathbb{Z} :\ \eta, \zeta \in \mathrm{B}[p + 2kr, r))$$

with formae

$$\{\, \mathrm{B}[p + 2kr, r) \mid k \in \mathbb{Z} \,\}.$$

Thus any interval $[\alpha', \beta')$ is an equivalence class under *some* equivalence relation.

Moving back to the more general problem of searching a space $\mathcal{S}$ that has n real-valued parameters,

$$\mathcal{S} = \prod_{i=1}^{n} \mathcal{I}_i,$$

where

$$\mathcal{I}_i = [\alpha_i, \beta_i] \subset \mathbb{R},$$

a suitable set of "locality" equivalence relations Ψ^L can be defined as

$$\Psi^L = \prod_{i=1}^{n} \mathcal{I}_i^\star,$$

where

$$\mathcal{I}_i^\star = \mathcal{I}_i^2 \cup \{\square\}.$$

(The "don't care" character is strictly redundant, but is left in for notational convenience.) This induces formae that can be described using exactly the same set as for Ψ^L, namely

$$\Xi^L = \prod_{i=1}^{n} \mathcal{I}_i^\star.$$

Thus a typical forma ξ with $n = 3$ can be written as $\langle\, \mathrm{B}[0.2, 0.1), \square, \mathrm{B}[0.5, 0.2) \,\rangle$, with the interpretation that a chromosome η is an instance of ξ if $0.1 \leq \eta_1 < 0.3$ and $0.3 \leq \eta_3 < 0.7$. Formally,

$$\eta \in \xi \quad \Longleftrightarrow \quad (\forall i \in \mathbb{Z}_n \; (\xi_i \neq \square) : \; \eta_i \in \xi_i).$$

If these equivalence relations are to be used, then a crossover operator should be constructed that both respects and properly assorts the formae they induce. Standard crossover with real genes would respect them, but would fail properly to assort them.

An example should make this clear. The sub-formae[7] $\mathrm{B}[0.4, 0.2)$ and $\mathrm{B}[0.6, 0.2)$ are compatible with intersection $\mathrm{B}[0.5, 0.1)$, but given genes $0.3 \in \mathrm{B}[0.4, 0.2)$ and $0.7 \in \mathrm{B}[0.6, 0.2)$ it is impossible for standard crossover to generate any value in $\mathrm{B}[0.5, 0.1)$ since the result of such a cross will always be either 0.3 or 0.7. The presence of Hamming cliffs also makes it immediately clear that standard crossover with binary genes will not respect these formae.

A more suitable crossover operator is

$$X^F : \mathcal{C} \times \mathcal{C} \times [0,1]^n \longrightarrow \mathcal{C}$$

with

$$X_i^F(\eta, \zeta, r) = r_i|\eta - \zeta| + \min\{\, \eta, \zeta \,\}.$$

which will be called *flat* crossover[8] (see figure 1). Given a pair of real-valued genes, this operator returns a random value within the interval between them. The choice is uniform provided each r_i is chosen uniformly. (The control set here is $\mathcal{A}_F = [0,1]^n$.) Plainly this operator respects formae from Ξ^L, for if the two genes have the same value then the interval they define has zero width. Moreover, compatible formae ξ and ξ' have overlapping intervals at each locus. Given $\eta \in \xi$ and $\eta' \in \xi'$, it is clearly possible to choose a set of r_i such that each gene of the child sits within the intersection $\xi \cap \xi'$.

Thus X^F respects and properly assorts formae from Ξ^L, composed of intervals of arbitrary widths in the search space $\mathcal{S}$. A genetic algorithm using this might be expected to perform well on a real-valued problem for which locality *is* the appropriate kind of equivalence to impose on solutions, utilizing the intrinsic parallelism that derives from each chromosome's being an instance of many locality formae.

Two related problems, however, remain. The first is the question of a suitable mutation operator. The second problem concerns a bias in the operator, namely that it systematically biases the search away from the ends of the intervals, violating ergodicity in the sense of principle 6.

