Evolving Presentations of Genetic Information: Motivation, Methods, and Analysis

Peter Lee
Stanford University
PO Box 14832
Stanford, CA 94309-4832
(650)497-6826
peterwlee@stanford.edu

June 5, 2002

Abstract

We present a hybrid genetic meta-algorithm that while applying a standard naïve genetic algorithm attempts to concurrently evolve a presentation of the genetic information that makes simple single-point crossovers increasingly effective. The idea is to attach positional information to genes and evolve them also by crossover and mutation, resulting in a presentation in which successful schema have shorter defining length and more likely to survive crossover. Initial results are promising and act as a proof-of-concept; improvements, future directions and an ideal solution are also discussed.

1 Motivation

The goal of the crossover operation in genetic algorithms is to combine into increasingly fit individuals superior notions of what is important or relevant to the task. As a concrete example of a crossover operator, given a genome of length n and a linear presentation of genes or genetic information, i.e. the ordered n-tuple $(g_1, g_2, \ldots, g_n)$, simple single-point crossovers yields $n-1$ possible crossover sites and therefore $n-1$ possible crossover actions. However, the limitation to single-point crossovers is artificially imposed and arguably without basis; after all, crossovers in nature are very much multi-point. Assuming that genes have to be preserved in crossover, there are $O(n^k)$ possible k-point crossovers for $k \ll n$. The total number of crossovers is clearly $2^{n-1} \gg n^k > n-1$, and clearly to restrict ourselves to k-point crossovers is unnatural. Indeed, much of the research in genetic algorithms and genetic programming has focused on realizing a more versatile crossover operation.

We shall see that crossovers are not created equal in that there are certain crossovers that yield quicker convergence to solutions when used in place of other crossovers, and we will give a concrete example to illustrate the importance of the crossover in Section 5. However, to try all 2^{n-1} crossover operators would be computationally infeasible, so then our goal is to build a genetic algorithm that gradually evolves such superior crossovers.

However, we have two options in attempting to evolve a better crossover operator. We can directly modify the crossover operator, but how to do this is not immediately clear. Better is to evolve the linear presentation of genetic information while still using a simple-crossover operator. After all, there is an instrinsic notion of **distance** between genes in relation to a fitness function, a notion that we will formalize and quantify in Sections 2 and 3. Intuitively, certain combinations of genes, or schema, may result in a particularly fit individual, while fragments of those combinations may be worthless. In this case, we shall say that those genes are somehow close to each other in that the effect of one gene depends on the value of the others. Conversely, if two genes somehow contribute independently to the fitness value, then we can say that they are further apart. Notice that if we want to keep successful schema intact with high probability, then the genes that make up such schema should be close on the chromosome for a shorter defining length and greater chance of surviving crossover. So in fact we see that evolving presentations of genetic information is an equally natural way to interpret evolving crossovers. Section 5, then, is really about the importance of the presentation of genetic information as well as the significance of the crossover operator.

The precise mechanics of the presentation of genetic information and resulting crossovers will be described in Section 6.

2 Gene correlation

Let us generalize our notion of an individual's genetic information. For simplicity, we will assume that the amount of genetic information an individual of a particular species can carry is bounded, hence without loss of generality we consider fixed-length chromosomes. Let n be the length of the chromosome and $G_1, G_2, \ldots, G_n$ be finite sets which we shall call **gene spaces**. Then we define the set of all possible **chromosomes**, which we shall call the **chromosome space**, to be

$$G = G_1 \oplus G_2 \oplus \cdots \oplus G_n$$

where $\oplus$ denotes Cartesian product or direct sum, and not exclusive or. An individual's genetic information is therefore simply an element $g \in G$.

Notice that the indexing of the gene spaces are somewhat misleading, for it implies that the genetic information of an individual is intrinsically ordered in some linear fashion. While this sequential presentation may be the case for most biological organisms, there is no reason to suspect that chromosomes should be presented in this way for artificial problems as well. In fact, bacteria have circular DNA, but more different are problems in image recognition and other high-dimensional problems, where data is intrinsically nonlinear. So the numerial indexing of gene spaces is somewhat deceiving; instead, we will use a more abstract approach and rewrite our chromosome space to be

$$G = \bigoplus_{\alpha \in \mathcal{A}} G_\alpha \tag{1}$$

where $\mathcal{A}$ is some finite indexing set.

Using the notion of a chromosome space, we can now very easily redefine some familiar concepts. Fitness can be defined as a function $f : G \to \mathbb{R}$ mapping the chromosome set to the set of real numbers. Maximum fitness is $\max\{f(g) | g \in G\}$, and average fitness of a subset $H \subset G$ of individuals can be written as, through an abuse of notion,

$$\bar{f}(H) = \frac{1}{|H|} \sum_{h \in H} f(H)$$

Furthermore, let C_G be a set of crossover operators, namely, elements that define a crossover operation between two arbitrary individuals. Our crossover function is $l, r : G^2 \times C_G \to G$, where $l(g_1, g_2, c)$ and $r(g_1, g_2, c)$ for the two children of g_1 and g_2 crossing over as specified by c. For example, for single-point crossovers of two chromosomes of length n, we can write C as $\{1, 2, \ldots, n-1\}$, where an element in C determines the crossover site.

