Appears in "Parallel Problem Solving from Nature 2", Eds: R. Manner, B. Manderick, Elsevier (Amsterdam), pp259-268, 1992.

Non-Linear Genetic Representations

Nicholas J. Radcliffe

Edinburgh Parallel Computing Centre, University of Edinburgh, King's Buildings, Edinburgh, EH9 3JZ, Scotland

Abstract

The limitations of linear chromosomes and conventional recombination operators are reviewed. It is argued that there are at least three classes of problems for which such representations and operators are likely to be ineffective. Methods for constructing operators which manipulate more complex structures with evolutionary search methods are presented, and it is argued that whenever possible, genetic operators and analogues of schemata should be defined directly in space of phenotypes, rather than in the genotype (representation) space.

1. INTRODUCTION

This paper considers the implications of earlier theoretical work on evolutionary search concerning the relationship between genetic representation, idealised genetic operators and performance correlations between solutions in the search space. The central thesis of the paper is that for many problems conventional linear chromosomes and recombination operators are inadequate for effective genetic search, and that for general problems non-linear representations are required. Section 2 considers representation issues in the abstract, focusing initially on the genotype-phenotype mapping, intrinsic parallelism and the ability of schemata to capture important regularities in performance characteristics in the search space, before discussing the various ways in which different workers have tried to respond to perceived limitations. Section 3 discusses various generalisations of the standard analysis of genetic algorithms, which are then used to explore three specific limitations of linear chromosomes with conventional operators. Briefly, these limitations arise when schemata are unable to describe important subsets of the search space (section 4), when key characteristics of solutions cannot by independently assigned (section 5), and when constraints are involved (section 6). The paper closes with a summary.

2. REPRESENTATION

2.1. The Genotype-Phenotype Mapping

In natural systems the distinction between genotype and phenotype is reasonably clear and has physical meaning: DNA is quite literally decoded to build a physical realisation of the system it "describes". It is therefore unsurprising that the same distinction is made in genetic algorithms and *evolutionstrategies.* A distinction between genotype and phenotype is not, however, necessary for evolution, for while it is trivially the case that in order to use a computer to manipulate any structures which are not naturally resident in a computer, a (computer) representation of them must be chosen, there is no requirement that the idealised genetic operators used in evolutionary search be defined with respect to the chosen representation, nor indeed with respect to any specific representation. If genetic operators are defined with respect to the structures which form the search space (the phenotypes), then it suffices that this specification be capable of being implemented reasonably efficiently on a computer using some representation.

2.2. Binary Representations and Intrinsic Parallelism

The "American" school of genetic algorithms, initiated by John Holland with the publication of his seminal book *Adaptation in Natural and Artificial Systems* (Holland, 1975), has tended to focus on linear string representations, and in particular on binary representations. These are most often manipulated by a rather small set of recombination operators. Initially, simple ("one-point") crossover was favoured, but was gradually superseded by two-point crossover, sometimes in its "reduced surrogate" form (Booker, 1987). More recently, n-point crossover with $n > 2$ (see Eshelman *et al.*, 1989) and so-called "uniform crossover" (see Syswerda, 1989 and Spears & DeJong, 1991) have also gained wider use. There have always been exponents of alternative operators and representations, most notably Lawrence Davis (e.g. Davis, 1991), and perhaps the sharpest controversy has centred upon whether it is appropriate to use binary representations—whether traditional or Gray-coded (Caruana & Schaffer, 1988)—for problems in which the parameters under consideration are real (Goldberg, 1990c). In the *Evolution-strategie* school, a more relaxed attitude towards alternative operators and representations appears always to have been taken (Baeck *et al.*, 1991).

It is worth pointing out that it has always been accepted even by the mainstream of the American school that there are certain problems for which alternative operators and representations must be used, the most widely studied examples being permutation problems such as the travelling salesrep problem (TSP). Although numerous operators for permutations have been developed—for examples see Goldberg & Lingle (1985), Oliver *et al.* (1987), Whitley *et al.* (1989), and Fox & McMahon (1991)—and a variation of schema analysis based on *o*-schemata has been developed (Goldberg, 1989), permutation problems seem largely to have been regarded as exceptional.

