Appears in "Foundations of Genetic Algorithms III", Ed: L.D. Whitley, M.D. Vose, Morgan Kaufmann (San Mateo), pp51-72, 1994.

# Fitness Variance of Formae and Performance Prediction

**Nicholas J. Radcliffe**
njr@epcc.ed.ac.uk
Edinburgh Parallel Computing Centre
University of Edinburgh
King's Buildings
EH9 3JZ
Scotland

**Patrick D. Surry**
pds@epcc.ed.ac.uk
Edinburgh Parallel Computing Centre
University of Edinburgh
King's Buildings
EH9 3JZ
Scotland

## Abstract

Representation is widely recognised as a key determinant of performance in evolutionary computation. The development of families of representation-independent operators allows the formulation of formal representation-independent evolutionary algorithms. These formal algorithms can be instantiated for particular search problems by selecting a suitable representation. The performance of different representations, in the context of any given formal representation-independent algorithm, can then be measured. Simple analyses suggest that fitness variance of formae (generalised schemata) for the chosen representation might act as a performance predictor for evolutionary algorithms. This hypothesis is tested and supported through studies of four different representations for the travelling sales-rep problem (TSP) in the context of both formal representation-independent genetic algorithms and corresponding memetic algorithms.

## 1 Motivation

The subject of this paper is representation in the context of evolutionary search. In particular, it explores questions such as what it might mean to talk about the quality of a representation, what the interaction between operators and representations might be, how one set of operators might be said to be better in some problem domain than some other set of (possibly "standard") operators, and—most importantly—what are the characteristics of representations that make evolutionary search easier or harder. The particular problem domain on which the paper will focus will be the travelling sales-rep problem (TSP), and results will be presented for preliminary empirical studies of this problem. This should not, however, distract from the primary goal, which will be to understand more about the nature of representations in evolutionary search.

This work grows out of previous work on *forma analysis* (Radcliffe, 1992, 1994), which is a generalisation of schema analysis (Holland, 1975). Forma analysis as developed thus far allows the generation of a (formal) "genetic" representation through the characterisation of a problem class. Significant effort has been devoted to understanding the kinds of representations that can result from arbitrary characterisation of general problem classes (though the discussions have tended to be couched in terms of "search spaces" rather than "problem classes") and a family of generic recombination operators has been developed. Each of the recombination operators in this family can be applied to any search problem given only a (sufficiently rich) characterisation of the problem class. This paper, together with Radcliffe & Surry (1994), extends this family of recombination operators and augments it with representation-independent forms of mutation and hill-climbing algorithms.

The result of having created representation-independent forms for all the classes of operators in common use in evolutionary computing is that formal, representation-independent algorithms can be defined. This permits variation of representation as an independent variable in the context of a fixed formal algorithm, allowing the influence of representation to be isolated and measured. This is achieved by comparing the performance of algorithms that are identical in all respects other than the representation chosen—ones that execute the same reproductive plan, with the same set of operators and the same parameters. This paper addresses the exploitation of evolutionary algorithms for search and optimisation, and is thus, in the terminology favoured by DeJong (1992), concerned with GAFO's (Genetic Algorithms for Function Optimisation) or perhaps ECFO's (Evolutionary Computing For Optimisation). In particular, it is formal genetic and memetic algorithms that will be considered, memetic algorithms being genetic algorithms that include local optimisers as operators (Moscato & Norman, 1992; Radcliffe & Surry, 1994).

There is considerable freedom in choosing exactly how to measure the performance of evolutionary algorithms. For example, it would be possible to choose to consider rate of convergence, time to solution, or robustness of results, off-line or "best seen" performance, number of evaluations or wall-clock time in almost any combination. Moreover, robustness is an issue not only with respect to differently seeded stochastic runs, but also as a function of different problem instances, perhaps of quite different complexities, and it may be that different relative performances will be achieved by different representations as the parameters of the algorithm, the operators used and the particular problem instance are varied. Despite this panoply of choices, it seems reasonable to expect that for at least some classes of problem there will be a broad congruence of results over these different measures. For other problem classes it might be possible to rank the performance of different representations with respect to a particular performance measure and some chosen set of parameters and operators.

The principal aim of this paper is to find and measure properties of representations that are well-correlated with their performance in evolutionary search. If this could be achieved it would both allow some level of performance prediction and increase understanding of the search techniques themselves. A key motivation for the development of forma analysis was a set of observations about the schema theorem (Holland, 1975) in its generalised form (Radcliffe, 1991; Vose & Liepins, 1991). These suggest that representations based on formae (generalised schemata) with lower fitness variance would be expected—other things being equal—to allow more effective search than those based on formae with higher fitness variance. Fitness variance of formae thus forms a natural candidate for a measure to act as a predictor of representation performance. This paper goes on to explore whether or not this proposition is supported through studies of the TSP.

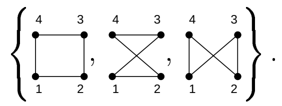
Hofmann (1993) began to test fitness variance of formae as a performance predictor by studying the travelling sales-rep Problem (TSP). He compared instantiations of random assorting recombination (RAR; Radcliffe, 1994) using various representations, and the *strategic edge crossover*, SEX, developed by Moscato & Norman (1992) as an extension of edge recombination (Whitley *et al.*, 1989). The present work develops these tests significantly further by considering fully representation-independent genetic and memetic algorithms with a range of representations.

## 2 A Review of Forma Analysis

Forma analysis provides methods for constructing representations of problems and for defining operators with respect to those representations. Radcliffe (1994) provides a detailed and precise formulation of forma analysis, while Radcliffe (1992) gives a more descriptive overview. The present section summarises only those parts necessary for the immediate goals of formulating and analysing representation-independent evolutionary algorithms.

### 2.1 Problem Class and Search Space

Let $\mathcal{T}$ be a class of search problems. In the present paper $\mathcal{T}$ will be the set of all travelling sales-rep problems. More specifically, let $\mathcal{T}$ be the set of all problem instances in the class, where a problem instance takes the form of the search space, $\mathcal{S}$, to which it gives rise. Given a particular collection of cities, the search space $\mathcal{S}$ is the set of all possible paths that visit every city exactly once—the set of all possible tours—and the aim is to find the shortest with respect to some metric. More precisely still, since there is no interest in the $2n$ equivalent paths through $n$ cities that arise from the freedom to choose the starting city and the direction of travel, $\mathcal{S}$ is the set of all *non-equivalent* tours. For example, given a set of four particular cities labelled 1 to 4, chosen for illustration to sit at the corners of a square,

$\mathcal{S} =$  (1)

A convenient way to represent a tour is by listing the sequence of city labels in the order in which they are visited, so that the first tour shown in the set might be represented by 1234. This is called the *permutation representation*, denoted $p$.

It is important to be able to distinguish between the representative of a solution under some representation (the formal chromosome, or genotype) and the solution itself (the phenotype). For a particular representation $\rho$, the set of all chromosomes in this representation will be denoted $\mathcal{C}^\rho$. Given a chromosome $X \in \mathcal{C}^\rho$, and the solution $x \in \mathcal{S}$ to which it corresponds, the notation $X^\rho$ will be used to mean “the solution represented by $X$ in representation $\rho$”. Thus $X^\rho$ is a member of $\mathcal{S}$, and in the current example $X^\rho = x$. It is thus accurate to write

$$\mathcal{S} = \{1234^p, 1243^p, 1324^p\}\,. \qquad (2)$$

In general, a TSP over $n$ cities has an associated search space of size $n!/2n = (n-1)!/2$, arising from the $n!$ different permutations of the city labels and the $2n$ equivalent forms for any tour. In this paper, tours shown in the permutation representation will be shown starting with city 1, followed by the lower-numbered of city 1’s two neighbours.

