Fitness Distance Correlation Analysis: An Instructive Counterexample

Lee Altenberg
Hawaii Institute of Geophysics and Planetology
University of Hawaii at Manoa, Honolulu, HI USA 96822
WWW: http://pueo.mhpcc.edu/ ~altenber/

©1997 by Lee Altenberg. Expanded from the version in Baeck, T. ed., *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, Morgan Kaufmann, San Francisco, CA.

Fitness distance correlation (FDC) has been offered as a summary statistic with apparent success in predicting the performance of genetic algorithms for global optimization. Here, a counterexample to Hamming-distance based FDC is examined for what it reveals about how GAs work. The counterexample is a fitness function that is 'GA-easy' for global optimization, but which shows no relationship between fitness and Hamming distance from the global optimum. Fitness is a function that declines with the number of switches between 0 and 1 along the bitstring. The test function is 'GA-easy', in that a GA using only single-point crossover can find the global optimum with a sample on the order of 10^{-3} to 10^{-9} of the points in the search space, an efficiency which increases with the size of the search space. This result confirms the suspicion that predictors for genetic algorithm performance are vulnerable if they are based on arbitrary properties of the search space, and not the actual dynamics of the genetic algorithm. The test function's solvability by a GA is accurately predicted, however, by another property—its evolvability, the probability that the genetic operator produces offspring that are fitter than their parents. It is also accurately predicted by FDC that uses a distance measure defined by the crossover operator itself, instead of Hamming distance. Mutation-based distance measures are also investigated, and are found to predict the GA's performance when mutation is the only genetic operator acting. A comparison is made between Hamming-distance based FDC analysis, crossover-distance based FDC analysis, evolvability analysis, and other methods of predicting GA performance.

1 Introduction

A still unsolved problem in evolutionary computation is to understand when a genetic algorithm (GA) will be effective at finding the global optimum of a search space. To understand, of course, means more than simply to run the GA in question and see how well it finds the global optimum. Rather, it means finding properties of the GA:

- that predict its behavior and require less computation than the actual search, or
- which are invertible—i.e. which can be used to *construct* a GA that has good performance.

Jones (1995) and Jones and Forrest (1995) have proposed 'fitness distance correlation' (FDC) as a candidate property for predicting the performance of a genetic algorithm in global optimization. In this approach, the Hamming distances between sets of bitstrings and the global optimum bitstring are compared with their fitness. Large negative correlations between Hamming distance and fitness are taken to be indicators that the system is easy to optimize with a GA. Large positive correlations indicate the problem is 'misleading' and selection will guide the population away from the global maximum. Near-zero correlations indicate that the GA does not have guidance toward or away from the optimum, and thus faces the same difficulties as random search. However, when the correlation coefficient is near zero, Jones finds that it is too simplistic a summary statistic and can be fooled. This necessitates a closer examination of the relation between fitness and distance, through the use of a scatter plot of fitness versus Hamming distance.

Jones states that FDC (comprising the use of the correlation coefficient and the fitness-distance scatter plot) is able to:

1. predict GA behavior on a number of well-studied problems,
2. illuminate problems whose behavior had been seen as surprising using other analytical frameworks, and
3. account for the performance of different problem encodings and representations.

Jones notes that this is both 'encouraging and alarming'—encouraging since FDC appears to work, but alarming since 'distance' is defined without reference to the genetic operators, the representation of the search space, or any of the dynamics of the genetic algorithm (Jones, 1995). Jones recommends that a stronger predictor of GA optimization performance would be FDC analysis using a distance measure based on the genetic operators themselves. Nevertheless, Hamming-distance based FDC is the source of the results Jones reports, so it is these results that we must consider.

With FDC we have a method of analysis that appears to predict the performance of genetic algorithms even though it does not directly incorporate any part of the GA dynamics. What are the implications of such a result? We can only surmise that either Hamming distance *is* in fact related to GA dynamics, or that this relationship fortuitously holds among the test cases Jones examined, in which case there may be counterexamples for which it does not hold.

Addressing this question, Jones notes that Hamming distance is strongly related to the mutation operator in classical bitstring genetic algorithms. The number of times that the mutation operator must be applied to transform a given string to the global optimum is monotonic with Hamming distance. However, it has been classically argued that the main role of mutation in genetic algorithms is to prevent premature convergence, and that recombination is the operator most important for GA performance. Therefore, one interpretation of Jones's results would be that mutation is a much more important determinant of GA performance, either generally or in the specific examples he examined.