The comments above notwithstanding, the reasons for the continued dominance of linear binary representations bear examination. Principal among these is a simple counting argument introduced by Holland in his original book (Holland, 1975), giving rise to the notion of *intrinsic parallelism.* The familiar observation here is that every chromosome of length n is an instance (member) of 2^n schemata. If it is accepted that the genetic algorithm processes schemata, rather than individual chromosomes, which is a possible interpretation of the Schema Theorem (Holland, 1975) this suggests that representations which maximise 2^n will give rise to the greatest degree of intrinsic parallelism. They will thus—it is argued—achieve maximum processing efficiency. Since chromosome length n, and hence 2^n, is maximised for binary representations, there is a wide-spread belief that these will be most effective.

There are a number of counter-arguments to this.

1. *Representation Independence.*
   Let $\mathcal{S}$ be a search space of size $|\mathcal{S}| = 2^n$, and let $\mathbb{B}^n$ be the set of binary strings of length n ($\mathbb{B} = \{0, 1\}$). Then there are $2^n!$ possible faithful representations of $\mathcal{S}$, corresponding

to the $2^n!$ invertible mappings in

$$\mathcal{R} = \{\, \rho \mid \rho : \mathcal{S} \longrightarrow \mathbb{B}^n,\ \rho \text{ injective} \,\}. \tag{1}$$

It is both the strength and the weakness of the Schema Theorem that it is independent of which of these $2^n!$ representations is chosen. Yet if a representation ρ is selected randomly from $\mathcal{R}$, it can be expected to preserve no information, and thus the schemata will carry no significance. In such circumstances, the Schema Theorem notwithstanding, the genetic algorithm cannot reasonably be expected to make progress faster than a random search. This is reconciled with the Schema Theorem through the observation that the *observed* fitness of a schema ξ, given a randomly-selected representation ρ from $\mathcal{R}$, is an extremely poor estimator of the fitness of other members of ξ not present in the current population, since there are no expected correlations between the performance of the members of any schema. Schemata with high variance for fitness cannot, in general, be processed effectively by genetic algorithms.

The critical point to note from this discussion is not that the genetic search can do no better than random search given a random representation (no search strategy can given access only to such a representation) but rather that the Schema Theorem still holds in this case. The careless interpretation that this suggests that a genetic search will out-perform a random enumeration is incorrect in these circumstances.

2. *Inappropriate Schemata*
A search space of size 2^n has 2^{2^n} subsets. Using a base-k representation, only $(k+1)^n$ of these subsets are available for processing as schemata. In many cases, subsets which group together the chromosomal representatives of solutions which share properties that might be expected to influence their fitness, not only do not form schemata, but are not even contained in any schema except the most general one ($\Box\Box\cdots\Box$, where $\Box$ is the "don't care" symbol) (Radcliffe, 1990, 1991a). For example, coding the integer range 0 to 15 on four bits in the conventional manner, the representatives of 7 and 8 (0111_2 and 1000_2 respectively) share membership of no schema except $\Box\Box\Box\Box$. Similarly, multiples of three are grouped together by no (traditional binary) schema except $\Box\Box\Box\Box$.

3. *Not Only Schemata Obey the Schema Theorem*
Finally, the counting argument which gives rise to the notion of intrinsic parallelism suggests that binary representations maximise the degree of intrinsic parallelism only if attention is restricted to conventional schemata (Radcliffe, 1990). This restriction is inappropriate, as has been argued by Antonisse (1989), Radcliffe (1991a) and Vose (1991), and is detailed below in section 3.

2.3. Representations and Operators

Vose & Liepins (1991) have pointed out that the difference between the simplest problems for genetic search and those normally considered to be the hardest (fully deceptive problems, Goldberg, 1990a, 1990b) is no more than a change of representation. Vose (1991) has also introduced the notion of a "global" schema, which is one immune to sampling error by virtue of being fitter than average in an *arbitrary* population if it is fitter than average in any *specific* population.

It is clear that the fitness variance of schemata is central to the way in which a genetic algorithm proceeds, for it is the *observed* fitness of a schema which occurs in the Schema Theorem, stochastically determining which schemata are reproduced. The standard (though largely unspoken) approach is to try to choose representations in which schemata are meaningful, usually by trying to use genes which form intuitively sensible units of inheritance, in line with Goldberg's "principle of meaningful building blocks" (Goldberg, 1989).

An alternative approach to trying to find a representation in which the conventional operators and schemata are appropriate is to try to define operators which directly manipulate solutions in "sensible" ways. If this approach is to be taken, the key question becomes the definition of "sensible" manipulation of solutions and collections of solutions. Three approaches to this can be distinguished.