Recall that the role of mutation for k-ary string representations is usually understood to be that of keeping the gene pool well-stocked, the fear being that, if an allele for some gene is not present in any member of the population, crossover will never be able to generate it and will thus not have access to

[7] Defined on a single gene.

[8] Affectionately known as "top hat."

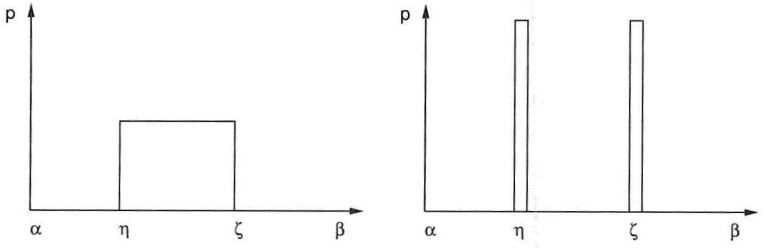

Figure 1: The graph on the left shows the probability distribution function for the one-dimensional flat crossover operator "top hat" when crossing η and ζ. The graph on the right shows the corresponding distribution function for traditional crossover with real genes and uniform crossover with real genes. The distribution function for traditional crossover with binary genes looks very different according to the chromosomes being crossed. For example, using 4-bit binary coding on the range $[0, 15]$, 7 (=0111B) and 8 (=1000B) can cross to produce *any* child under uniform crossover, whereas 0 (=0000B) and 8 (=1000B) can only produce copies of themselves.

the entire search space. This observation, which motivated the principle of ergodicity, suggests that the two problems mentioned can be tackled together by defining a mutation operator that only inserts extremal values into the gene pool, thus countering the bias of X^F. As before, given n genes per chromosome (now real-valued), a set of n point mutation operators is defined according to

$$M_i^R : \mathcal{C} \times \{ \alpha_i, \beta_i \} \longrightarrow \mathcal{C}$$

with

$$M_i^R(\eta_1 \eta_2 \ldots \eta_n, a) = \eta_1 \eta_2 \ldots \eta_{i-1} a \eta_{i+1} \ldots \eta_n.$$

The difference between this and standard mutation is that, instead of using the interval $[\alpha_i, \beta_i]$ as the control set $\mathcal{A}_i$, only the endpoints α_i and β_i are now used. If both parents are selected according to fitness, such mutations should be applied *before* crossing over to reduce the probability of generating a child of very low fitness, which then fails to reproduce.

As an illustration of these ideas, De Jong's standard test suite of functions [6] were examined using both a standard binary representation with uniform crossover and a real-valued representation using flat crossover as defined above. Following Eshelman, *et al.* [7] the functions are described here only summarily in table 1.

Of the five functions, good performance might reasonably be expected on f_1, f_2, and f_4, which are (essentially) smooth, whereas very poor performance would be expected on f_5. Reasonable performance might also be anticipated on f_3, which, though not smooth, is reasonably local in nature. The results for off-line and best-seen performance are shown in figures 2 through 7. An

fn.	dim	space size	description
f_1	3	1.0×10^9	parabola
f_2	2	1.7×10^6	Rosenbrock's Saddle
f_3	5	1.0×10^{15}	step function
f_4	30	1.0×10^{72}	noisy quadratic
f_5	2	1.7×10^{10}	Shekels foxholes
f_6	2	1.7×10^{10}	Random foxholes

Table 1: De Jong's test suite of functions f_1 through f_5, augmented by random foxholes.

extra function f_6 is also included, which is a variation on Shekel's foxholes in which the positions of the foxholes are random rather than in a regular grid.

A comparison is shown between the same genetic algorithm using both binary and real representations, with parameters selected to give good performance with binary representations. Following Schaffer, *et al.* [21], the point mutation rate was made inversely proportional to the chromosome length, and was thus higher when using real representations (with fewer genes) than for their binary counterparts. Baker's Stochastic Universal Sampling procedure [2] and rank-based selection, broadly *à la* Baker [1], were used. Flat crossover and extremal mutations, as described above, were used for the real-valued case, and uniform crossover was used for the binary trials. The results are all averages over 100 runs.