### 2.2 Representation

It is necessary to be rather precise about representations in this study. For present purposes, a distinction will be made between *genetic* representations and *allelic* representations. A genetic representation includes a collection of *genes,* each of which has a well-defined value for every chromosome in $\mathcal{C}^\rho$. As usual, the values that genes take will be termed *alleles,* and will formally be labelled with the gene to which they correspond. It is a requirement of a (formal) genetic representation that given all the gene values it be possible to identify the unique solution to which they collectively correspond (if they do correspond to some solution). If all combinations of alleles correspond to solutions in $\mathcal{S}$ the genes (and the representation) are said to be *orthogonal.* It is *not* a requirement that representations be orthogonal, and indeed all of the representations that will be used in this paper are non-orthogonal. The permutation representation of the TSP used in equation 2 qualifies as a formal genetic representation under this definition. For an $n$-city TSP, is has $n$ genes, $(g_1, g_2 \ldots, g_n)$, the $i$th of which describes the city that is visited $i$th in the tour. Notice that not all combinations of alleles correspond to solutions—for example, 1223 does not—so the representation is non-orthogonal.

In an *allelic* representation, genes are *not* required. Instead, a set of properties is defined, each of which a solution may or may not have, and a solution is represented by the set of properties it has. For example, in the *undirected edge representation* of the TSP, denoted $u$, a solution is represented by the set of the edges that it contains. Thus, the tour $1234^p$ contains the (undirected) edges 12, 23, 34 and 14, so

$$1234^p = \{12, 23, 34, 14\}^u. \tag{3}$$

Each of the chosen properties will be referred to as an *allele* and it is a requirement of an allelic representation that every collection of alleles corresponds to a unique solution (if it does correspond to some solution). This use of the term allele is non-standard, and arguably an abuse, but is natural and convenient for the purposes of this paper.

Having introduced both (formal) genetic representations and (formal) allelic representations the notion of a *chromosome* (or genotype, or genome) can be made precise. In the case of genetic representations this is more-or-less the familiar object, though formally will consist simply of the set of alleles characterising a solution, usually together with an ordering of the genes. Thus, using a tuple $(a, b)$ to specify that the $a$th city visited is the city with label $b$, the formal chromosome corresponding to $1243^p$ is $\{(1,1),(2,2),(3,4),(4,3)\}$. In the case of an allelic representation, again the chromosome corresponding to a solution is simply the set of alleles that the solution contains.

It should be mentioned briefly that when (formal) genetic and allelic representations are discussed in this paper there is no intended implication that the actual coding used to store individuals in a computer need take the form described. The formal representation or representations that an evolutionary computation uses are defined by the actions of operators on individuals rather than any details of implementation.

### 2.3 Formae

A *forma* is a generalisation of the familiar concept of a *schema.* A forma can be viewed for present purposes as any collection of solutions in which the corresponding chromosomes share particular alleles. Formae are typically given names such as $\xi$, and specified through a *description set* denoted $\langle\xi\rangle$. The description set is simply the set of alleles that a chromosome must have in order for the solution it represents to be a member of the forma in question, and is thus closely related to the set of defining positions (and defining values) familiar from schema analysis. Since the formal chromosome used here is precisely the set of alleles, it is clear that

$$\xi = \{X^\rho \in \mathcal{S} \mid X \supset \langle\xi\rangle\}. \tag{4}$$

For example, working in the permutation representation for a 4-city TSP, the forma containing all tours that have city 2 in their second position would be given by

$$\xi = \{1234^p, 1243^p\} \tag{5}$$

and described by

$$\langle \xi \rangle = \{(2,2)\}, \tag{6}$$

This is probably more familiar to most readers as the *o*-schema □2□□ (Goldberg & Lingle, 1985). Similarly, in the undirected edge representation the forma $\xi'$ consisting of those tours that contain the 12 edge is

$$\xi' = \{1234^p, 1243^p\} = \big\{\{12, 23, 34, 14\}^u, \{12, 24, 34, 13\}^u\big\}, \tag{7}$$

which is conveniently described by

$$\langle \xi' \rangle = \{12\}. \tag{8}$$

### 2.4 Recombination, Respect, Transmission and Assortment

Evolutionary computing has given rise to a large and growing collection of recombination operators. When the representations used are orthogonal, certain properties tend to be common to almost all such operators, and seem barely worthy of comment. When non-orthogonal representations are used, however, these properties are much less universal and become more salient. The three properties of respect, transmission and assortment are of particular relevance and are defined below. These are normally discussed with respect to formae, but are here discussed with reference instead to a representation.

A recombination operator is said to *respect* a representation if and only if every child it produces contains all the alleles common to its two parents.

A recombination operator is said to *transmit genes* with respect to a given (formal) *genetic* representation if and only if each allele in every child it produces is present in at least one of that child's parents. It is easy to see that a recombination operator that transmits *genes* is respectful.

A recombination operator is said to *transmit alleles* with respect to a given (formal) *allelic* representation if and only if each allele in every child it produces is present in at least one of its parents. It is *not* necessarily the case that an operator that transmits *alleles* for an allelic representation is respectful.

The distinction between gene transmission and allele transmission is important. For example, the Edge Recombination Operator in its original form (Whitley *et al.*, 1989) sought to transmit as many undirected edges from the two parents as possible, and was thus striving to achieve allele transmission, which it did typically with over 99% success. The operator was then modified to achieve strict respect by placing all edges common to the parents in the child at the start (Whitley *et al.*, 1991), resulting in a quite different (and apparently more successful) operator.

A recombination operator is said to be *assorting* (with respect to a given representation) if and only if it is possible for it to produce any solution that contains only alleles present in the two parents. It is not necessary for it to be possible to achieve this in a single recombination: repeated incestuous recombination may be required. In the latter case, the operator is said to be *weakly,* rather than *properly,* assorting.

The most familiar crossover operators from genetic algorithms (such as $N$-point crossover, reduced-surrogate crossover (Booker, 1987), parameterised uniform crossover (Spears & DeJong, 1991), shuffle crossover (Schaffer *et al.*, 1989) etc. are respectful, transmitting and assorting with respect to familiar string representations, which are typically orthogonal. However, for non-orthogonal representations assortment is often incompatible with respect and gene transmission. Many operators for non-orthogonal representations fail to have some or any of these properties.

## 3 Representation-independent Recombination

Three representation-independent recombination operators have previously been introduced, under the names of *random respectful recombination* ($R^3$), *random transmitting recombination* (RTR) and *random assorting recombination* (RAR). As the names suggest, $R^3$ is always respectful, RTR is always transmitting and RAR is always assorting, in each case with respect to the representation for which they are instantiated. For orthogonal representations, RAR and RTR are equivalent and thus assort, transmit and respect, and are in fact equivalent to uniform crossover. In the special case of binary representations, RTR reduces to $R^3$.

For all the representations of the TSP considered here, RAR is the most relevant operator of the family. It is described in detail in Radcliffe (1994), and may be thought of as a generalisation of uniform crossover. A newer representation-independent form of recombination is *Generalised $N$-point Crossover* (GNX), which is now described.

### 3.1 Generalised N-point Crossover

In constructing a generalised form of $N$-point crossover, it is convenient to consider only genetic representations. The difficulty in applying conventional crossover operators is that not all combinations of gene values are legal. Let $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_N\}$ be a set of cross points, with $0 < \ell_1 < \ell_2 < \cdots < \ell_N < n$. This breaks a parent (genetic) chromosome $X$ into the $N + 1$ segments

$$(X_1, X_2, \ldots, X_{\ell_1 - 1}),\ (X_{\ell_1}, X_{\ell_1+1}, \ldots, X_{\ell_2-1}),\ \ldots,\ (X_{\ell_N}, X_{\ell_N+1}, \ldots, X_n), \tag{9}$$

and breaks the second parent chromosome, $Y$, into corresponding segments.