1. Countless workers have simply hand-designed *ad hoc* recombination operators which directly manipulate solutions. Obvious examples include, but are by no means restricted to, the various permutation operators developed for the TSP and job-shop scheduling. In some cases the principles used to design the operators have been made explicit. For example, the "cycle crossover operator" (Oliver *et al.*, 1987) ensures that every position in the child permutation is identical to the corresponding position in one or other parent. Whether operators are defined in the abstract or with respect to a particular (usually linear string) representation varies. An interesting example of defining operators directly in phenotype space is John Koza's *subtree-swap* recombination operator for manipulating the parse tree of lisp programs (Koza, 1990).

2. Lawrence Davis (1989) has pioneered and championed an approach which involves hand-generating several recombination (and other) operators, and applying them all with probabilities which are adapted by a credit assignment scheme related to "bucket brigade". This approach, which has produced impressive empirical results on a range of problems, allows a degree of automated operator selection, and "hybridisation" with domain-specific heuristics (Davis, 1991).

3. The final approach is to try to specify in the abstract what is meant by "sensible" properties for recombination and other operators. This approach has been taken independently, though formulated very differently, by Vose and Liepins (Vose, 1991, Vose & Liepins, 1991) and by Radcliffe (Radcliffe, 1990, 1991a, 1991b, 1992c). Their observations and principles for operator design have significant overlap, and form the basis for the rest of this paper.

3. GENERALISATIONS OF SCHEMA ANALYSIS

3.1. Generalised Schemata and Intrinsic Parallelism Redux

Antonisse (1989) argued that higher cardinality representations offer a higher degree of intrinsic parallelism than binary representations because the "don't care" character $\square$ should be interpreted not simply as a wildcard, but as the power set of the allele set for its position. The Schema Theorem still applies to these extended schemata, and the degree of intrinsic parallelism is higher than for binary representations (Antonisse, 1989).

Vose (1991) and Radcliffe (1991a) have gone further, independently demonstrating that the "Schema" Theorem actually applies to *arbitrary* subsets of the search space provided only that the disruption coefficients for the chosen operators are computed appropriately. Radcliffe terms these arbitrary subsets *formae,* whereas Vose calls them *predicates.* Given a recombination operator, Vose & Liepins (1991) tried to select the predicates (subsets of the representation space) which the operator could manipulate effectively, and formulated this in terms of a *lattice* of predicates. Radcliffe sought to use formae to capture information about *relevant* subsets of the search space, and then to constrain genetic operators to manipulate a *given* collection of formae effectively. Radcliffe's formulation interprets formae as equivalence classes induced by arbitrary equivalence relations over the search space.

3.2. Respect and Invariance

Perhaps the most basic condition to impose on recombination operators is that if both parents are members of some forma, then all of their children produced by recombination should also be members of that forma. Radcliffe (1991a) says that a recombination operator which ensures this *respects* a given set of formae. Equivalently, Vose (1991) says that a lattice of predicates for which this holds true is *invariant* under the given recombination operator.

For example, suppose that eye colour is deemed to be an important determinant of fitness for humans. Eye colour is an equivalence relation which partitions people into equivalence classes (formae) labelled green, blue, brown and so forth. A recombination operator which respects eye colour would ensure that all children born of two blue-eyed parents had blue eyes. The idea is to specify a number of equivalence relations, which between them completely specify each possible solution, and to ensure that they are simultaneously respected.

Respect (or equivalently, invariance) helps to ensure that once found, useful information is retained and exploited. All of the standard crossover operators for linear strings (n-point and uniform crossovers) respect schemata.

3.3. Assortment, Separability and Closure

In addition to respect/invariance, it is desirable to ensure that useful recombination is possible so that the properties characterised by the formae/predicates which are present in parents can be mixed. Radcliffe (1991a) formulates this by saying that a recombination operator *assorts* a set of formae if, given two parent solutions, one of which is a member of some forma ξ_1, and the other of which is a member of some second forma ξ_2, the recombination operator is capable of generating a child in the intersection $\xi_1 \cap \xi_2$, provided that this is non-empty. The assortment is said to be *proper* if the child can always be generated with a single recombination, and *weak* if repeated applications may be required (crossing one parent with its partner, and then with some number of generations of descendents).

For example, suppose that eye and hair colour have been chosen as equivalence relations to induce formae. Then if one parent has blue eyes and the other has brown hair, and providing that these are not incompatible characteristics, an assorting recombination operator must be able to generate a child which has both blue eyes and brown hair. Conventional crossover operators for linear chromosomes assort schemata. Uniform crossover assorts them properly, but n-point crossover for any *fixed* n assorts them only weakly.