Another interpretation is that there is a deep relationship between Hamming distance and the recombination operator. Jones does not discuss this possibility. Recombination does not fit easily into an FDC framework, because it involves pairs of bitstrings, so distance cannot be defined simply between individual bitstrings. And the formation of pairs of bitstrings on which recombination operates depends on the distribution of bitstrings at each generation of the GA; recombination can thus be considered to be a frequency-dependent operator. Finally, recombination, including single-point crossover, can create offspring that are a great Hamming distance from their parents and from each other. Thus recombination would seem to destroy any relationship between Hamming distance and GA dynamics.

Even in the absence of a dynamical foundation for FDC, several investigators have been applying it to new domains (e.g see Collard and Escazut, 1996). But the apparent success of FDC on the wide suite of fitness functions remains an unanswered question.

To approach this problem, it should be noted that FDC analysis consists of several conjectures:

1. If there is a large positive fitness distance correlation ($r \geq 0.15$), then the problem is misleading and the GA will be led away from the global optimum;
2. If there is a large negative fitness distance correlation ($r \leq -0.15$), then the problem is straightforward and the GA will find the global optimum with relatively good performance;
3. If the fitness distance correlation is near zero ($-0.15 < r < 0.15$), the prediction is indeterminate:
   (a) If the fitness-distance scatter plot shows no relationship between fitness and Hamming distance, the problem is GA-difficult;
   (b) certain structures that appear in the scatter plot will indicate that the problem is straightforward, or misleading, as the case may be.

This paper focuses specifically on 3(a). I leave to another study the problem of whether counterexamples to FDC Conjectures 1. or 2. can be found—i.e a GA-hard problem with large negative FDC, or a GA-easy problem with large positive FDC. Here, I produce a counterexample to 3(a) consisting of a GA-easy fitness function that shows no relationship between Hamming distance and fitness. To produce this counterexample, I follow the suggestion of Jones and use the crossover operator itself to define 'distance'. This counterexample demonstrates that arbitrary heuristics about search spaces are vulnerable when they do not incorporate the mechanisms of search. Other predictors of GA optimization performance that do incorporate the GA dynamics will be examined, and these are found to accurately predict GA performance with this fitness function.

2 Constructing a Counterexample

The key to constructing a counterexample to FDC is to devise a 'distance' measure in terms of the recombination operator. Single-point crossover is examined here. Single-point crossover can transform any pair of complementary bitstrings into any other pair of complementary bitstrings through its repeated application. The number of crossovers needed to transform a complementary pair into the global optimum and its complement can thus be used to define a 'crossover distance'. To make a fitness function that is straightforward to optimize, we would like each bitstring to have a path to the global optimum (through repeated application of crossover) in which fitness is monotonically increasing. The expectation is that this will allow the genetic algorithm to produce the next fittest bitstring along this path using crossover, amplify this bitstring through selection, and subsequently produce the next fittest bitstring through crossover, and so forth. The number of crossovers it takes to reach the global optimum can serve as the 'distance' measure. With this definition of distance, the fitness function should produce a large negative fitness distance correlation coefficient.

How can such a fitness function be defined? A possibility is suggested by examining Figure 1, which shows a sequence

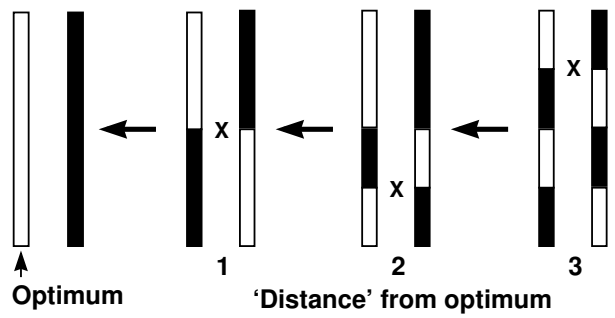

Figure 1: A path from a set of complementary bitstrings to the optimal bitstring via single-point crossover. The number of crossover events can be taken as the 'crossover distance' between bitstrings.

of crossover events on complementary bitstrings that produce a path to the optimal bitstring (set to be the bitstring of all 1s).

We notice that as one moves farther from the optimum along this path, the number of discontinuities between 0 and 1 increases by one with each step. So, we can let the number of discontinuities be the candidate measure of distance.