The first phase of GNX's operation uses the same genetic material as ordinary $N$-point crossover, i.e., alternate segments from the two parents. It proceeds by picking a random order to visit the $N+1$ segments (irrespective of the parents to which these segments are assigned). Within each segment, the alleles are "tested" in a random order. An allele is "tested" by seeing whether it can be placed in the child—i.e. whether it is compatible with those alleles that have already been accepted. If compatible, the new allele is inserted, otherwise it is discarded. Because in general after this process has terminated the child will still be incomplete, a second phase then commences in which the genetic material discarded by ordinary $N$-point crossover (the 'complementary' alternating sections) is used to try to fill in any gaps. The segments are again visited in a random order and the alleles within them are tested in random sequence. If the child is still not fully specified after this, it is completed at random from amongst the legal combinations of alleles, or by some other patching method. In this study, patching is always random. The general pattern of progress of GNX is shown in figure 1.

An example using the TSP may help to clarify this. Consider the permutation representation for the TSP and G2X with cross points 3 and 6 with parents given by

$$\begin{aligned} X &= (1, 2, 3 \mid \underline{4}, \underline{5}, \underline{6} \mid 7, 8), \\ Y &= (\underline{1}, \underline{5}, \underline{4} \mid 3, 8, 7 \mid \underline{2}, \underline{6}), \end{aligned} \tag{10}$$

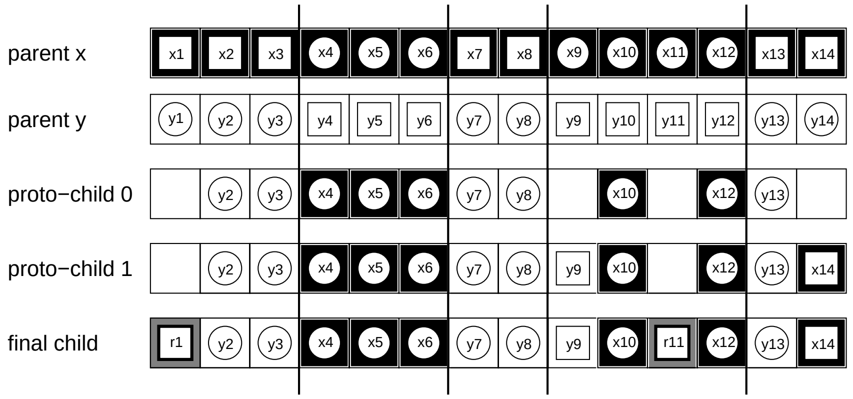


Figure 1: GNX first copies gene values from alternating segments (circled) of the parent chromosomes, visiting the segments and testing the genes within these segments in a random order. Gene values are copied to the child only if they are compatible with those already present. In this example, genes 2–8, 10, 12 and 13 are assigned in this way, resulting in proto-child 0. For genes not set by this process, alleles from the unused segments (boxed) of the parents are then tested for inclusion, again in random sequence. In the example, genes 9 and 14 are assigned thus, to give proto-child 1. Genes still not fixed after this process are assigned either at random from the set of legal combinations, or by some heuristic or other patching procedure. In this example, genes 1 and 11 fall into this category.

where the underlined alleles are the ones that would normally be chosen by $N$-point crossover. Suppose the order in which the segments are tested is $(2, 3, 1)$. Then the second segment of $X$ will be inserted whole, giving the proto-child $(\square, \square, \square, 4, 5, 6, \square, \square)$. Alleles in the third segment from $Y$ will then be tested in a random order. Whichever order is chosen in this case, the 2 will be accepted and the 6 rejected, giving the proto-child $(\square, \square, \square, 4, 5, 6, 2, \square))$. The first segment of $Y$ is then tested, and only the 1 will be accepted, giving the final proto-child, $Z_0$, at the end of the first phase as

$$Z_0 = (1, \square, \square, 4, 5, 6, 2, \square). \qquad (11)$$

The untested segments are then visited in random order, Only the first and third segments from $X$ are relevant here, and whatever the order of testing, the 3 and the 8 will be accepted, and the 2 and the 7 will be rejected, giving the proto-child at the end of the second phase

$$Z_1 = (1, \square, 3, 4, 5, 6, 2, 8). \qquad (12)$$

Since this child is still incomplete, it must be patched. In this case however, only one legal chromosome has the required allele pattern, so the final child is given by

$$Z = (1, 7, 3, 4, 5, 6, 2, 8). \qquad (13)$$

## 4 Representation-independent Mutation and Hill-climbing

No effort has previously been devoted—to the best of the authors' knowledge—to defining representation-independent mutation and hill-climbing operators. Since mutation is widely recognised as playing a vital rôle in evolutionary search it is clearly essential to the aim of this paper to include such an operator; given a move operator in the form of mutation, the construction of a hill-climber is a simple matter. The present section attempts to set the discussion of representation-independent mutation operators in a reasonably general context before specialising to construct one pertinent to the particular problem class—the TSP—considered.

A number of considerations affect the formulation of a generalised mutation operator, including the characteristics of mutation operators in normal use, the perceived function of mutation and the behaviour desired of a generalised mutation operator in special limits (such as the case of orthogonal representations). The different strands of evolutionary computing use rather different sorts of mutation operators. One nearly universal characteristic, however, is that they ensure ergodicity, i.e. that the entire search space remains accessible from any population, and indeed from any individual. In most cases mutation operators can actually move from any point in the search space to any other point directly, but the probability of making "large" moves is very much smaller than that of making "small" moves (at least with small mutation rates). For example, in evolution strategies mutation is typically Gaussian on a parameter-by-parameter basis, while in genetic algorithms with orthogonal representations a probability of (uniformly) choosing a new value for each gene is most commonly used, giving rise to a binomial distribution for the number of mutations made. It is a characteristic of most mutation schemes that relatively unlikely "large" moves can be effected by a series of smaller moves, thus making large, potentially fruitful moves possible with reasonable probability through iteration provided that the intermediate points (solutions) are themselves viable in the context of the reproductive plan used.

In order to clarify notions of "large" and "small" mutations, a metric (distance measure) will be introduced over the representation space $\mathcal{C}^{\rho}$. In the case of genetic representations, this is straightforward. A distinction is first made between *cardinal* and *ordinal* genes. The alleles for ordinal genes are naturally ordered, as, for example, when the gene represents a continuous or contiguous variable, and in this case it makes sense to define a variable distance between alleles (normally the Euclidean distance). With cardinal genes, such as those found in all the genetic representations of the TSP considered, there is no particular relationship between the various alleles for a given gene, so the discrete metric is used, making the distance between any pair of distinct alleles one. The distance between solutions is then computed simply by summing the distances between the gene values at each locus. For cardinal genes, this measure reduces to the familiar Hamming distance.

Allelic representations are a little more complex to handle. It is implicit in the definition of genes that every chromosome has the same number of alleles, because every gene has precisely one well-defined value for each solution (and there is no other source of alleles). In the case of allelic representations this need not be the case. For example, if the search space consists of sets of variable size, it would sometimes seem appropriate to take the (variable number of) elements of the set as alleles. A satisfactory approach for allelic representations is to define an overlap or similarity measure in the first instance by counting the number of alleles that are common to the two solutions in question. A distance is then formed by subtraction, possibly after normalisation, from a suitable number, being careful to ensure that a solution's distance from itself is zero. In the particular case of the TSP, the only allelic representation used in this paper (the undirected edge representation) has the property that every solution has the same number of alleles, so this complication may be ignored.

### 4.1 Binomial Minimal Mutation

The specific form of mutation required for present purposes is a generalisation of the standard gene-wise mutation operator from genetic algorithms since the TSP representations considered are all essentially cardinal. The *binomial minimal mutation* operator (BMM) can be defined for any representation—genetic or allelic—provided that the following conditions are met.

1. Every solution has the same number, $n$, of alleles. The distance between two chromosomes $X$ and $Y$ will be taken to be $n$ minus the number of alleles that they share and will be written $D(X, Y)$.
2. It is computationally feasible to identify *minimal mutations.* Considering a pair of chromosomes $X$ and $Y$, $Y$ will be said to be a minimal mutation of $X$ provided that there is no solution $Z$ ($\neq X$) for which $D(X, Z) < D(X, Y)$. Although not absolutely required, BMM is formulated primarily for the case in which the distance between a chromosome and each of its minimal mutations is the same for all chromosomes, and every chromosome has the same number of minimal mutations.
3. A sequence of minimal mutations from any point in the representation space $\mathcal{C}^\rho$ can generate any point in $\mathcal{C}^\rho$, and thus any point in $\mathcal{S}$.