As predicted, flat crossover with real genes performs extremely well on the smooth f_1, f_2, and f_4, out-performing binary representations. On f_3, although less effective than the binary case, the global optimum is still consistently found in reasonable time.

The results for Shekel's foxholes are rather more surprising. With the standard foxhole configuration (a 5×5 grid with spacing 16), the binary representation appears superior, though the real representation performs amazingly well considering the crossover operator it uses was only designed to respect *locality* formae, which have no obvious relevance to this problem. Notice, however, that points differing by 16.384 are very close in Hamming distance under the binary representation, making it easy to hop from one foxhole to another. For this reason, a second set of trials was performed using foxhole coordinates chosen at random. In this case, the real representation using flat crossover gives slightly superior performance to the binary representation.

8.3 Periodicity

Dealing with general periodicities, unsurprisingly, is harder. Constructing equivalence relations Ψ^P capable of capturing general periodicities is not difficult: suitable relations are specified by a position p, a radius r (to allow

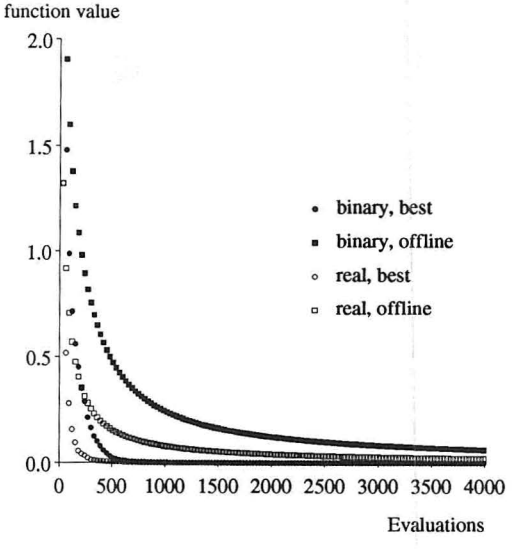

Figure 2: De Jong's f_1. The "real" traces use the "flat" crossover operator, which chooses a random value in the range bounded by the parents' genes. The binary trace is the same algorithm using binary uniform crossover.

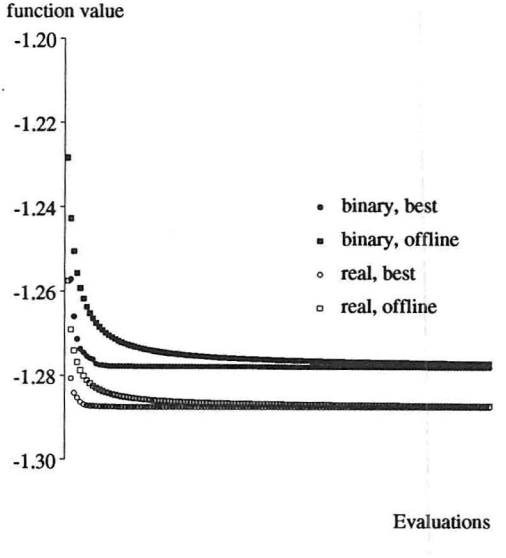

Figure 3: De Jong's f_2.

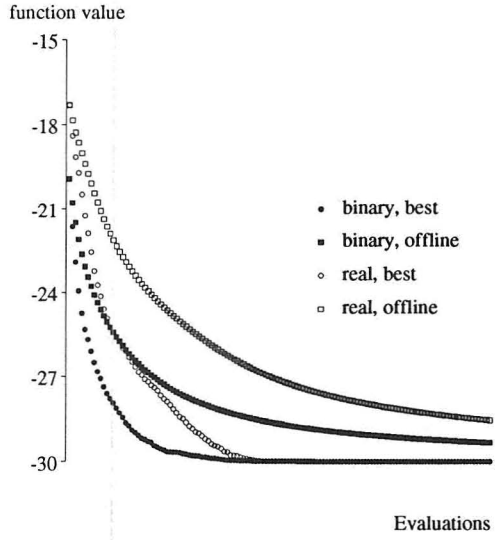

Figure 4: De Jong's f_3.