Whereas respect contributes to exploitation of information already collected about formae, proper assortment contributes to thorough exploration by ensuring that all possible combinations of formae already found can be generated. (Mutation also plays an important rôle in exploration.)

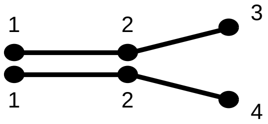

Figure 1. The upper tour fragment from a TSP is a member of the forma described by $\{23\}$, and which contains all tours which include the 2–3 or the 3–2 edge. Similarly, the lower tour fragment is a member of the forma described by $\{24\}$. Both tours are also a member of the forma described by $\{12\}$. Thus, if these formae are to be respected, all children of two tours containing these fragments must contain the 1–2 edge. This prevents a child being produced which is a member of the forma described by $\{23, 24\}$, the intersection of those described by $\{23\}$ and $\{24\}$, as required by proper assortment. Thus respect and assortment cannot both be satisfied, so these formae are non-separable (Radcliffe, 1991b).

It is important to note that respect and assortment are not necessarily compatible conditions. A set of formae which can be simultaneously respected and assorted is said to be *separable*, and a recombination operator which respects and properly assorts them is said to *separate* the formae. Schemata are clearly separable since uniform crossover separates them. An example of non-separable formae for the TSP is given in Radcliffe (1991b), and summarised in figure 1. In Radcliffe (1991b) a class of *random respectful recombination* operators (R^3) was introduced which automatically separate any set of separable formae.

Vose & Liepins (1991) have a related but different way of trying to ensure that effective recombination can occur, and here the difference of emphasis between fixing the operator and asking which predicates are manipulated by them and fixing the formae and trying to impose conditions on the operator comes to the fore. They introduce a notion of a *closed* lattice of predicates, which is essentially one in which it is always possible to make progress from large predicates (cf. low order schemata) to small predicates (cf. high order schemata) without violating invariance.

4. PROBLEM I: SEPARABILITY

The discussion above shows that the first difficulty with linear chromosomes and conventional genetic operators is that there are problems for which conventional schemata cannot capture the desired characteristics. This is because schemata are separable (can be simultaneously respected and assorted) whereas some quite reasonable sets of formae, such as those for the TSP discussed in figure 1, are not separable.

When faced with such a problem, a decision has to be taken as to how best to proceed given the lack of separability of the desired formae. It should be apparent that the non-separability of a set of formae is a fundamental difficulty if the formae are believed to be the appropriate ones for capturing the performance regularities in the search space and it is accepted that respect and assortment contribute to effective genetic search. Defining an operator which either respects or assorts an arbitrary set of formae is trivial (Radcliffe, 1990). Radcliffe (1992c) has also shown that a set of parameterised operators can be introduced which assort arbitrary formae and for which the degree of violation of respect can be controlled. In essence, the greater the degree of violation of respect that is permitted, the more “thorough” is the assortment. Clearly these options do not exist when using conventional linear string representations and operators which manipulate schemata.

5. PROBLEM II: LINEAR CHROMOSOMES AND ORTHOGONALITY

Although respect and assortment are not always compatible, they are not particularly strong conditions. Their most noticeable weakness is that while respect specifies that children must inherit any shared characteristics of their parents (when these can be expressed as common forma membership), it says nothing about which characteristics a child should have when its parents differ. Continuing with the example used earlier, it is desirable that if one parent has blue eyes and the other has brown eyes then recombination should produce only children with blue or brown eyes. To impose this, it is necessary to formalise the notion of a gene in the context of arbitrary formae.

A set Ψ of equivalence relations over a search space $\mathcal{S}$ will be said to *cover* $\mathcal{S}$ if the two members of every pair of solutions lie in different equivalence classes (formae) for at least one relation in Ψ. Such a set of relations can be used to generate a representation for $\mathcal{S}$. An algebraic structure can be imposed on Ψ by defining the intersection of a set of equivalence relations as their logical conjunction, so that two solutions are equivalent under a pair of equivalence relations only if they are equivalent under each of the pair. Given this structure, two kinds of *basis*—orthogonal and independent—can be introduced. A subset E of Ψ will be said to *span* Ψ if all of the members of Ψ can be constructed by intersection of the members of E. A spanning set E is then said to constitute an *orthogonal basis* for Ψ (Radcliffe, 1991b) if given any choice of equivalence class for each of the relations in E, a solution in $\mathcal{S}$ exists in their mutual intersection. E is said to constitute an *independent basis* for Ψ (Radcliffe, 1992b) if none of the members of E can be constructed by intersecting other members of Ψ. Orthogonality is the stronger condition, and implies independence (Radcliffe, 1992a).