In order to explore the notion of correlation between genes (or more precisely, gene spaces), let us examine something simpler first: a polynomial of two variables. We shall ask the question, when do we consider these two variables to be correlated?

Suppose we have $f : \mathbb{R}^2 \to \mathbb{R}$ taking $(x, y) \mapsto 1 + x^2 + y^2$. Certainly, the value of $f(x, y)$ depends on both x and y, but more important are the separate effects of x and y on $f(x, y)$. Consider the relation $f(x+1, y) = f(x, y) + 2x + 1$, which states that the effect of incrementing x by 1 has the effect of incrementing $f(x, y)$ by $2x + 1$. But notice that the effect on $f(x, y)$ was independent of the value of y, namely, regardless of the value of y, the effect of incrementing x will always be the same. It is natural now to claim that x is independent of y, and similarly y is independent of x.

Now consider $f(x, y) = xy$. The effect of incrementing x depends on the value of y, namely, $f(x+1, y) = f(x, y) + y$. So we suspect that x and y are somehow correlated.

Definition 1 (Correlation of variables) *Let X, Y, and Z be arbitrary spaces and $f : X \times Y \to Z$. We say that the variables in X and Y are* **(additively) independent** *or* **(additively) uncorrelated** *with respect to f if there exists functions $g : X \to Z$ and $h : Y \to Z$ such that for all $x \in X$ and $y \in Y$*

$$f(x, y) = g(x) + h(y)$$

If no such decomposition exists then we say that the variables are **(additively) dependent** *or* **(additively) correlated***.*

It is clear that x and y are uncorrelated in $f(x, y) = 1 + x + y$ for we can write $g(x) = 1 + x$ and $h(y) = y$. But it is not immediately rigorously clear that x and y are correlated in $f(x, y) = xy$, so we provide a short proof. Suppose $f(x, y) = g(x) + h(y)$, so we have $f(x + 1, y) = g(x + 1) + h(y)$ and therefore $g(x + 1, y) + h(y) = g(x) + h(y) + y$, implying $y = g(x + 1) - g(x)$, a contradiction. Interestingly enough though, x and y are uncorrelated in $\log f$.

We are now in a position to define the notion of correlation between genes. First, a bit of terminology: let $G = \oplus_{\alpha \in \mathcal{A}} G_\alpha$ be a chromosome space, and let $\mathcal{B} \subset \mathcal{A}$. Then we define:

$$G_{\mathcal{B}} = \bigoplus_{\beta \in \mathcal{B}} G_\beta$$

Notice that $G_{\mathcal{B}}$ is simply a projection of G. Similarly, let $x_{\mathcal{B}}$ be the coordinate-wise restriction of x to $\mathcal{B} \subset \mathcal{A}$.

Definition 2 (Correlation of genes) *Let $G = \oplus_{\alpha \in \mathcal{A}} G_\alpha$ be a chromosome space with fitness function $f : G \to \mathbb{R}$. Two gene spaces G_β and G_γ, $\beta, \gamma \in \mathcal{A}$, are called* **(additively) independent** *or* **(additively) uncorrelated** *if there exists two functions $g : G_{\mathcal{A} \setminus \{\beta\}} \to \mathbb{R}$ and $h : G_{\mathcal{A} \setminus \{\gamma\}} \to \mathbb{R}$ such that for all $x \in G$,*

$$f(x) = g(x_{\mathcal{A} \setminus \{\beta\}}) + h(x_{\mathcal{A} \setminus \{\gamma\}})$$

More generally, we say that two disjoint collections X and Y of gene spaces are independent or uncorrelated when any space from X and any space from Y are independent.

Without loss of generality, we can assume a fitness function f depends on all of its gene spaces, for otherwise we can just discard that gene space. Therefore by the definition above, all gene spaces are correlated with themselves, so the binary relation of two genes being correlated is reflexive as well as symmetric. If we interpret gene spaces as discrete points, this binary relation of correlation forms an undirected simple graph, which we call the **correlation graph**. The components of this graph are exactly pairwise disjoint and uncorrelated and will be called **correlation components**.

3 Degree of correlation

Disjoint uncorrelated collections of gene spaces or correlation components may not be a common occurence in real-life fitness functions. But the relation of correlation between two gene spaces need not be limited the black and white situation of correlated or uncorrelated. Referring back to polynomials in two variables, consider the fitness functions $f(x, y) = 1 + x + y + 2^{100}xy$ and $g(x, y) = 1 + x + y + 2^{-100}xy$ for $x, y \in [-1, 1]$. We would like to say that x and y are weakly correlated with respect to g, for the effect of changes in x depends very little on the value of y, whereas for f this is not the case. So consider two gene spaces G_β and G_γ that are correlated. Then the fitness function $f = f_1 + f_2 + \cdots + f_m$ must be written as a sum of functions, without loss of generality exactly one of which, say, f_1, requires both the G_β and G_γ genes, for otherwise by definition they would be uncorrelated. Notice then that there are multiple ways of decomposing f. However, we can assign the **degree of correlation** between G_β and G_γ to be the minimum value of $|f_1|_\infty$

over decompositions of f, where $|f_1|_\infty$ denotes the maximum value of f_1. So if there is a decomposition such that $|f_1|$ is very small, then the degree of correlation is small and we can say that G_β and G_γ are weakly correlated. Notice that our choice of the $|\cdot|_\infty$ norm was arbitrary; we can use the L^1, L^2, or L^p norm, $1 \leq p < \infty$ and not the L^∞ norm.