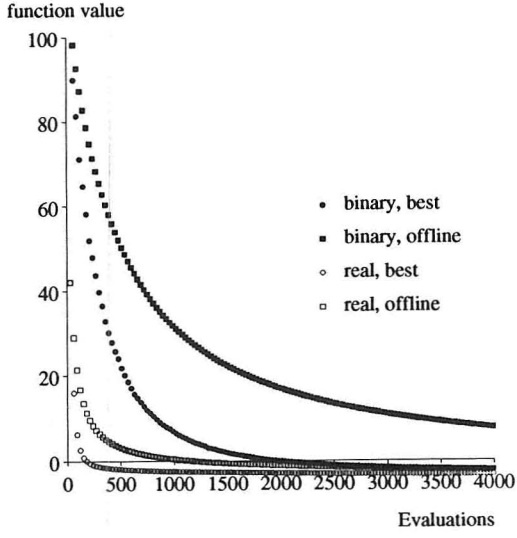

Figure 5: De Jong's f_4.

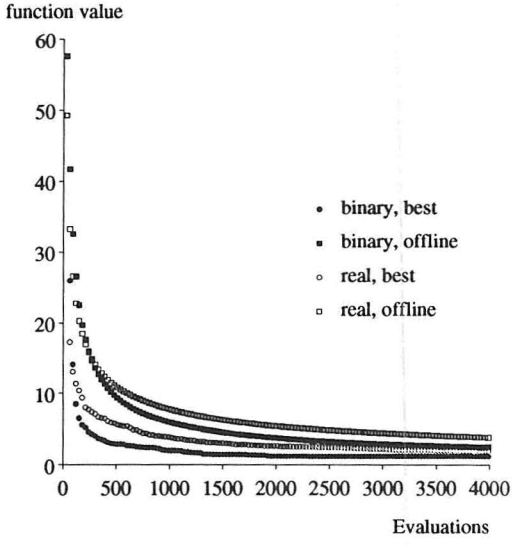

Figure 6: Results for De Jong's f_5.

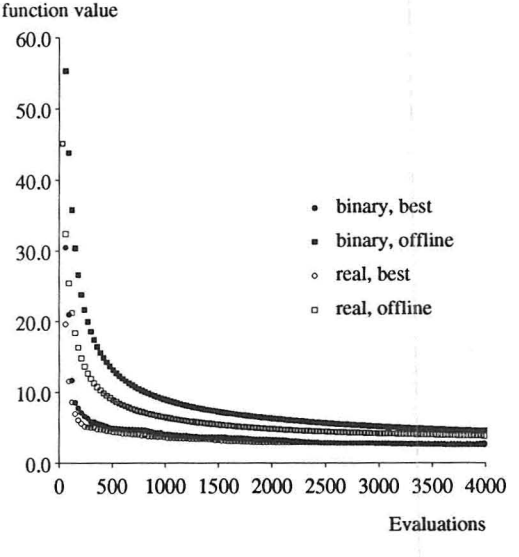

Figure 7: Results for f_6: randomly placed foxholes.