The equivalence relations in an orthogonal basis E for a set Ψ of equivalence relations which covers $\mathcal{S}$ are called *basic equivalence relations* and play the rôle of genes, while the *basic formae* which they induce play the part of alleles. (Basic formae are analogous to first-order schemata, which are isomorphic to alleles.) In this case, uniform crossover acting on the representation induced by E separates the formae, which are in this case isomorphic to conventional schemata. In the more general case, however, only a (non-orthogonal) independent basis can be found for Ψ (Radcliffe, 1992a). Given only such a basis, the formae induced by the relations in Ψ may or may not prove to be separable. The basic equivalence relations and basic formae will still play the respective rôles of genes and alleles, but now not all combinations of alleles will be legal, and it is in this sense that the combination of conventional linear chromosomes and conventional operators can be seen to be inadequate. If the formae *are* separable, the *inheritance crossover operator* introduced in Radcliffe (1992b) will separate them and ensure that all genes in the children are taken from one or other parent; if not, it becomes necessary to sacrifice respect (and strict gene transmission); the thoroughness of assortment attainable will then depend on the degree of violation of respect which will be tolerated. The structure of the space of equivalence relations is shown in figure 2.

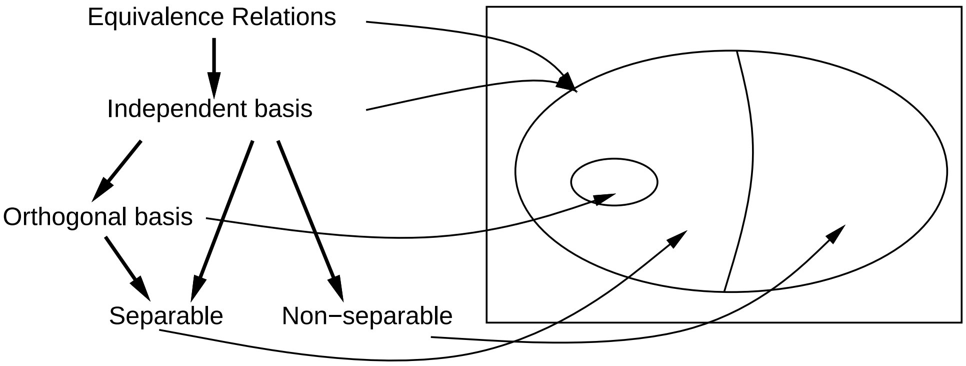

Figure 2. The structure of the space of equivalence relations.

6. PROBLEM III: LINEAR CHROMOSOMES AND CONSTRAINTS

The final problem with linear chromosomes and conventional crossover operators which will be addressed in this paper concerns general constraints. Constraints are normally considered to be highly problematical for genetic algorithms, and the "standard" approach is to use penalty functions to discourage their violation (e.g. Richardson *et al.*, 1989). These approaches have not, however, been conspicuously successful. Other workers, including Davis & Steenstrup (1987) and Michalewicz & Janikow (1991), have addressed this issue and argued for the building of operators which "understand" the constraints and are guaranteed not to violate them by virtue of this understanding.

In this context, it is important to distinguish between "real" constraints, such as linear or polynomial constraints on sets of parameter values, and those which are artifacts of the representation employed. For example, the TSP is sometimes viewed as being "constrained" to have exactly one copy of each city on the chromosome, whereas the more natural interpretation is that the TSP is an *unconstrained* search problem in which the search space is the set of permutations of the city labels. Similarly, if a binary representation is used for a range 0–10, this is sometimes viewed as a constraint that the values 1011_2 to 1111_2 are illegal, whereas in reality this "constraint" is a mere artifact of the representation used.

In both cases, however, the problem is made significantly easier simply by defining operators in the real search space and then merely producing *implementations* of the designated operators for the particular representation chosen.

7. SUMMARY

Theoretical developments from other papers have been collected together and used to argue that the combination of conventional linear chromosomes and standard recombination operators is inadequate for general problems. Three specific classes of problems for which the conventional paradigm fails have been discussed, covering cases in which schemata are unable to describe important subsets of the search space, those in which key characteristics of solutions cannot be independently assigned, and those in which constraints are involved. In all cases potential resolutions have been identified which rely on defining operators directly in the search space (rather than in the representation space) with the aid of high level constructs called *formae*, which are generalisations of schemata.