This notion of weak and strong correlation corresponds exactly to the intrinsic distance between genes; weakly correlated genes should be further apart and strongly correlated genes should be closer. For simplicity, we shall not make explicit the changes in our theory if we take into account weak and strong correlation.

4 The schema theorem

In order to give a quantitative discussion why the presentation of genetic information affects how quickly superior individuals evolve, we must examine the evolutionary pressures placed on such individuals with superior presentations. The schema theorem states that

$$\langle m(H,t+1)\rangle \geq m(H,t)\cdot\frac{f(H,t)}{\bar{f}(t)}\left[1-p_c\cdot\frac{\delta(H)}{l-1}\right][1-p_m]^{o(H)}$$

where $\langle m(H,t+1)\rangle$ is the expected number of individuals that represent the schema H at generation $t+1$, $m(H,t)$ is the expected number of individuals that represent the schema H at generation t, $f(H,t)$ is the average fitness value of the individuals containing schema H at generation t, $\bar{f}(t)$ is the average fitness of the population at generation t, p_c is the crossover probability, p_m is the mutation probability, $\delta(H)$ is the defining length of H, $o(H)$ is the order of H and l is the string length.

The schema theorem above is written for a fixed presentation of genetic information; some of the terms would not be well-defined if we allowed changes of genetic structure as our new algorithm would. However, notice that the schema theorem can be more generally written as

$$\langle m(H,t+1)\rangle \geq m(H,t)\cdot A\cdot B\cdot C$$

where A is the expected change in frequency of the schema H, B is the probability of an individual representing schema H still representing schema H after crossover, and C is probability of an individual representing schema H still representing schema H after mutation. The reader will realize that the goal of evolving gene presentations is to increase B for superior schema. However, the nature of the schema theorem does not allow us to say that a certain presentation is better than another presentation because it provides only a lower and not upper bound on $\langle m(H,t+1)\rangle$.

5 Significance of presentation

To illustrate the importance of how genetic information is presented, consider the following trivial problem which we shall try to solve using genetic algorithms. For a sequence $b = \{b_1, b_2, \ldots, b_n, b_{n+1}, \ldots, b_{2n}\}$ of binary digits, we wish to maximize the following function:

$$f(b) = \mathrm{Maj}(\{b_1,\ldots,b_n\})^2 + \mathrm{Maj}(\{b_{n+1},\ldots,b_{2n}\})^2$$

where we define, for any finite collection $\mathcal{S}$ of 0s and 1s:

$$\begin{aligned} m &= \text{number of elements in } \mathcal{S} \\ \mathrm{num}(\mathcal{S},i) &= \text{number of occurrences of } i \text{ in } \mathcal{S} \\ \mathrm{Maj}(\mathcal{S}) &= \max\{\mathrm{num}\{\mathcal{S},\mathtt{0}), \mathrm{num}(\mathcal{S},\mathtt{1})\} - \lceil\frac{m}{2}\rceil \end{aligned}$$

Notice that $\mathrm{Maj}(\mathcal{S}) \geq 0$. Clearly, $f(b)$ has four global maxima, namely $\{b_1,\cdots,b_n\}$ must identically be 0 or 1 and similarly $\{b_{n+1},\cdots,b_{2n}\}$ must also be identically 0 or 1 for a maximum score of $2\lfloor\frac{n}{2}\rfloor^2$.

We applied a classical genetic algorithm to solve this problem, using crossovers, mutation, and elitist selection. Unsurprisingly, all runs converged rapidly to some global maximum even for large values of n.

The quick convergence to an optimal individual can be explained in several ways. First, all the advantages of a genetic algorithm comes into play. However, one must not underestimate the importance of how effectively genetic information was presented and crossed. In this preceding example, two individuals A and B were crossed as follows: $\{a_1, \ldots, a_{2n}\}$ and $\{b_1, \ldots, b_{2n}\}$ became $\{a_1, \ldots, a_i, b_{i+1}, \ldots, b_{2n}\}$ and $\{b_1, \ldots, b_i, a_{i+1}, \ldots, a_{2n}\}$ for some uniformly random i, $1 \leq i \leq 2n-1$. In particular, given a highly fit individual, crossover was sure to produce a child that inherited at least some of the parent's superior genes. For example, let A be the individual

$$\overbrace{\mathtt{100\cdots0}}^{n \text{ genes}}\overbrace{\mathtt{011\cdots1}}^{n \text{ genes}}$$

The fitness of this individual is $f(A) = 2(\lfloor \frac{n}{2} \rfloor - 1)^2$. And in particular, no matter how this individual is crossed, namely, regardless of its partner and crossover site, at least one of its offspring A' will have fitness $f(A') \geq (\lfloor \frac{n}{2} \rfloor - 1)^2$, a sharp bound. In general, a fit individual is extremely likely to have a child that is relatively fit.