This definition of distance is somewhat artificial in terms of actual GA dynamics, because crossover on complementary pairs of bitstrings will be a relatively infrequent event in the population. But all that is required here is that this 'distance' will be sufficiently related to the GA dynamics that it can be used to produce a GA-easy fitness function that is a counterexample to FDC Conjecture 3(a).

In order for selection to guide the search along the path shown in Figure 1, the fitnesses need to be monotonically increasing along the path toward the optimum. A fitness function which decreases with the number of discontinuities between 0s and 1s will have that property. Furthermore, this fitness function would have these important properties:

1. Symmetry with respect to bit value, that is $f(\boldsymbol{x}) = f(\mathbf{1} - \boldsymbol{x})$, where $\boldsymbol{x} \in \mathcal{S} = \{0,1\}^L$ is a bitstring of length L, and $f(\boldsymbol{x})$ is the fitness of $\boldsymbol{x}$. Thus the fitness distance correlation coefficient will be zero.
2. The additive fitness components for individual bits will be zero.

Regarding property 1, the fitness distance correlation coefficient is:

$$r = C_{FD}/\sigma_F \sigma_D$$

where $C_{FD} = \mathrm{Cov}[f(\boldsymbol{x}), \mathcal{H}(\boldsymbol{x}, \hat{\boldsymbol{x}})]$, $\sigma_F = \mathrm{Var}[f(\boldsymbol{x})]^{1/2}$, $\sigma_D = \mathrm{Var}[\mathcal{H}(\boldsymbol{x}, \hat{\boldsymbol{x}})]^{1/2}$, $\mathcal{H}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is the Hamming distance, $\hat{\boldsymbol{x}} = \mathbf{1}$ is the global optimum, and the variances and covariances are taken over a the entire search space. $\mathrm{Cov}[f(\boldsymbol{x}), \mathcal{H}(\boldsymbol{x}, \hat{\boldsymbol{x}})] = 0$ by the symmetry conditions $f(\boldsymbol{x}) = f(\mathbf{1} - \boldsymbol{x})$ and $\mathcal{H}(\boldsymbol{x}, \hat{\boldsymbol{x}}) - \boldsymbol{E}[\mathcal{H}(\boldsymbol{x}, \hat{\boldsymbol{x}})] = \boldsymbol{E}[\mathcal{H}(\boldsymbol{x}, \hat{\boldsymbol{x}})] - \mathcal{H}(\mathbf{1} - \boldsymbol{x}, \hat{\boldsymbol{x}})$, hence $r = 0$.

Regarding property 2, the additive non-epistatic fitness component of any single bit can be derived from a decomposition of the fitness function into epistatic interactions between bits (Cockerham, 1954; Reeves and Wright, 1995). Bit-value symmetry immediately implies that all additive non-epistatic effects (i.e. $\boldsymbol{E}[f(* \ldots * x_k * \ldots *) - \boldsymbol{E}[f(* \ldots *)]$) will be zero, and all the non-zero fitness components will come from epistatic interactions between bits. In other words, at any bit position, a 0 bit will have the same average fitness as a 1 bit, and only schemata with more than one defining bit will have an average fitness different from the average for the entire search space.

Since the FDC coefficient is zero for this test function, the Hamming distance FDC analysis would predict that the fitness function will be hard for a GA to optimize if no special structure is revealed by the fitness-distance scatter plot. Therefore, as a counterexample, the fitness function needs to be shown to be GA-easy with no relationship between Hamming distance and fitness.

Jones (1995) finds cases that are GA-easy but which have a near zero FDC. In these cases, an examination of a scatter plot of fitness versus distance reveals structure in the fitness landscape that are consistent with it being GA-easy. For example, in the Liepins and Vose transformation (Liepins and Vose, 1991), the FDC coefficient is zero because half of the search space has an FDC value of 1, while the other half has -1, and this is revealed by an 'X' structure in the FD scatter plot.

A requirement for this counterexample, therefore, is that the FD scatter plot reveal no discernible structure between fitness and Hamming distance. The following formulation is used to achieve this.