References

Antonisse, 1989. Jim Antonisse. A new interpretation of schema notation that overturns the binary coding constraint. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1989.

Bäck *et al.*, 1991. Thomas Bäck, Frank Hoffmeister, and Hans-Paul Schwefel. A survey of evolution strategies. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1991.

Booker, 1987. Lashon Booker. Improving search in genetic algorithms. In Lawrence Davis, editor, *Genetic Algorithms and Simulated Annealing*. Pitman (London), 1987.

Caruana and Schaffer, 1988. Richard A. Caruana 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.

Davis and Steenstrup, 1987. Lawrence Davis and Martha Steenstrup. Genetic algorithms and simulated annealing: An overview. In Lawrence Davis, editor, *Genetic Algorithms and Simulated Annealing*. Pitman (London), 1987.

Davis, 1989. Lawrence Davis. Adapting operator probabilities in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1989.

Davis, 1991. Lawrence Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold (New York), 1991.

Eshelman *et al.*, 1989. Larry J. Eshelman, Richard A. Caruana, and J. David Schaffer. Biases in the crossover landscape. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1989.

Fox and McMahon, 1991. B. R. Fox and M. B. McMahon. Genetic operators for sequencing problems. In Gregory J. E. Rawlins, editor, *Foundations of Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1991.

Goldberg and Lingle Jr, 1985. David E. Goldberg and Robert Lingle Jr. Alleles, loci and the traveling salesman problem. In *Proceedings of an International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates (Hillsdale), 1985.

Goldberg, 1989. David E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley (Reading, Mass), 1989.

Goldberg, 1990a. David E. Goldberg. Genetic algorithms and walsh functions: Part I, a gentle introduction. *Complex Systems*, 3:129–152, 1990.

Goldberg, 1990b. David E. Goldberg. Genetic algorithms and walsh functions: Part II, deception and its analysis. *Complex Systems*, 3:153–171, 1990.

Goldberg, 1990c. David E. Goldberg. Real-coded genetic algorithms, virtual alphabets, and blocking. Technical Report IlliGAL Report No. 90001, Department of General Engineering, University of Illinois at Urbana-Champaign, 1990.

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

Koza, 1990. John R. Koza. Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Stanford University, 1990.

Michalewicz and Janikow, 1991. Zbigniew Michalewicz and Cezary Z. Janikow. Handling constraints in genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1991.

Oliver *et al.*, 1987. I. M. Oliver, D. J. Smith, and J. R. C. Holland. A study of permutation crossover operators on the travelling salesman problem. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1987.

Radcliffe, 1990. Nicholas J. Radcliffe. *Genetic Neural Networks on MIMD Computers*. PhD thesis, University of Edinburgh, 1990.

Radcliffe, 1991a. Nicholas J. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5(2):183–205, 1991.

Radcliffe, 1991b. Nicholas J. Radcliffe. Forma analysis and random respectful recombination. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 222–229. Morgan Kaufmann (San Mateo), 1991.

Radcliffe, 1992a. Nicholas J. Radcliffe. The algebra of genetic algorithms. Technical Report EPCC–TR92–11, in preparation, Edinburgh Parallel Computing Centre, University of Edinburgh, 1992.

Radcliffe, 1992b. Nicholas J. Radcliffe. Genetic set recombination. In Darrell Whitley, editor, *Foundations of Genetic Algorithms II (to appear)*. Morgan Kaufmann (San Mateo, CA), 1992.

Radcliffe, 1992c. Nicholas J. Radcliffe. Genetic set recombination and its application to neural network topology optimisation. *Neural Computing and Applications*, 1(1), 1992.

Richardson *et al.*, 1989. John T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some guidelines for genetic algorithms with penalty functions. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–195. Morgan Kaufmann (San Mateo), 1989.

Spears and De Jong, 1991. William M. Spears and Kenneth A. De Jong. On the virtues of parameterised uniform crossover. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 230–236. Morgan Kaufmann (San Mateo), 1991.

Syswerda, 1989. Gilbert Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo), 1989.

Vose and Liepins, 1991. Michael D. Vose and Gunar E. Liepins. Schema disruption. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 237–243. Morgan Kaufmann (San Mateo), 1991.

Vose, 1991. Michael D. Vose. Generalizing the notion of schema in genetic algorithms. *Artificial Intelligence*, 1991.

Whitley *et al.*, 1989. 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.