However, our confidence of a fit parent giving rise to a fit child is completely lost of the presentation of genetic information is changed, thereby effectively changing the crossover operation. Whereas the genetic information of an individual was presented as $\{b_1, b_2, \ldots, b_{2n}\}$, suppose we now present the genetic information as

$$a_1, a_{n+1}, a_2, a_{n+2}, \ldots, a_n, a_{2n}$$

Then our individual A can potentially give rise to two children with zero fitness, a large difference to our earlier greatest lower bound. In fact, it is easy to see that *any* individual can give rise to two children of zero fitness.

The two different presentations of genetic information clearly partially determine how effective the crossover is at producing fit individuals.

6 Evolving effective presentations

We have established that arbitrary gene presentations can result in poor convergence. Since gene correlation may not be necessarily obvious given an evaluation function, we attempt to evolve presentations that gradually place correlated genes closer to one another, thereby creating a better crossover and quicker convergence. We do this by attaching positional information to each gene and having a crossover operator that acts with respect to the positional information. The result is that common crossover operators can be evolved, in particular the equivalent of multi-point crossover, circular chromosome-crossover, and in fact any crossover as defined as follows:

Definition 3 *Let $x, y \in G = \oplus_{\alpha \in \mathcal{A}} G_\alpha$ be two chromosomes in a chromosome space. Then a crossover C is said to be* **gene-preserving** *if for each $\alpha \in \mathcal{A}$, the genes of x and y at position α are exactly the genes of the children x' and y' at position α, although not necessarily in that order.*

We now describe the positional crossover operator mentioned above, and instead of getting mired in subscripts, we simply present an example that can be easily generalized. Let G consist of 16 gene spaces, labelled $G_1, \ldots, G_{15}$. Then x and y are each associated with x_{pos} and y_{pos}, two permutations of $1, \ldots, 15$. For example, let $x_{pos} = (13, 4, 5, 8, 15, 6, 12, 9, 14, 10, 1, 7, 3, 2, 11)$. Then we present the genetic information of x as

$$x_{13}\ x_4\ x_5\ x_8\ x_{15}\ x_6\ x_{12}\ x_9\ x_{14}\ x_{10}\ x_1\ x_7\ x_3\ x_2\ x_{11}$$

where $x_i \in G_i$. Similarly, let y be presented as

$$y_{10}\ y_1\ y_8\ y_6\ y_2\ y_5\ y_{12}\ y_{11}\ y_{13}\ y_9\ y_4\ y_{14}\ y_7\ y_{15}\ y_3$$

Then, as in a single point crossover, we select a crossing site between 1 and 14, so suppose $p = 6$. Then proceed as follows: the first 6 positions of x' will the the first 6 positions of x. The last 9 positions of y' will the the last 9 positions of x. The last 9 positions of x' will be from y. Clearly, x' is missing the 12, 9, 14, 10, 1, 7, 3, 2, and 11 genes. We will take them from y, but in the order as they appear in y, and similarly for y'. So we get

$$x_{13}\ x_4\ x_5\ x_8\ x_{15}\ x_6\ y_{10}\ y_1\ y_2\ y_{12}\ y_{11}\ y_9\ y_{14}\ y_7\ y_3$$

and

$$y_8\ y_6\ y_5\ y_{13}\ y_4\ y_{15}\ x_{12}\ x_9\ x_{14}\ x_{10}\ x_1\ x_7\ x_3\ x_2\ x_{11}$$

Notice that this crossover is not symmetric, in that the distances between close gene spaces in x are preserved much more strongly than in y. For this reason, in our implementation we shall preserve more strongly the individual with a higher score.

7 Convergence of presentations

Consider a gene space where correlated variables are split into two correlation components, namely, $G = \oplus_{\alpha\in\mathcal{A}} G_\alpha$ and $\mathcal{A}$ is a disjoint union of $\mathcal{B}$ and $\mathcal{C}$, and any two gene spaces from $\mathcal{B}$ and $\mathcal{C}$ are uncorrelated. Then we would like a result that states that given some presentation of the genes in $\mathcal{B}$ and $\mathcal{C}$, a presentation that keeps them distinct always presents better convergence characteristics than a presentation that interleaves genetic information from $\mathcal{B}$ and $\mathcal{C}$.

Notice that the schema theorem already suggests this result. As mentioned before, the schema theorem only presents a lower bound, and more importantly is for fixed presentations, whereas we now are evolving the presentation and therefore the crossover operator.