2.1 The test function

First I will define the number of discontinuities between 0s and 1s in the bitstrings, $\boldsymbol{x} \in \{0,1\}^L$, where L is the length of the bitstring:

$$D(\boldsymbol{x}) = \sum_{i=1}^{L-1} |x_i - x_{i+1}| \ \in [0, L-1],$$

The Hamming distance, $\mathcal{H}(\boldsymbol{x}, \boldsymbol{y})$ between two bitstrings, $\boldsymbol{x}$ and $\boldsymbol{y}$, is

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{L} |x_i - y_i| \ \in [0, L].$$

The function $D(\boldsymbol{x})$ could be used directly in a fitness function such as $F(\boldsymbol{x}) = e^{-\alpha\, D(\boldsymbol{x})} \in (0,1]$. This is of the same form as the fitness function used by Prügel-Bennett and Shapiro (1994) in a GA analysis that uses a simple spin glass energy model.

However, for our purpose, $D(\boldsymbol{x})$ can not be used directly for the fitness function because it displays a strong relationship with Hamming distance, contrary to what is sought. The range of $D(\boldsymbol{x})$ is constrained by $D(\boldsymbol{x}) \leq 2\ \min[\mathcal{H}(\boldsymbol{x}, \mathbf{1}), \mathcal{H}(\boldsymbol{x}, \mathbf{0})]$, where $\mathbf{1}$ is the optimum bitstring. For example, any bitstring that has a single 1 and the rest of the bits 0, or vice versa, can have at most 2 discontinuities, and must have at least one. Numerical analysis shows that the expected number of discontinuities over all bitstrings a given distance H from the optimum is a simple function of H:

$$\boldsymbol{E}[D(\boldsymbol{x}) \mid \mathcal{H}(\boldsymbol{x}) = H] = 2\ \frac{(L-H)\ H}{L}, \qquad (1)$$

where for simplicity I will write $\mathcal{H}(\boldsymbol{x}, \mathbf{1}) = \mathcal{H}(\boldsymbol{x})$.

I will use equation 1 to normalize $D(\boldsymbol{x})$, and produce the following as the fitness function:

$$F(\boldsymbol{x}) = \max[1 - \frac{D(\boldsymbol{x})\ L}{2\ [L - H(\boldsymbol{x})]\ H(\boldsymbol{x})},\ 0], \qquad (2)$$

with one modification: in order that there be only one global maximum, I reduce the fitness of the all 0s bitstring to: $F(\mathbf{0}) = 1 - \frac{1}{2(L-2)}$. This will perturb the FDC coefficient to a still insignificant negative value, on the order of -2^{-L}.

What we expect from a genetic algorithm using crossover with this fitness function is that as evolution proceeds, bitstrings with longer and longer sections of contiguous 1s or contiguous 0s should predominate, and these contiguous segments should be recombined to form even longer segments. Finally, a bitstring with all 1s, the global optimum, should be produced by the population.

3 Results

To establish this counterexample it now remains to be shown that:

1. the test fitness function displays no relationship between fitness and Hamming distance from the optimum, and
2. the test function is easy for a genetic algorithm to optimize.

3.1 Does the test function show a fitness distance relationship?

The relationship between Hamming distance and fitness for the test function is shown in the fitness-distance scatter plot, Figure 2. The first graph in the figure shows randomly sampled bitstrings, while the second shows the bitstrings that are produced during a run of the genetic algorithm (see section 3.2 for GA implementation details).

Although the distribution of points is not uniform throughout the graph, it is difficult to discern any structure in which