*Binomial minimal mutation* takes a single parameter $p_m \in [0, 1]$, which is a probability. If the conditions listed above are met, BMM proceeds by first choosing a number of minimal mutations to make by sampling the binomial distribution $B(n, p_m)$, where, as before, $n$ is the number of alleles in each chromosome. It then generates the mutated child by repeatedly choosing a randomly (uniformly) selected minimal mutation until the requisite number have been performed.

There are a few points worth noting about this operator. First, in the limit of orthogonal genes, standard mutation is recovered provided that no mutation in the sequence is accepted if it generates a chromosome already visited in the sequence of minimal mutations. While this check is technically desirable, it is relatively troublesome to implement in general. For small values of $p_m$ and for representations in which the number of minimal mutations for each chromosome is large, the check makes little practical difference, merely distorting the binomial distribution slightly, so in the experiments for this paper, checks for previously visited mutations have not been performed.

Secondly, notice that there is no suggestion that the minimal mutations should have distance 1 from the solution to be mutated; indeed, while this will clearly be the case for orthogonal genetic representations, it is the case for none of the four representations considered in this paper.

Finally, given chromosomes $X$, $Y$ and $Z$, it is not necessarily the case that $D(X, Y) < D(X, Z)$ implies that $Y$ can be generated from $X$ in fewer minimal mutations than can $Z$.

### 4.2 Representation-independent Hill-climbing and Memetic Algorithms

Given a move operator, which in the present case will be taken to be the minimal mutation operator, it is a simple matter to define a family of hill-climbing operators. A hill-climber can be obtained by repeatedly trying moves generated by the move operator, in some sequence, and accepting all those that improve upon the current solution. This process continues until none of the moves that the operator can generate improves the solution. Numerous strategies are possible for generating the moves, but in the authors' experience the more random the order, the better will be the expected performance. The method used here generates moves in a pre-determined order for any particular hill-climbing, but this order changes each time the hill-climber is invoked. Davis (1991) has argued that it is usually desirable to accept neutral moves (ones that have no effect on fitness), but this is not central to the current discussion, and since neutral moves are unlikely in general TSP instances, this has not been allowed.

The application of a local optimiser such as the hill-climber described always succeeds in generating a local optimum (with respect to minimal mutations). Following Radcliffe & Surry (1994), which also contains further details of the hill-climbing techniques used, a *memetic algorithm* is defined to be a genetic algorithm in which local optimisation is applied to all solutions before evaluation. This can be thought of as a genetic algorithm applied in the subspace of local optima, with local optimisation acting as a repair mechanism for children lying outside this subspace (i.e. not being locally optimal). Because the evaluation function for the TSP is decomposable—the length of a child tour similar to its parent can be computed more easily given the length of the parent tour—hill-climbing is relatively cheap, so memetic algorithms might be expected to perform relatively better than genetic algorithms for this problem.

## 5 Formae and Performance Prediction

The primary aim in this work is to find measurable properties of representations that are correlated with the performance of an evolutionary algorithm. The candidate indicator studied will be fitness variance of the formae associated with the chosen representation. One motivation for this choice derives from the Schema Theorem (Holland, 1975), which has been shown both by Vose (1991) and Radcliffe (1991) to apply to general subsets of the representation space provided that disruption coefficients are computed appropriately. This motivation has previously been described in detail, so here the focus is instead on the interactions between formae, genetic operators and population update mechanisms.

Consider first those formae whose members in the current population are fitter than average. Along with the chromosomes that instantiate them, these formae are selected for reproduction more often, or selected for deletion less often, than formae with lower observed performance. Consider now the effects of applying genetic operators. If the recombination operator used respects the representation, this will ensure that whenever two parents are recombined their child will instantiate all formae of which they share membership. If the recombination operator is not very disruptive, even when only a single parent is a member of such formae the probability of generating new instances of them is relatively high. If the recombination operator is non-respectful, the extent to which these arguments apply depends on the degree to which respect is violated. Similarly, for (typical) low mutation rates, a child produced by mutation will share membership of most formae of low to medium order with its parent. Finally, although hill-climbing from a random starting point will usually produce a chromosome very different from its parent, when the parent is in the vicinity of a local optimum (with respect to that hill-climbing operator) as will tend to be the case after recombination in an effective memetic algorithm, it is likely that hill-climbing will not change a very large number of alleles, so forma membership is still to some extent preserved.

Turning now to forma construction—i.e., the sampling of chromosomes in formae of which the parents are not instances—observe that such newly sampled formae are likely to overlap in their allele composition with formae currently seen to be performing well, particularly to the extent that the recombination operator used is respectful. This is important because the way in which evolutionary (and particularly genetic) search is thought to proceed is by sampling smaller (higher order) formae that are intersections (compositions) of larger (lower order) formae of high relative fitness. Thus recombination is thought of as gradually building up complex solutions by combining "fit" components. While challenging search problems will exhibit some possibly large degree of non-linearity, so that combining components that are observed to be (relatively) fit will not always result in ever-fitter individuals, this approach retains some validity.

If the recombination operator used is transmitting, it can further be observed that alleles common to chromosomes and formae exhibiting above average performance at the current time step are also preferentially selected, both individually and collectively. The essence of the notion of assortment is that assorting recombination operators are capable of bringing together any compatible (non-competing) formae from the parents in a child. This seems to be essential given the model of genetic (and more widely evolutionary) search considered, the wide-spread use of non-assorting recombination operators for non-orthogonal representations notwithstanding. Needless to say, in the presence of selective pressure towards better solutions and formae, the new formae of which instances are created tend to be intersections of fitter, larger (lower order) formae.

Though somewhat imprecise, these arguments suggest that there are important senses in which evolutionary algorithms are directed by observed forma fitnesses, especially when recombination plays a major rôle, and when the recombination operators are (more strongly) respectful, transmitting and assorting. The schema theorem also points to the important rôle observed forma fitnesses play in guiding the search. Given this, the distribution of fitnesses within formae would seem to be central to how search proceeds. It is therefore worth considering the distributions that might be expected or desired.

First, it seems clear that for any challenging search problem large formae will have wide distributions of fitnesses, while if the search is to be guided by forma fitnesses it would be helpful if smaller formae had narrower distributions. Indeed, if formae were simply random collections of solutions from the search space it would be impossible to collect any useful directional information from collecting fitness samples from formae. Ultimately, of course, the interest is really in the fitness of the fittest member of each sampled forma, but unfortunately that is unavailable without exhaustively searching each forma. It is therefore probably necessary to restrict consideration to statistical measures available from sampling formae. Of these, the variance is the most familiar measure of spread, and it is this that will be considered.

These considerations suggest that the variance and other moments of formae from any representation in which the genes carry useful fitness information would be a maximum for large formae and fall towards zero for very small formae. More importantly, however, it might be further expected that representations for which fitness variance tends to fall more quickly as a function of increasing forma order would give evolutionary search algorithms stronger and more exploitable information than those with broader fitness distributions. For this reason the experiments in this paper will measure fitness variance and various other fitness moments for formae in four different representations for the TSP in the hope that these moments will tend to be inversely correlated with the performance of the algorithm. Related experiments have been conducted previously by Hofmann (1993), though he did not use a fully representation-independent algorithm and was more concerned with the comparison between RAR and a domain-specific operator, the Strategic Edge Crossover operator (SEX; Moscato & Norman, 1992), which is a variant of the Edge Recombination operator (Whitley *et al.*, 1989).

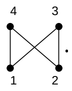
## 6 Representations for the TSP

In the empirical tests that follow, four representations of the TSP will be used. Two have already been introduced—the permutation representation $p$ (section 6.1), which gives rise to traditional *o*-schemata and is a (formal) genetic representation, and the undirected edge representation $u$ (section 6.2), which is allelic. Additionally, a directed edge (genetic) representation $d$ will be used (section 6.3) as well as a genetic variation $c$ of the undirected edge representation (section 6.4), that uses compound genes and is equivalent to the corner representation suggested by Hofmann (1993). These four representations are then characterised in sections 6.5–6.7.