Let $S^{\mathcal{B}}$ be the set of all schema whose specificity lies only in $\mathcal{B}$, namely, all schema that places no restriction on $\mathcal{A}\setminus\mathcal{B}$. Then assuming that gene spaces in $\mathcal{B}$ are finite, we have

$$\left|S^{\mathcal{B}}\right| = \prod_{\beta\in\mathcal{B}} \left(|G_\beta| + 1\right)$$

We further define $S^{\mathcal{B}}_n \subset S^{\mathcal{B}}$ to be the subset of schema of specificity n. Finally, by an abuse of notation let us by denote $f(S)$ the average fitness of all individuals that satisfy some schema $s \in S \subset S^{\mathcal{A}}$.

Definition 4 *Let $G = \oplus_{\alpha\in\mathcal{A}} G_\alpha$ be a chromosome space. Then $\mathcal{B}\subset\mathcal{A}$ is a set of* **correlated genes** *if there is a path in $\mathcal{B}$ between any two genes in $\mathcal{B}$ such that each edge is spanned by correlated genes. In other words, $\mathcal{B}$ lies entirely in some correlation component.*

Definition 5 *Let $\mathcal{B}$ be a group of correlated genes, and let $n = |\mathcal{B}|$. Then $\mathcal{B}$ is called* **normal** *if for any element $g \in G$ such that $f(g) > f(G)$, $f(S^{\mathcal{B}}_i(g))$ is increasing with i, $1 \le i \le n$.*

Notice that $\mathrm{Maj}(b_1, b_2, \ldots, b_n)$ is normal with respect to its inputs.

Definition 6 *Let $A, B \subset \mathbb{N}$ and $|A| = |B| = n < \infty$. Let a_i and b_i denote the i-th largest element in A and B respectively. Then we say that A is* **as clustered** *than B if $a_j - a_i \le b_j - b_i$ for all $1 \le i < j \le n$. If inequality holds for some i and j, we say that A is* **more clustered** *than B.*

Definition 7 *A* **linear presentation** *of a chromosome space $G = \oplus_{\alpha\in\mathcal{A}} G_\alpha$ is a total order of $\mathcal{A}$, or more precisely, a bijection between $\mathcal{A}$ and $\{1, 2, \ldots, |\mathcal{A}|\}$. In particular, a presentation assigns a unique* **index** *to every gene space.*

Definition 8 *Let P and Q be presentations of some chromosome space. Then P* **converges faster** *than Q if a single-point crossover genetic algorithm using P on average converges to a solution more quickly than if Q is used.*

Definition 9 *Let $\mathcal{A} = \bigcup_{i=1}^{n} \mathcal{A}_i$ with $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ if $i \neq j$. Then a presentation P of $\mathcal{A}$ is said to be* **as clustered** *with respect to $\{\mathcal{A}_i\}_{i=1}^{n}$ as another presentation Q if for all $\mathcal{A}_i$, the indices of $\mathcal{A}_i$ in P is as clustered as the indices of $\mathcal{A}_i$ in Q.*

Notice that this property is a partial order and but isn't necessarily total.

Lemma 1 *Let $G = \oplus_{\alpha \in \mathcal{A}} G_\alpha$ be a chromosome space. Suppose $\mathcal{A} = \bigcup_{i=1}^{n} \mathcal{A}_i$ with $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ if $i \neq j$, and that each $\mathcal{A}_i$ is a group of correlated genes. Then given a presentation P of $\mathcal{A}$ that is more clustered than a presentation Q of $\mathcal{A}$, we have that P on average converges to a solution more quickly than Q.*

To see this, notice that since P is more clustered than Q, crossovers in P will shorten correlation components less than crossovers in Q. And because the components are normal, they will on average contribute a higher score in P than in Q.

8 Results

Our code was written in a high-level interpreted language for quicker development and easier tweaking[1]. No particular emphasis was placed on optimizations, so timing figures are not necessarily indicative of performance. Instead, we shall look at how quickly our runs converged to an optimal solution. However, it should be noted that runs that used our hybrid algorithm took approximately 30-50% longer than the other runs where genetic presentations were not altered. Judging by the graphs, the time difference would nullify the quicker convergence of the new algorithm. However, there are two reasons why the time difference may be insignificant. First, as mentioned before, all code was written in a high-level interpreted language and therefore was highly unoptimized, particularly the crossover and mutation code. So timing figures do not accurately reflect the amount of computational work the two algorithms demand. Second, quicker convergence implies fewer evaluations of individuals are required for equal results. Since our fitness function was almost trivial and implemented quite efficiently, the amount of time saved by not having had to evalute more individuals was not significant, and had the fitness function been much more computationally expensive, then our new algorithm would require much less computational work for equal results.