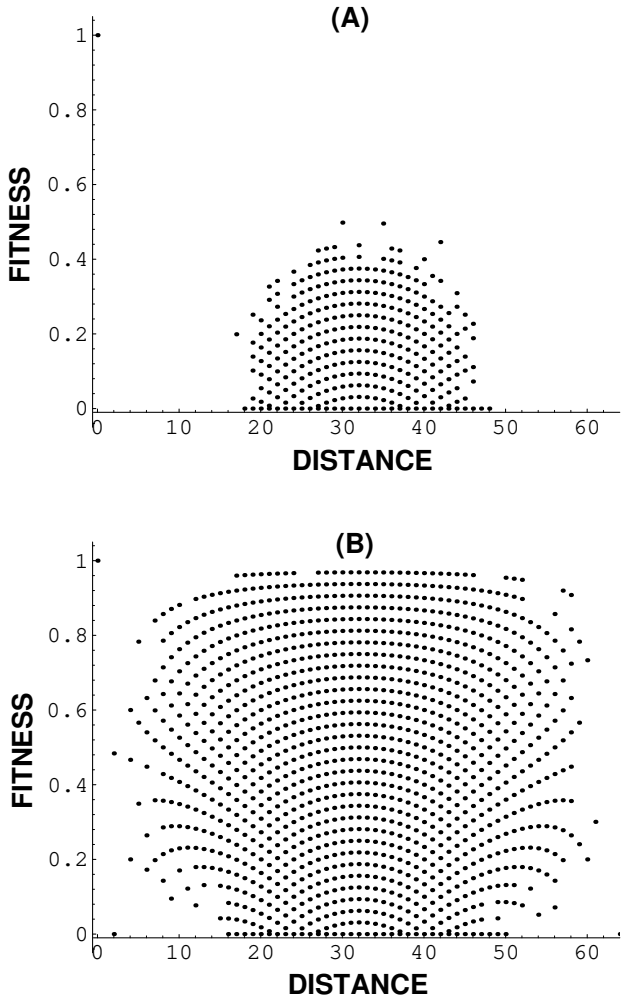

Figure 2: A scatter plot of the distribution of fitnesses vs. Hamming distance for the test function. $L = 64$ bits. (A) 40,000 randomly sampled bitstrings. (B) 500,000 samples taken during a GA run. The global optimum is shown at (0,1).

Table 1: Performance of the genetic algorithm on the test function. Columns are: the length of the bitstrings, the population size, the maximum number of fitness evaluations, the mean number of fitness evaluations among runs that found the global optimum, the portion of runs that found the global optimum, and the portion of the search space that was sampled in successful runs, which is approximated by $10^{3.53-0.25L}$.

Bits	Pop	Evals: Max	Evals: Mean	Success	Portion
24	2K	300K	56.3K	100%	3×10^{-3}
32	3K	300K	164K	93%	4×10^{-5}
50	10K	2,000K	10^6	41%	10^{-9}

Hamming distance could be a guide to the GA search. In particular, there are no points in the vicinity of the global optimum, $(0, 1)$. The next-fittest points are those of intermediate Hamming distance $\mathcal{H}(\boldsymbol{x}) = L/2$.

3.2 Is the test function GA-Easy?

The global optimum of the test function can be fairly easily found by a genetic algorithm. Results of some numerical simulations are given in Table 1. Additional simulations could refine these numbers, but they are adequate for the purposes addressed here.

Each of the entries are the result of 200 runs of the genetic algorithm. The GA is a 'simple genetic algorithm' that uses roulette wheel selection (so the fitness coefficients were in fact the values of the fitness function), discrete generations, random mating, and 100% probability of single-point crossover in the creation of offspring. No mutation was used in order to present a clear picture about the crossover operator.

As can be seen, as the size of the search space is increased, the percentage of runs that find the global optimum (Success %) decreases, and the mean number of bitstrings that need to be evaluated also increases. The reason that some runs fail to find the global optimum is premature convergence, in which one or more regions of the bitstring lose their polymorphism, with either 1s or 0s going toward fixation. However, as the bitstrings get longer a dramatic decrease can be seen in the portion of the search space that needs to be searched to find the global optimum. Over this range of bitstring lengths, the portion of the search space is closely approximated by the expression $10^{3.53-0.25L}$.

Although there is no standard definition for 'GA-easy', it is reasonable to use random search as the control for comparison. In random search, the expected number of points that need to be evaluated to sample the global optimum is half the size of the search space. The proportions of the search space sampled to find the global optimum, as shown in Table 1, are orders of magnitude smaller than that for random search, which should provide a satisfactory criterion for the test function being GA-easy.

In summary, this fitness function meets both requirements for it to be a counterexample to FDC Conjecture 3(a): the fitness function is GA-easy, and it shows no relationship to Hamming distance from the global optimum.

3.3 What predictors would show the test function to be GA-Easy?

FDC analysis using Hamming distance has been found to wrongly predict the difficulty of optimizing the test function studied here. But the question remains, what kinds of analysis would be better predictors? There are several candidates in the literature:

1. parent-offspring correlation, ρ_{op} (Weinberger, 1990; Manderick *et al.*, 1991);
2. evolvability (Rechenberg, 1973; Beyer, 1993; Altenberg, 1995);
3. simulated dynamics (Altenberg, 1995; Grefenstette, 1995);
4. analysis of variance for schemata (Radcliffe and Surry, 1995; Reeves and Wright, 1995); and
5. fitness distance correlation where distance is derived from the genetic operators (Jones, 1995; Jones and Forrest (1995)).

Some brief remarks on several of these methods will be made.