## 6.1 The Permutation Representation

The permutation representation is a formal genetic representation, and has already been introduced in section 2.2. Here the $i$th gene identifies the $i$th city in the tour. Thus

$$1324^p = \qquad . \tag{14}$$



The first gene is fixed to have allele 1 and is thus formally redundant but this will not prove problematical. A complication does arise, however, from the freedom to choose the direction in which a tour is traversed. While this degeneracy can be formally removed with the convention introduced previously (always choosing the second city to have a lower numbered label than the $n$th), this is not wholly satisfactory. This problem is discussed in section 6.6.

The RAR$^p$ and GNX$^p$ operators are easy to implement for this representation, while the definition of BMM$^p$ depends on the observation that the minimal mutation of a permutation is generated by exchanging the positions of two cities. An example of such an exchange would be

$$143526^p \overset{\text{BMM}^p}{\longmapsto} 153426^p, \tag{15}$$

by exchange of genes 2 and 4. Minimal mutations are at distance 2 from their parents in this representation. (In terms of the number of edges broken, this corresponds to a four-change.) It is easy to see that all permutations can be generated by a sequence of such city exchanges, so the conditions for BMM are satisfied.

Gene transmission is incompatible with assortment in this representation, as the reader will be able to verify by trying to construct a member of the forma described by $\{(3,3),(5,2)\}$ from parents $123456^p$ and $134526^p$ without violating transmission.

## 6.2 The Undirected Edge Representation

The undirected edge representation has also been introduced previously and is allelic. In this representation a tour is represented by the set of undirected edges it contains. As an example of the representation,

$$1324^p = \{13, 23, 24, 14\}^u. \tag{16}$$

RAR$^u$, while somewhat harder to construct than for the permutation representation, is straightforward, and the factor of 2 that is problematical for the permutation representation does not arise since the representation makes no reference to the direction in which a tour is traversed.

It is easy to see that the minimal mutation for this representation is the reversal of a sub-tour, which is a 2-change in terms of the number of edges broken. For example,

$$\{14, 34, 35, 25, 26, 16\}^u = 143526^p \overset{\text{BMM}^u}{\longmapsto} 125346^p = \{12, 34, 35, 25, 46, 16\}^u, \tag{17}$$

reversing the section from positions 2 to 5 inclusive. (In general, solutions are written in the permutation representation in this paper, even when they are being manipulated with respect to other representations.) Minimal mutations are at distance 2 from their parents in this representation. Although this operator is often referred to as *inversion*, this term will be avoided here because of possible confusion with the general re-linking operator (Holland, 1975).

To see that transmission is incompatible with assortment in this representation, observe that generating an instance of the forma described by $\{24, 23\}$ from parents $123456^p$ and $124356^p$ is impossible without violating transmission.

It is possible to apply GNX$^u$ to the undirected edge representation by constructing an associated *pseudo-genetic representation.* This is achieved by arranging the edges in the order and sense in which they are encountered following the tour they describe in an arbitrarily chosen direction. The resulting list of edges will have every city exactly once as the "first" end of an edge, and can thus be used for alignment, allowing application of GNX$^u$. While the chromosome resulting from this process is the same as that used in the directed edge representation, the edge is still considered to be undirected, so that when determining whether an edge can be added to a proto-child, its sense may be reversed if necessary. Thus, GNX$^u$ based on undirected edges is different from the the same operator for directed edges.

## 6.3 The Directed Edge Representation

The directed edge representation is a (formal) genetic representation and represents a tour by the set of directed edges it contains. It is genetic because genes are well defined, the $i$th corresponding to the city visited after city $i$. Thus, identifying this representation by the superscript $d$,

$$1423^p = \{\overrightarrow{14}, \overrightarrow{23}, \overrightarrow{31}, \overrightarrow{42}\}^d. \tag{18}$$

As with the permutation representation, the degeneracy arising from the freedom to choose the direction in which to traverse the tour may be formally removed by convention, but again this is not wholly satisfactory. This is discussed in section 6.6.

The construction of RAR$^d$ and GNX$^d$ for this representation is again straightforward. The minimal mutations for the directed edge representation are 3-changes, for example,

$$143526^p \overset{\text{BMM}^d}{\longmapsto} 142356^p, \tag{19}$$

which corresponds to cycling the edges $\{\overrightarrow{43}, \overrightarrow{52}, \overrightarrow{26}\}$ to become $\{\overrightarrow{42}, \overrightarrow{56}, \overrightarrow{23}\}$. Minimal mutations are at distance 3 in this representation. The reader will quickly become convinced that such 3-changes suffice to generate all tours, and that 1-changes and 2-changes are not possible in this representation.

Attempting to construct an instance of the forma described by $\{\overrightarrow{23}, \overrightarrow{31}\}$ from parents $123456^p$ and $126543^p$ should convince the reader that transmission is incompatible with assortment here also.

## 6.4 The Corner Representation

The final representation is in some ways the most complex, and is again based on undirected edges, but forms a (formal) genetic representation, denoted $c$. In this case, there is a gene corresponding to each city, each of which takes compound alleles consisting of the unordered pair of edges centred on that city. For example,

$$1423^p = \Big\{\big(1,\{3,4\}\big), \big(2,\{3,4\}\big), \big(3,\{1,2\}\big), \big(4,\{1,2\}\big)\Big\}^c \tag{20}$$

where a tuple $(a, \{b, c\})$ indicates that city $a$ has neighbours $b$ and $c$. Defining RAR$^c$ and GNX$^c$ is straightforward.

The minimal mutation is sub-tour reversal (a 2-change), so the same example as for undirected edges may be used, specifically

$$143526^p \overset{\text{BMM}^c}{\longmapsto} 125346^p. \tag{21}$$

Notice, however, that in this case the new solution is at distance $4$ from its parent in this representation.

The incompatibility of transmission and assortment may be verified for this representation by attempting to generate an instance of the forma described by $\{(4, \{3, 5\}), (6, \{3, 7\})\}$ from parents $12345678^p$ and $12367458^p$.

## 6.5 Linkage Considerations

The positioning of genes on a chromosome defines its *linkage*. With crossover operators based on transferring contiguous portions of the genome from parents to children, such as $N$-point crossover, linkage can significantly affect algorithmic performance. In the case of the permutation representation, where the $i$th gene specifies the $i$th city visited, the static linkage achieved by placing the $i$th gene at the $i$th locus seems natural. With the other genetic representations—directed edges and corners—and the pseudo-genetic representation based on undirected edges, it is less clear that placing the $i$th gene at the $i$th locus is sensible. This is because in each of these cases, the $i$th gene is associated with the $i$th-*labelled* city, rather than the $i$th city in the tour, and the city labels are (usually) arbitrary. A simple way to determine the linkage adaptively is to re-link the genome in one of the two possible orders achieved by following the tour in a consistent direction. It seems at least possible that GNX using this linkage will perform better than using the essentially random linkage achieved by determining locus from city number. Both forms of linkage will therefore be tried with GNX.

## 6.6 Redundancy and Degeneracy

In order to classify representations further, it is useful to introduce the distinct notions of redundancy and degeneracy. These notions are often confused, a failing of which the authors, amongst others, have previously been guilty.

A representation is said to exhibit *redundancy* if a solution can be uniquely determined from a subset of its genes (or in the case of allelic representations, its alleles). All four representations considered contain some redundancy, because the last allele value can always be determined from the others (owing to the cyclic nature of the tour). The corner representation, however, contains vastly more than the other representations, actually specifying each edge twice. Thus fully half of the genetic material in the corner representation is formally redundant—specifying every other corner in the tour would suffice. Notice, however, that no *fixed* set of corners (those centred on a particular half of the cities) would be adequate, which is why the representation used specifies the corner for each city. High redundancy necessarily gives rise to a high degree of non-orthogonality, but it is not clear to what extent—if any—this inhibits the search.