We present a more general version of the majority problem from Section 5, noting that Lemma 1 applies. For simplicity, we have the positive integer variables $s = \textit{groupsize}$ and $p = \textit{numgroups}$. Then we have a binary chromosome of length $n = \textit{groupsize} \times \textit{numgroups}$, with individuals $g = (b_1, b_2, \ldots, b_n)$. The fitness function will be

$$f(g) = \overbrace{\text{Maj}(\underbrace{b_{i_1}, b_{i_2}, \ldots, b_{i_s}}_{\textit{groupsize}\text{ terms}}) + \text{Maj}(\underbrace{b_{i_{s+1}}, \ldots, b_{i_{2s}}}_{\textit{groupsize}\text{ terms}}) + \cdots + \text{Maj}(\underbrace{b_{i_{(p-1)s+1}}, \ldots, b_{i_n}}_{\textit{groupsize}\text{ terms}})}^{\textit{numgroups}\text{ terms}}$$

where $b_{i_1}, b_{i_2}, \ldots, b_{i_n}$ is some permutation of $b_1, b_2, \ldots, b_n$.

For various values of *groupsize* and *numgroups*, we ran our algorithm for *generations* generations and averaged our results over *trials* trials. The graphs below plot the mean normalized score of the best-of-generation individual. Since we use an elitist approach the scores are always increasing. Mutation occurred in two different ways. First, the actual genes were randomly mutated by toggling the gene, which was a bit. Second, in the presentation of genetic information, consecutive gene positions could be exchanged, e.g, $x_1\ x_2\ x_3\ x_4$ could become $x_1\ x_3\ x_2\ x_4$.

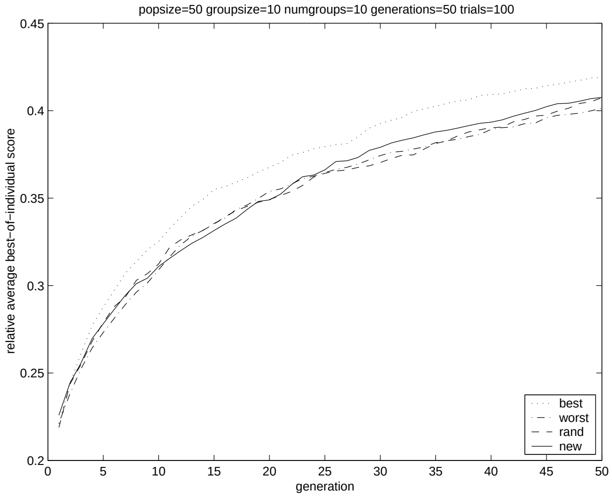
In total, we used four different algorithms for each set of test parameters. The algorithms `best`, `rand`, and `worst` were all simple fixed-presentation genetic algorithms with single-point crossover. The `best` algorithm used the ideal presentation of genetic information, namely $(b_{i_1}, b_{i_2}, \ldots, b_{i_n})$. The `rand` algorithm

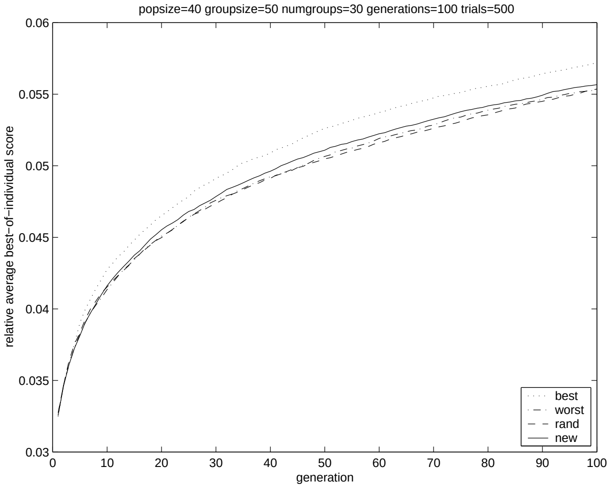
[1] All computations were performed with MATLAB 6.1.0.450 (R12.1) on a Sun Blade 2000 workstation with dual 900 MHz processors and 2GB RAM.

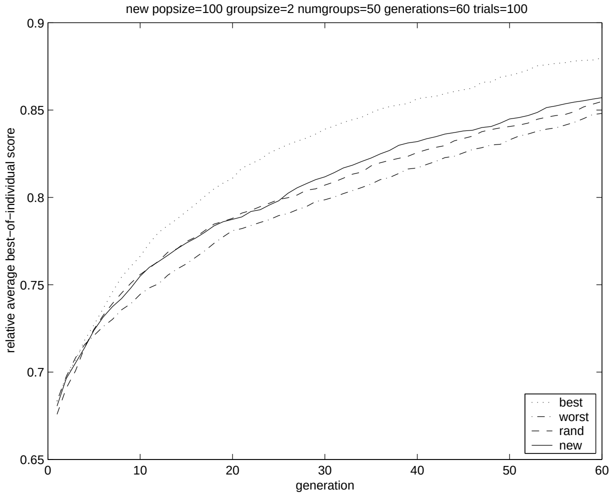
used a random (but fixed) permutation of $b_1, \ldots, b_n$ as its presentation. Finally, the `worst` algorithm uses a particularly bad presentation, though not necessarily the worst:

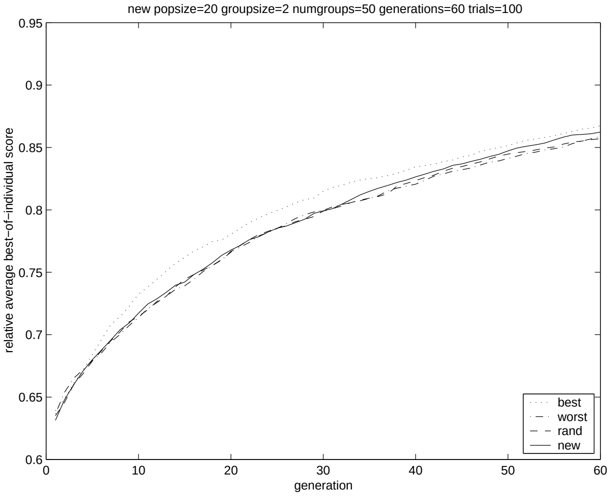
$$\begin{array}{cccccc} (& b_{i_1}, & b_{i_{s+1}}, & \ldots & b_{i_{(p-1)s+1}}, & \\ & b_{i_2}, & b_{i_{s+2}}, & \ldots & b_{i_{(p-1)s+2}}, & \\ & \vdots & \vdots & \ddots & \vdots & \\ & b_{i_s}, & b_{i_{2s}}, & \ldots & b_n &) \end{array}$$

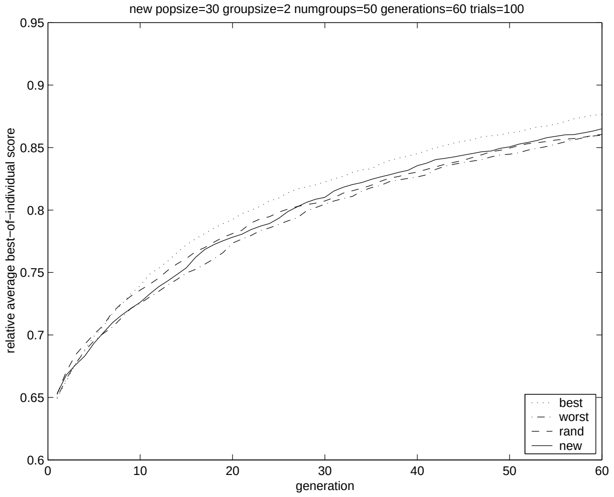
The `new` algorithm refers to our hybrid presentation-evolving algorithm with crossover and mutation as discussed.

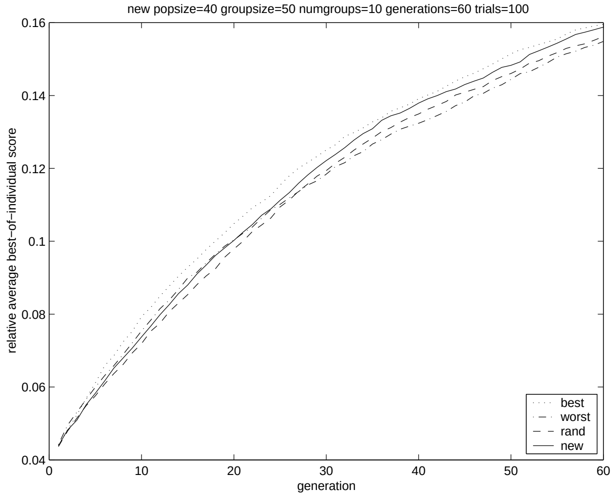


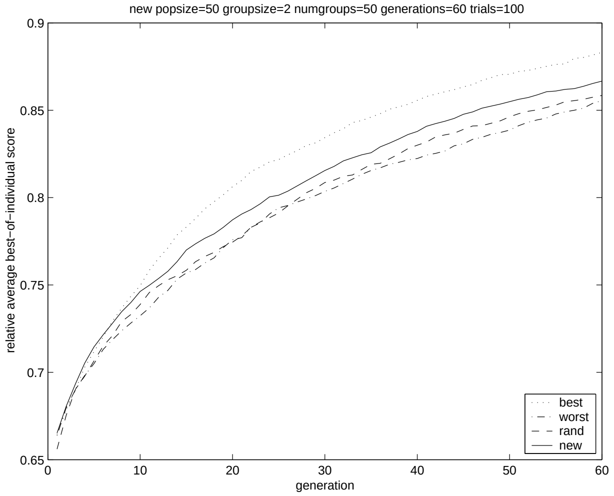


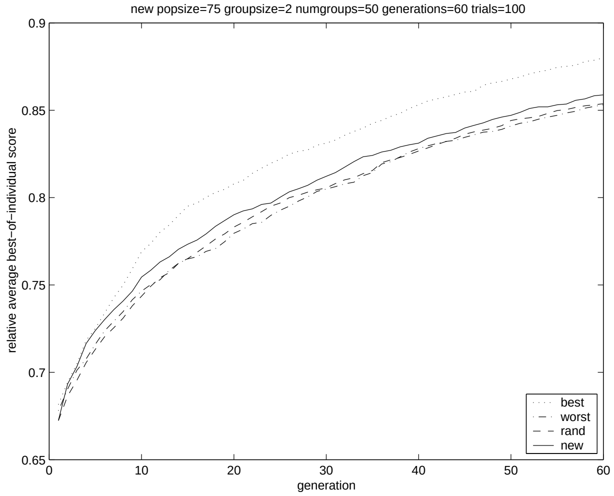












As expected, `best` outperforms `rand`, which in return outperforms `worst`. The `new` algorithm comes in between `best` and `rand`, indicating that it was successful in evolving a better-than-average presentation. In fact, in certain cases the `new` algorithm significantly outperforms `rand` and approaches the upper-bound `best`.