ξ_1	0			5		10		15	20
ξ_2	0		4		8	12	16		20
ξ_3	0								20
ξ_4			4			10	16		

Figure 8: Four "locality" formae. Each number is the center of a half-open interval of width half. $\xi_3 = \xi_1 \cap \xi_2$, and ξ_4 is compatible with both ξ_1 and ξ_2.

for fuzziness), and a period T that is an integral multiple of r. Given these, and again simplifying to functions of one real variable, two chromosomes are equivalent if they lie in intervals of radius r centered about points separated by a multiple of the period T. Formally,

$$\eta \sim \eta' \iff \Big(\exists\, k, k' \in \mathbb{Z} :\ \eta \in \mathrm{B}[p + kT, r) \text{ and } \eta' \in \mathrm{B}[p + k'T, r)\Big).$$

These equivalence relations are extremely flexible, subsuming the previous "locality" relations immediately by setting T to zero. If a crossover operator could be constructed that both respected and properly assorted these relations it might be expected that an extremely powerful algorithm for real-valued problems would result.

Sadly, no such operator exists. To see this, consider the formae ξ_1 to ξ_4 in figure 8, each with radius $r = 0.5$. The numbers in figure 8 indicate the centers of the intervals that the formae comprise, so ξ_3 consists of $\mathrm{B}[0, 0.5)$ and $\mathrm{B}[20, 0.5)$. Notice that $\xi_3 = \xi_1 \cap \xi_2 \neq \emptyset$, so $\xi_1 \bowtie \xi_2$. Consider chromosomes $10 \in \xi_1 \cap \xi_4$ and $4 \in \xi_2 \cap \xi_4$. If a crossover operator X^P is to respect ξ_4 then it must be the case that for all $a \in \mathcal{A}_P$: $X^P(4, 10, a) \in \xi_4$; that is, all possible children of 4 and 10 must be members of ξ_4. If it is to assort ξ_1 and ξ_2 properly then there must be some $a' \in \mathcal{A}_P$ for which $X(4, 10, a') \in \xi_1 \cap \xi_2$, that is, it must be possible to cross 4 and 10 to produce a chromosome that is an instance of both ξ_1 and ξ_2. These conditions are incompatible, however, because $\xi_4 \cap \xi_1 \cap \xi_2 = \emptyset$.

It should be emphasized that this is not a failure of the forma analysis, which has simply revealed that general periodicities are extremely hard for a genetic algorithm to be sensitive to. It has been demonstrated that no crossover operator can both fully respect and properly assort the formae Ξ^P induced by Ψ^P, but it is quite possible for an operator partially to respect and assort them. Indeed, uniform crossover does this. Whether an operator can be constructed that better respects and/or assorts Ξ^P remains an open question.

9. Conclusion

Intrinsic parallelism, the key concept underpinning genetic search, has been shown not to be restricted to k-ary string representations. Given a suitable set of equivalence relations and a crossover operator that both respects and properly assorts its equivalence classes (formae) without excessive disruption, any genetic algorithm will exhibit intrinsic parallelism. These ideas have been applied to standard crossover operators to provide another insight into the sometimes-claimed superiority of the uniform crossover operator over traditional one- and two-point crossover, and to apply genetic algorithms more effectively to some real-valued problems. They could equally well be applied to other problems for which k-ary string representations and schemata are not obviously appropriate. Such areas include neural networks, the TSP, and graph optimization.

Acknowledgments

I would like to thank Mike Norman for many useful late-night discussions over a period of two years, without which this work would have been unlikely to come to fruition, and also for his careful criticism of earlier drafts of this paper. I am also indebted to Richard Rohwer for reading a draft of the paper.

Some of this work was supported by the Edinburgh Parallel Computing Centre, a multidisciplinary project that receives funding from the Department of Trade and Industry, the Computer Board, and the Science and Engineering Research Council. The author acknowledges support from the Univerity of Edinburgh and from Industrial Affiliates.

References

[1] James Edward Baker, "Adaptive Selection Methods for Genetic Algorithms," in *Proceedings of an International Conference on Genetic Algorithms* (Lawrence Erlbaum Associates, Hillsdale, 1985).

[2] James Edward Baker, "Reducing Bias and Inefficiency in the Selection Algorithm," in *Proceedings of the Second International Conference on Genetic Algorithms* (Lawrence Erlbaum Associates, Hillsdale, 1987).