In contrast, a representation is said to exhibit *degeneracy* if more than one chromosome is used to represent the same solution, (i.e. if the genotype-phenotype mapping is non-injective). Both the permutation representation and the directed edge representation exhibit degeneracy, while the undirected edge representation and the corner representation do not. While folding out the factor of

$n$ arising from the need to fix a starting city is easily handled, the degeneracy associated with the direction of travel is more problematical.

There are a number of possible responses to degeneracy:

1. *Gauge Fixing.* As has been stated earlier, the problem can be removed formally by a convention such as that used in this paper—always choosing the direction so that the second city has a lower numeric label than the $n$th. The problem with this is that some very similar solutions have almost maximally different representations. For example, $162453^p$ is a minimal mutation of $126453^p$, but in standard form is represented as $135426^p$. This is a practical problem for recombination, which will fail to recognise that the two solutions have anything in common except the redundant first gene and equidistant fourth gene, and leads to a certain "brittleness".
2. *Ignore the problem.* It is possible simply to leave the evolutionary search to use two different representatives for each solution. This avoids the brittleness of the "gauge fixing" approach, but at the cost of searching a larger space with two optima, and (more importantly) failing to allow recombination to recognise even when it is manipulating identical parent solutions. While there is some evidence that evolutionary search is still reasonably effective in these circumstances, it is hard to avoid the suspicion that its efficiency is reduced.
3. *Align before Recombination.* The practical difficulties of degeneracy manifest themselves during recombination. An alternative approach involves computing the distance between the first parent the second traversed in one sense, and then computing the distance with the second parent reversed. Recombination is performed with second parent in the sense that minimises the distance. This is similar in spirit to an approach used by Montana & Davis (1989) for removing the well-known hidden node degeneracy in feed-forward neural networks before recombination. This avoids the problem of brittleness and recombination's consequent inability to recognise certain very similar solutions, but is arguably somewhat arbitrary.

All of these options have drawbacks. For this study, the second option has been chosen as the simplest.

### 6.7 Characteristics of TSP Representations

Before proceeding to the experimental results, it will prove useful to summarise the key characteristics of the four representations considered. These are presented in table 1.

## 7 Experiments and Results

Figure 2 shows the standard deviation of fitness within formae as a function of order for the four different representations. Each point is based on 100 samples from each of 100 formae of the given order, for the 100-city Krolak 'C' problem from TSPLIB (Reinelt, 1990). Graphs of higher fitness moments are qualitatively rather similar, and are not shown.

TSP optimisation experiments were then conducted using both genetic and memetic algorithms, using $\text{RAR}_2$ and G2X for recombination, as well as edge recombination for a domain-specific comparison. For all representations except permutations, G2X was applied with both fixed linkage and the "tour-following" scheme described in section 6.5. The order of application of operators was recombination (with probability $1.0$) of two parents chosen by probabilistic binary tournament selection, followed by BMM (defined with respect to the chosen representation) with a rate chosen so

| representation | | cardinality | | degeneracy | mutation-order | | linkage | |
|---|---|---|---|---|---|---|---|---|
| | rep. type | | redundancy | | (edges) | (genes) | | variance |
| permutation | genetic | $O(n)$ | low | $\times 2$ | 4 | 2 | good | high |
| directed edge | genetic | $O(n)$ | low | $\times 2$ | 3 | 3 | random | medium |
| undirected edge | allelic | — | low | none | 2 | 2 | none | medium |
| undirected edge | pseudo | $O(n)$ | low | $\times 2$ | 2 | 2 | random | medium |
| corner | genetic | $O(n^2)$ | high | none | 2 | 4 | random | low |

Table 1: This table shows the main characteristics of the representations studied for the TSP. Cardinality indicates the number of allele values per gene (where applicable). Degeneracy and redundancy are discussed in section 6.6. Mutation order lists first the number of edges and second the number of gene values changed by a minimal mutation. The linkage column describes the appropriateness of the "natural" linkage for the representation, and variance gives the relative fitness variance of formae, as shown in figure 2. Note that the undirected edge representation is shown twice, once in its allelic form and once as the pseudo-genetic representation discussed in section 6.2.

that the mean number of mutations per chromosome was one. In the case of the memetic algorithms, minimal-mutation-based hill-climbing was then performed to find a local optimum. A panmictic population of size of 100 was used with a steady state update scheme. All performance results are averages over 20 runs, and show the length of the best tour in the population relative to the optimum. Each representation started from the same set of 20 randomly-generated populations. Error bars are omitted since they are in all cases smaller than the tick sizes.

Both the 100-city Krolak 'C' and the 442-hole PCB drilling problem from TSPLIB were studied. The genetic algorithm for the smaller problem used probabilistic binary tournaments with $p = 0.7$ for selection and replacement, with elitism, and allowed duplicates. Results for these runs are shown in figures 3, 4 and 5. A memetic algorithm (not shown) solves this problem to optimality extremely quickly.

The larger problem used a more aggressive GENITOR-style plan adapted from Whitley *et al.* (1989), using tournament selection with probability 1.0, and replacement of the worst individual. Duplicate solutions were forbidden. For edge recombination only, a zero mutation rate was used in line with its creators' recommendation (to achieve maximal edge transmission). Both genetic and memetic algorithms were used for this problem, with genetic results shown in figures 6 and 7 and memetic results in figure 8.

In general, the results show fitness variance of formae to be a powerful indicator of algorithmic performance. The results expected on this basis would be that for any fixed algorithm, corners should perform best, permutations worst, and the edge representations somewhere in between. The only discrepancies that need to be explained are now addressed in turn.

First, in four of the five direct comparisons, undirected edges out-perform directed edges despite similar forma variance. Factors explaining this might include the difference in mutation operators (two-changes for undirected edges versus three-changes for directed edges) and the greater disruptiveness of recombination for directed edges. The one case in which directed edges do better is a very aggressive plan with RAR. Here, it is possible that mutation is performing a more important search rôle, and greater recombinative disruption effectively increases the mutation rate advantageously.

Secondly, while permutations generally perform poorly, as forma variance would suggest, in one case (GNX on the 100 city problem) they perform rather better than might be expected. This is probably in part because GNX takes long contiguous chunks from parents, thus effectively transmitting many edges even for permutations. As noted in figure 2, if the formae considered are restricted to be those with contiguous defining positions, forma variance falls almost exactly to that for the edge-based representations, largely explaining this anomaly.

The final discrepancy concerns the performance of the corner representation relative to that of undirected edges. Here, the general pattern is that when good linkage is used and maintained, performance is very similar, but when poorer linkage is used, or is severely disrupted (as with RAR), corners significantly outperform edges. This is wholly understandable since corners intrinsically carry much more linkage information, in the sense that every corner specifies an adjacency of two edges. The only case in which this pattern breaks down is for the smaller problem where corners and undirected edges perform similarly with RAR.

Other points evident from the results are that GNX consistently out-performs RAR, linkage effects are (perhaps unsurprisingly) rather strong and the memetic algorithms produce dramatically better results in absolute terms than do genetic algorithms on the problems considered. More surprisingly, GNX with both the corner and undirected edge representations, even with non-optimised parameters, appears at least competitive with, and arguably superior to, edge recombination.