9 Conclusion

Initial results are certainly promising since our new algorithm is certainly extremely rough and far from ideal. In fact, almost all parts of the algorithm can be improved, but this first attempt is intended as a proof-of-concept. In particular, the crossover of two permutations as described in Section 6 is rather arbitrary. There are various other methods of crossing two permutations, and they correspond loosely to methods of crossover of itineraries for the traveling salesman problem. For example, partially matched crossover, order crossover, and cycle crossover can all be substituted for our crossover method, which was constructed for its analytic simplicity.

However, we are still unnecessarily presenting genetic information in an inherently linear fashion and still using a single-point crossover. Suppose now that our gene spaces are intrinsically embedded as points on a torus in $\mathbb{R}^3$, meaning that the intrinsic distance between two gene spaces is approximated by the Euclidean distance. Then we would like to present the genome in this manner and not use the ideal linear presentation, which would simply be the projection of the points on the torus onto a line. Crossover, then, would simply consist of slicing the torus in two and recombining as per a simple crossover.

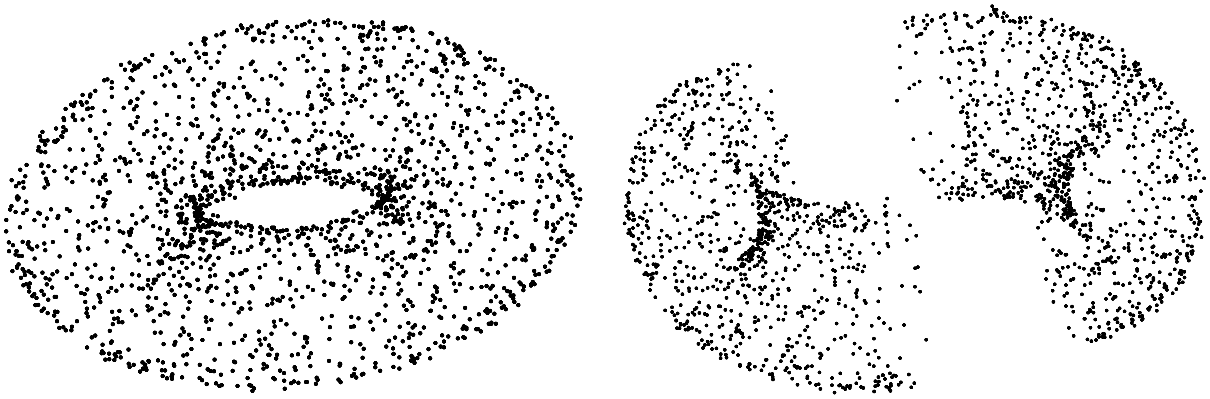


Figure 1: Intact torus with points representing gene spaces; possible crossover slice

It may be possible to approximate the intrinsic distance between any two gene spaces by examining the scores of two parents, the scores of their two children, and the distance between the two genes in their presentations. Given these distances, then it is possible to embed the gene spaces as points into a low-dimensional manifold in $\mathbb{R}^n$ using multidimensional scaling or similar techniques. The advantage of this embedding is that if the Euclidean distance in $\mathbb{R}^n$ approximates the intrinsic distance, then the slicing crossover operator as illustrated on the torus above would be very effective for two reasons: first, correlated genes would remain intact with high probability since they would be close in $\mathbb{R}^n$ and slicing would probably not separate them, and second, slicing would still be an effective method of exchanging genes and therefore superior combinations of genetic information, the precise goal of the crossover operator.

References

[1] S. Forrest and M. Mitchell. Relative Building-Block Fitness and the Building-Block Hypothesis. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 109–126. Morgan Kaufmann, San Mateo, California, 1993.

[2] D. Goldberg and K. Sastry. A practical schema theorem for genetic algorithm design and tuning.

[3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learing.* Addison Weslay, 1989.

[4] David E. Goldberg, Bradley Korb, and Kalyanmoy Deb. Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems*, 3(5):493–530, 1989.

[5] David E. Goldberg, Bradley Korb, and Kalyanmoy Deb. Erratum: "Messy genetic algorithms: motivation, analysis and first results" [Complex Systems **3** (1989), no. 5, 493–530]. *Complex Systems*, 5(1):101, 1991.

[6] John H. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, Mich., 1975. An introductory analysis with applications to biology, control, and artificial intelligence.

[7] Alden Wright. The exact schema theorem.