3.3.1 Parent-Offspring Correlation

Parent-offspring correlation analysis would predict that the test function presented here is GA-easy, because ρ_{op} using crossover will be high in this case. This is because $|D(\boldsymbol{x}) + D(\boldsymbol{y}) - [D(\boldsymbol{u}) + D(\boldsymbol{v})]| \leq 2$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the products of a crossover between parents $\boldsymbol{x}$ and $\boldsymbol{y}$. Thus, the expected offspring fitness cannot be far from the mean parental fitness.

However, a counterexample to the predictive ability of ρ_{op} has already been shown in Altenberg (1995). An incorrect prediction of poor GA performance can be made by ρ_{op} if the average offspring fitness does not increase with parental fitness, but the variance does, so that the chance of producing fitter offspring remains high as parental fitness increases. Thus one can have $\rho_{op} = 0$ and at the same time rapid evolution toward the global optimum. The use of ρ_{op} as a predictor of GA performance is thus secure only within certain constraints on the fitness function.

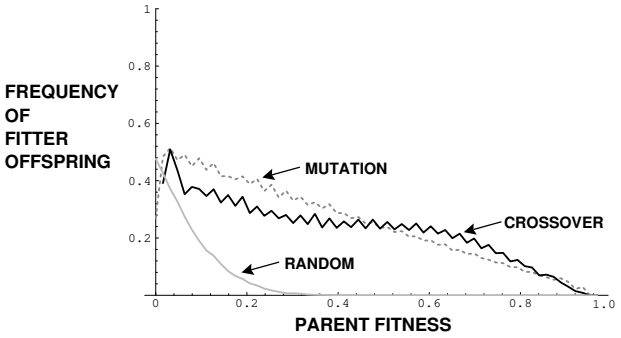

Figure 3: The evolvability of the population under mutation or crossover, with random search plotted for comparison. The probability that offspring are fitter than their parents is plotted as a function of parental fitness. With crossover, the X coordinate is the maximum of the two parents' fitnesses.

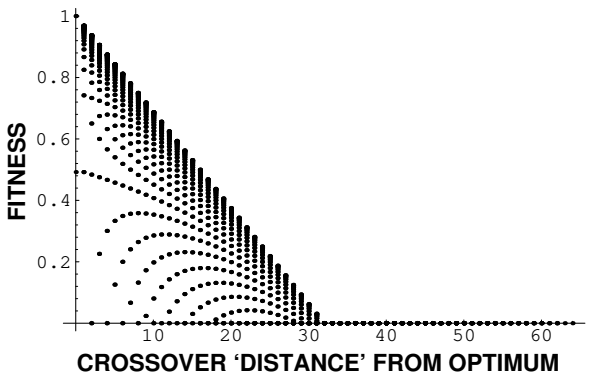

Figure 4: A plot of the test fitness function versus crossover distance—the number of discontinuities between 0s and 1s in the bitstring. The bitstring length is $L = 64$. Values for the entire search space are plotted.

3.3.2 Evolvability Analysis

The chance that a genetic operator produces offspring fitter than their parents is called 'probability of success' in the evolution strategy literature (Rechenberg, 1973; Beyer, 1993), and 'evolvability' by Altenberg (1995). An examination of evolvability for the test function is revealing.

Figure 3 shows numerical results for the proportion of offspring that are fitter than parents under the test function. Values are obtained in the actual runs of the GA, rather than from random sampling of bitstrings. This is consequential for crossover, since its product depends on the composition of the population. A tally is made of the number of times offspring are fitter than both parents. This is done by splitting the fitness values into 256 bins. When an offspring is fitter than either parent, the bin corresponding to the fittest parent has its tally increased by one. At the end of the GA run, the tallies in each bin are normalized by the total number of parental pairs corresponding to that bin, which yields the frequency data in Figure 3.

To compare crossover with mutation and random search, single-bit mutants are generated from each parent during the GA run and tallies made in the same way to give the frequency that mutation produces fitter offspring. These mutants are not included in the offspring populations, so mutation is not part of the GA dynamics. In the same fashion, the probability that random bitstrings are fitter than parents is tallied, which simply produces the fitness distribution for the entire search space (when the frequencies are subtracted from 1) .

In Figure 3 we see that crossover and mutation both maintain high evolvability throughout the range of fitness values. Both are roughly linear functions of fitness. For comparison, the evolvability value of random search rapidly vanishes as fitness increases, as expected.