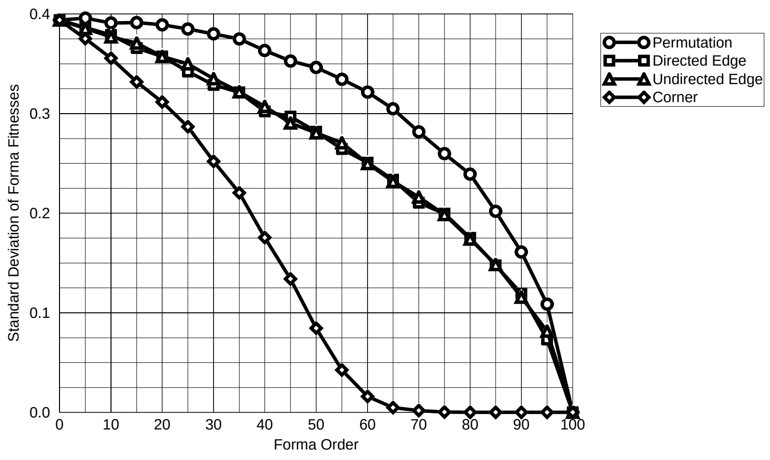


Figure 2: The graph shows the mean standard deviation of fitness for 100 samples drawn from 100 randomly generated formae at each order shown for the four representations considered. They are based on the 100 city Krolak ‘C’ problem. Note that if the graphs were shown as functions of forma size, rather than forma order, the corner line would be coincident with those for the edge representations. Further, if the formae are restricted to have adjacent defining positions, the permutation line becomes almost coincident with the edge lines. Confidence intervals (standard errors) are omitted as they are smaller than the tick marks shown.

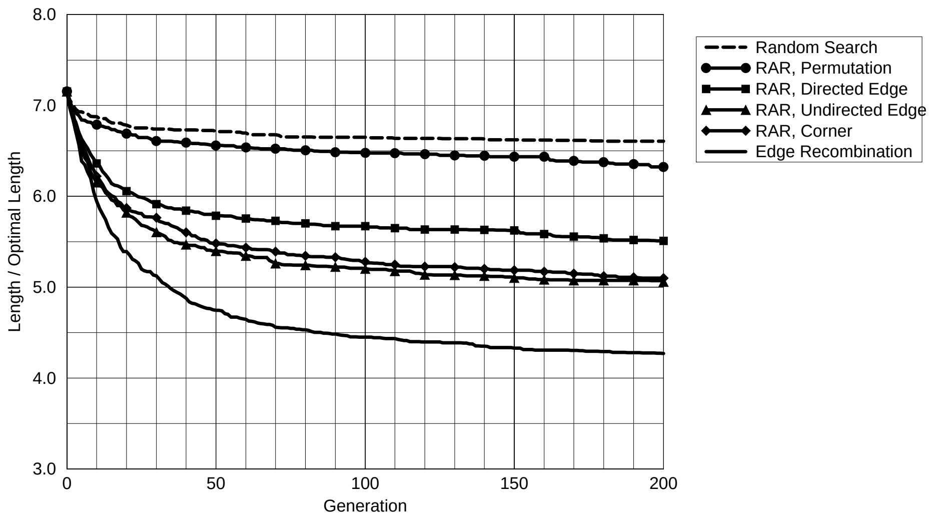


Figure 3: The graph shows the results produced by representation-independent genetic algorithms on the 100-city Krolak ‘C’ problem, instantiated with four representations using the RAR recombination operator. For comparison, a curve for random search is shown, as is Whitley *et al.*’s edge recombination operator, modified by deferring patching of tour fragments until all parent edges are exhausted.

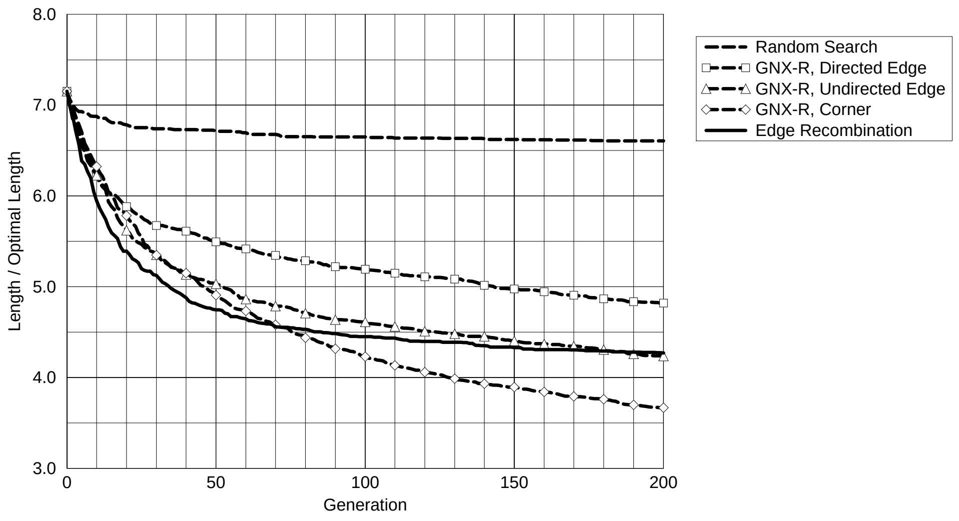


Figure 4: The graph shows the results produced by representation-independent genetic algorithms on the 100-city Krolak ‘C’ problem, using the GNX-R recombination operator (c.f. figure 3). These runs use the “natural” linkage associated with the representation. For the edge-based and corner representations, this linkage is essentially random (the gene’s locus being determined by its city label).

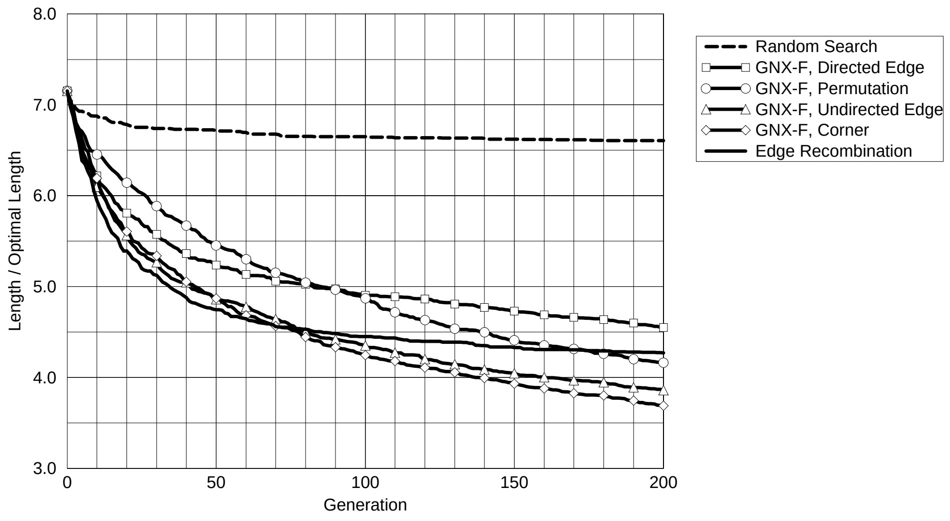


Figure 5: The graph shows the results produced by representation-independent genetic algorithms on the 100-city Krolak 'C' problem, using the GNX-F recombination operator (c.f. figures 3, 4). These runs use a dynamic "tour-following" linkage.

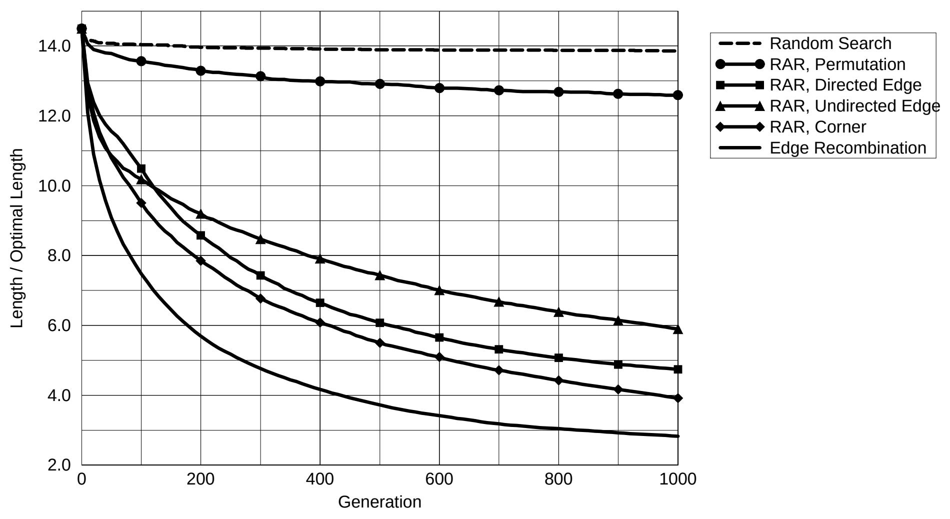


Figure 6: The graph shows results for a GENITOR-style genetic algorithm with RAR on the 442-hole PCB drilling problem corresponding to those in figure 3.