[3] Richard K. Belew, John McInerny, and Nicol N. Schraudolph, "Evolving Networks: Using the Genetic Algorithm with Connectionist Learning," CSE Technical Report #CS90–174 (University of California at San Diego, La Jolla, 1990).

[4] Richard A. Caruna and J. David Schaffer, "Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms," in *Proceedings of the 5th International Conference on Machine Learning* (Morgan Kaufmann, Los Altos, 1988).

[5] Kenneth A. De Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems" (Doctoral dissertation, University of Michigan, 1975), *Dissertation Abstracts International*, **36**(10), 5140B (University Microfilms No. 76-9381).

[6] Kenneth A. De Jong, "A Genetic-Based Global Function Optimization Technique," Technical Report 80-2 (University of Pittsburgh, 1980).

[7] Larry J. Eshelman, Richard A. Caruna, and J. David Schaffer, "Biases in the Crossover Landscape," in *Proceedings of the Third International Conference on Genetic Algorithms* (Morgan Kaufmann, San Mateo, 1989).

[8] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, 1989).

[9] David E. Goldberg, "Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction," *Complex Systems*, **3** (1989) 129–152.

[10] David E. Goldberg, "Genetic Algorithms and Walsh Functions: Part II, Deception and Its Analysis," *Complex Systems*, **3** (1989) 153–171.

[11] David E. Goldberg, "Real-Coded Genetic Algorithms, Virtual Alphabets, and Blocking," *Complex Systems*, forthcoming.

[12] David E. Goldberg, and Robert Lingle, Jr., "Alleles, Loci and the Travelling Salesman Problem," in *Proceedings of an International Conference on Genetic Algorithms* (Lawrence Erlbaum Associates, Hillsdale, 1985).

[13] John Grefenstette, Rajeev Gopal, Brian Rosmaita, and Dirk Van Gucht, "Genetic Algorithms for the Travelling Salesman Problem," in *Proceedings of an International Conference on Genetic Algorithms* (Lawrence Erlbaum Associates, Hillsdale, 1985).

[14] Steven Alex Harp, Tariq Samad, and Aloke Guha, "Towards the Genetic Synthesis of Neural Networks," in *Proceedings of the Third International Conference on Genetic Algorithms* (Morgan Kaufmann, San Mateo, 1989).

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975).

[16] David J. Montana and Lawrence Davis, "Training Feedforward Neural Networks Using Genetic Algorithms," in *Proceedings of the 5th International Conference on Machine Learning* (Morgan Kaufmann, Los Altos, 1988).

[17] Michael Norman, "A Genetic Approach to Topology Optimisation for Multiprocessor Architectures," Edinburgh preprint (1988).

[18] Dominic M. N. Prior, Nicholas J. Radcliffe, Michael G. Norman, and Lyndon J. Clarke, "What Price Regularity?" *Concurrency: Practice and Experience*, **2**(1) (1990) 55–78.

[19] N. J. Radcliffe, "Genetic Neural Networks on MIMD Machines" (Doctoral dissertation, Edinburgh University, 1990).

[20] G. Shaefer Craig, "The ARGOT Strategy: Adaptive Representation Genetic Optimizer Technique," in *Proceedings of the Third International Conference on Genetic Algorithms* (Morgan Kaufmann, San Mateo, 1989).

[21] J. David Schaffer, Richard A. Caruna, Larry J. Eshelman, and Rajarshi Das, "A Study of the Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimisation," in *Proceedings of the Third International Conference on Genetic Algorithms* (Morgan Kaufmann, San Mateo, 1989).

[22] Gilbert Syswerda, "Uniform Crossover in Genetic Algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms* (Morgan Kaufmann, San Mateo, 1989).

[23] Darrell Whitley, Timothy Starkweather, and D'Ann Fuquay, "Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator," in *Proceedings of the Third International Conference on Genetic Algorithms* (Morgan Kaufmann, San Mateo, 1989).

Received October 11, 1990.