Figure 3 suggests that the mutation operator should be as good or better than crossover at finding the global optimum of the test function, since its evolvability values remain on the same order as fitness increases, and actually exceed that of crossover for the highest fitnesses. However, a finer resolution magnification near fitness 1.0 would show that for the next-to-fittest bitstrings (strings with a run of $L/2$ zeros and a run of $L/2$ ones), mutation has no chance of producing the global optimum, while crossover has a positive chance that depends on the population composition. In simulations where the GA is run using mutation and no recombination, it converges on these next-to-fittest bitstrings, and never finds the global optimum of all 1s. This feature of the dynamics is thus predicted by the evolvability analysis.

3.3.3 Crossover-Based Fitness Distance Correlation Analysis

Jones (1995) suggests that a more accurate version of FDC might be developed if the genetic operators themselves could be used to define the distance from the optimum. Because recombination is a frequency-dependent operator, it can not be strictly mapped to a static distance metric. The dynamical effects of recombination depend inextricably on the composition of the population. Nevertheless, if we can relax the stringency of conditions we require of the crossover-based distance, a number of candidate measures can be proposed.

First is the simple implementation of the heuristic used in the derivation of the test function: the crossover distance is set to be $D(\boldsymbol{x})$, the number of discontinuities between 0s and 1s in the bitstring $\boldsymbol{x}$. This value corresponds to the minimum number of crossovers required to transform a bitstring into the optimal bitstring. Using this distance measure, an FDC scatter plot is shown in Figure 4.

This crossover-based FDC scatter plot shows a clear negative FDC coefficient for the test function, and predicts

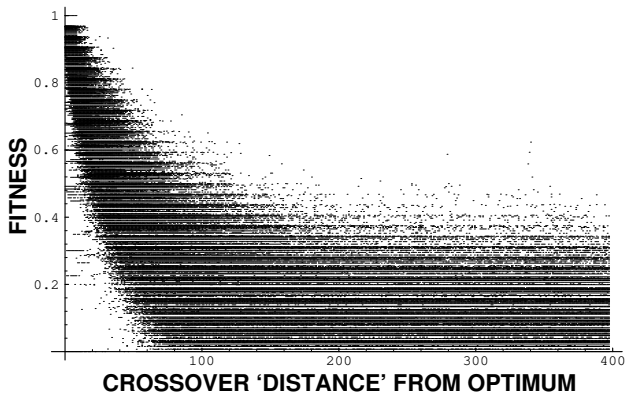

Figure 5: A plot of the test fitness function versus crossover distance—the number of generations of single-point crossover starting with the global optimum and its complement. The initial population contains the recombinant pairs of offspring from single crossovers between the optimum and its complement. Subsequent generations are made by replacing the population with crossover recombinants from randomly mated parents. This is plotted for 398 generations. Population size is 1500, bitstring length is $L = 64$.

accurately that it will be straightforward to optimize using crossover.

A second crossover-based distance measure is derived by applying crossover in reverse. We wish to give a distance value of 1 to all the bitstrings that can produce the global optimum through a single application of crossover. A distance value of 2 is given to the all the bitstrings that can produce the distance 1 bitstrings, and so forth, until the entire search space is labeled. The first set of bitstrings can be generated by applying crossover to the global optimum and its complement. The validity of this reverse application of crossover depends on the fact that crossover is symmetric in the transition probabilities between pairs of parents and pairs of recombinants. [1]

In this second method of generating a crossover-based distance, a population is initialized with the pairs of recombinant offspring from single crossovers between the global optimum bitstring and its complement. This population is randomly mated to produce new parental pairs, each of which are replaced by the pairs of their recombinant offspring to generate the next generation. This new generation will be labeled with a distance value of 2. This process is iterated, and bitstrings that are further and further away from the global optimum are generated. The generation number is used as the 'crossover distance' value.

[1] It should be noted that the crossover-based distances investigated here are specific to single-point crossover. In the case of free recombination ('uniform crossover'), the entire search space would have to be considered to be one recombination-distance unit away from the global optimum, because a single recombination event can yield any given bitstring if the correct bits are transmitted from complementary parents. Different genetic operators my thus present challenges for defining operator-based distances.

Table 2: Performance of the mutation-selection genetic algorithm on the test function. Columns are: the length of the bitstrings, the mean number of fitness evaluations among runs that found a sub-optimum, the portion of runs that found a sub-optimum, and the portion of the search space that was sampled in successful runs, which is approximated by $10^{3.36-0.24L}$. Mutation probability was 0.2, population size was 200, the maximum number of fitness evaluations was $2,000,000$, and 200 runs were made for each entry.