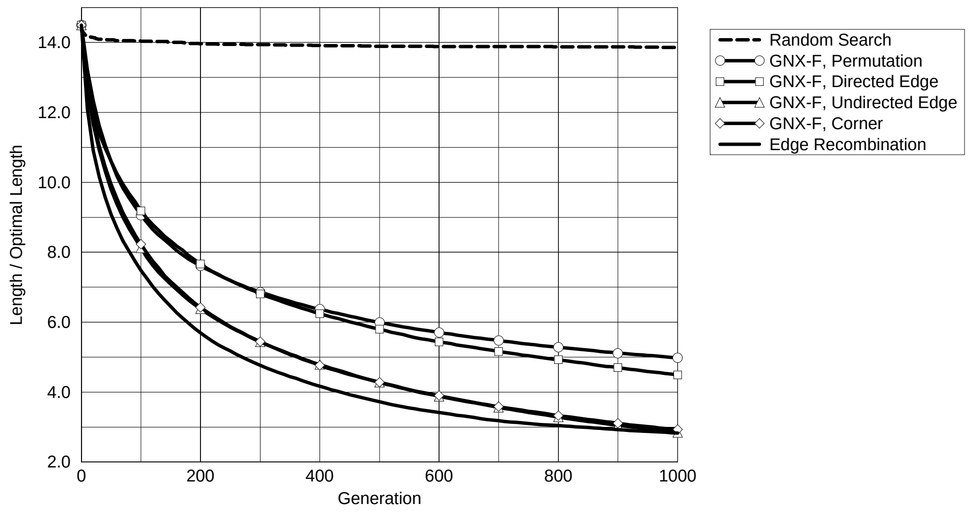


Figure 7: The graph shows results for a GENITOR-style genetic algorithm with GNX on the 442-hole PCB drilling problem corresponding to those in figure 5. Results for the inferior random linkage corresponding to figure 4 are omitted.

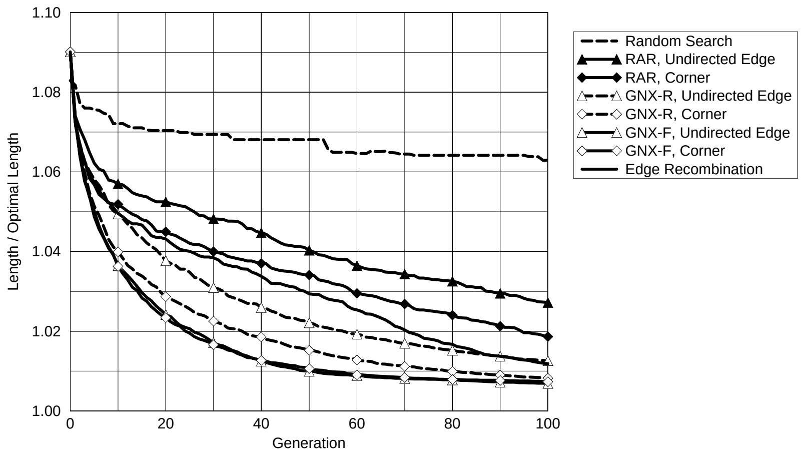


Figure 8: The graph shows the same problem as is presented in figures 6 and 7, but now using a memetic algorithm that applies full minimal-mutation-based optimisation before each evaluation. Note the different scales used on the axes. Results for the permutation representation are omitted owing to the extremely long generation times required. Furthermore, even on a per-generation comparison, results are not competitive with those shown. Results are also omitted for directed edges, again because of excessive compute times required. These arise because the minimal mutation for this representation is a 3-change, which requires at least $O(n)$ more computation than the 2-changes needed for undirected edges and corners.

## 8 Discussion and Conclusions

The general pattern of results supports the hypothesis that the fitness variance of forma exhibited by a representation acts as a good predictor of its performance in formal genetic and memetic algorithms. Indeed, given the large number of potentially relevant differences between the four representations considered, the predictive power of forma variance is rather impressive. While results have only been gathered for one problem class (the travelling sales-rep problem) and a limited range of representation-independent algorithms, they provide a powerful case for corresponding studies in other problem domains.

The best results overall were produced with the corner representation, which has a number of unusual features. Principal among these is its extremely high cardinality and compound allele structure. The results therefore provide further evidence that the traditional advocacy of low cardinality representations as universally appropriate is misguided.

The results obtained with GNX are at least competitive with, and arguably superior to, those obtained with edge recombination. Since this is widely regarded as the best form of recombination for the TSP when a genetic algorithm without local search is used, this is a significant finding. Moreover, since GNX is applicable to *any* (formal) genetic representation (including non-orthogonal representations) it may well prove effective in other problem domains. These studies have demonstrated that the construction of formal representation-independent operators and algorithms is not merely of theoretical import, but can provide competitive practical search tools.

Linkage has long been recognised as an important theoretical characteristic of chromosomes in the context of recombination, but has rarely been shown to have a major effect on performance in practice. These experiments have clearly demonstrated linkage effects, and shown that adaptive linkage strategies—albeit not the traditional inversion-based approach—can yield superior performance. The best results achieved were with corners, which contain intrinsic linkage information, and undirected edges when linked by tour following.

Finally, these results strongly confirm the dramatically superior performance that can be achieved by incorporating local search in genetic algorithms for TSP to yield memetic algorithms. The TSP is a natural candidate for memetic search because the fitness function is decomposable, allowing very cheap testing of minimal mutations, but it should be noted that even if full evaluation is performed at each memetic step the overall performance of the memetic algorithms discussed is still superior.

## References

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

Lawrence Davis, 1991. Bit-climbing, representational bias, and test suite design. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Kenneth A. De Jong, 1992. Genetic algorithms are NOT function optimizers. In Darrell Whitley, editor, *Foundations of Genetic Algorithms 2*. Morgan Kaufmann (San Mateo, CA).

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

Reimar Hofmann, 1993. Examinations on the algebra of genetic algorithms. Diploma Thesis, Technical University of Munich, Department of Computer Science.

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

David J. Montana and Lawrence Davis, 1989. Training feedforward neural networks using genetic algorithms. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 762–767.

Pablo Moscato and Michael G. Norman, 1992. A “memetic” approach for the travelling salesman problem — implementation of a computational ecology for combinatorial optimisation on message-passing systems. In *Proceedings of the International Conference on Parallel Computing and Transputer Applications*. IOS Press (Amsterdam).

Nicholas J. Radcliffe and Patrick D. Surry, 1994. Formal memetic algorithms. In Terence C. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, pages 1–16. Springer-Verlag, Lecture Notes in Computer Science 865.

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

Nicholas J. Radcliffe, 1992. Non-linear genetic representations. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature 2*, pages 259–268. Elsevier Science Publishers/North Holland (Amsterdam).

Nicholas J. Radcliffe, 1994. The algebra of genetic algorithms. *Annals of Maths and Artificial Intelligence*, 10:339–384.

Gerhard Reinelt, 1990. TSPLIB. Available by anonymous FTP from softlib.rice.edu.

J. David Schaffer, Richard A. Caruana, Larry J. Eshelman, and Rajarshi Das, 1989. 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).

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

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

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

Darrell Whitley, Timothy Starkweather, and D’Ann Fuquay, 1989. Scheduling problems and traveling salesmen: The genetic edge recombination operator. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Darrell Whitley, Timothy Starkweather, and Danial Shaner, 1991. The traveling salesmen and sequence sheduling: Quality solutions using genetic edge recombination. In Lawrence Davis, editor, *Handbook of Genetic Algorithms*. Van Nostrand Reinhold (New York).