Bits	Mean Evals	Success	Portion
24	67.5K	100%	4.0×10^{-3}
32	325K	99.5%	7.6×10^{-5}
40	705K	82.5%	6.4×10^{-7}

In this algorithm, a small proportion of offspring will be produced which have already appeared in an earlier generation, and should rightly be excluded from the population. However, this contamination by lower distance bitstrings should be negligible for very large search spaces and population sizes that are large enough to make it unlikely that identical parents mate.

A fitness-distance scatter plot can be produced from these iterations, and is shown in Figure 5. This plot shows a clear negative correlation between fitness and crossover distance for the first 100 or so generations. After a certain point, the population has reached a quasi-equilibrium, and the distribution of fitnesses does not change with further generations. Under FDC Conjecture 1., this scatter plot correctly predicts that the test fitness function should be straightforward for the GA to optimize.

Therefore, this test function produces support for FDC Conjecture 1 under two different crossover-based distance measures, while it produces a contradiction to FDC Conjecture 3(a) under the Hamming distance measure.

3.3.4 Mutation-Based Fitness Distance Correlation Analysis

Another test of these two operator-based distance measures is afforded by the behavior of mutation-selection genetic algorithms (with no recombination) on the test function. Under mutation as the sole genetic operator, it is difficult for the GA to find the global optimum of the test function, but easy for it to find one of the near-optimum bitstrings, one with only one or a few discontinuities between 0s and 1s. Table 2 shows the success of the mutation-selection GA in finding the next-to-fittest sub-optima, a bitstring $(\ldots 00001111\ldots)$ or $(\ldots 11110000\ldots)$ of $L/2$ zeros and $L/2$ ones, for a variety of bitstring lengths. Its

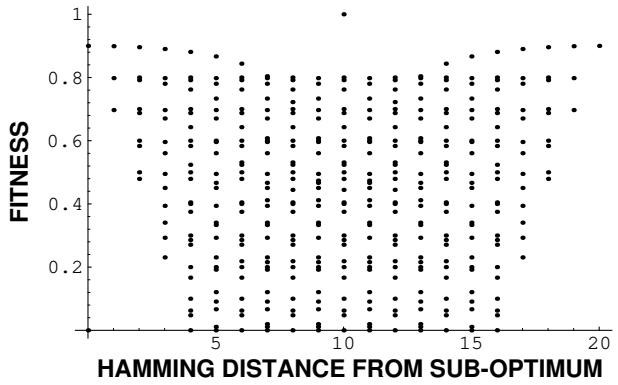

Figure 6: A plot of the test fitness function versus Hamming distance to the sub-optimal bitstring $(\ldots 00001111\ldots)$. Bitstring length is $L = 20$.

performance in finding the sub-optima is comparable to the performance of the recombination-only GA in finding the global optimum.

Can the two operator-based distance measures developed in Section 3.3.3 predict this behavior of the mutation operator on the test function?

The first operator-based distance measure in Section 3.3.3 is the minimum number of applications of the operator needed to transform one bitstring into another. In the case of single bit-flip mutation, this is simply the Hamming distance. Therefore, the mutational equivalent to the crossover distance plot of Figure 4 is already familiar, namely Figure 2, the Hamming-based scatter plot. However, Figure 2 was made using the global optimum as the reference point for the distance values.

Suppose that one of the sub-optimal bitstrings is used as the reference point for the Hamming distance measure? In this case, we obtain the fitness-distance scatter plot shown in Figure 6. The fitness distance correlation coefficient is zero, because the plot is symmetrical about the Hamming distance of $L/2$. However, it displays interesting structures in the scatter plot. Above fitness values of 0.8, there is a path of monotonically increasing fitnesses from points a Hamming distance of 7 away from the sub-optimum, and a mirror image path going to the complementary sub-optimum. The plot does not reveal whether mutation can move from one point on the path to the next, but in fact it can; the points are the bitstrings with a single discontinuity between 0s and 1s. In the numerical simulations of the mutation-selection GA, these paths comprised the typical sequence in which the sub-optimal bitstrings evolved. So Figure 6 suggests that the the sub-optimal bitstrings may be straightforward to find with the mutation-selection GA, and Figure 2 suggests that the global optimum will be difficult to find with the mutation-selection GA. These predictions are in accord with the results from running the GA.

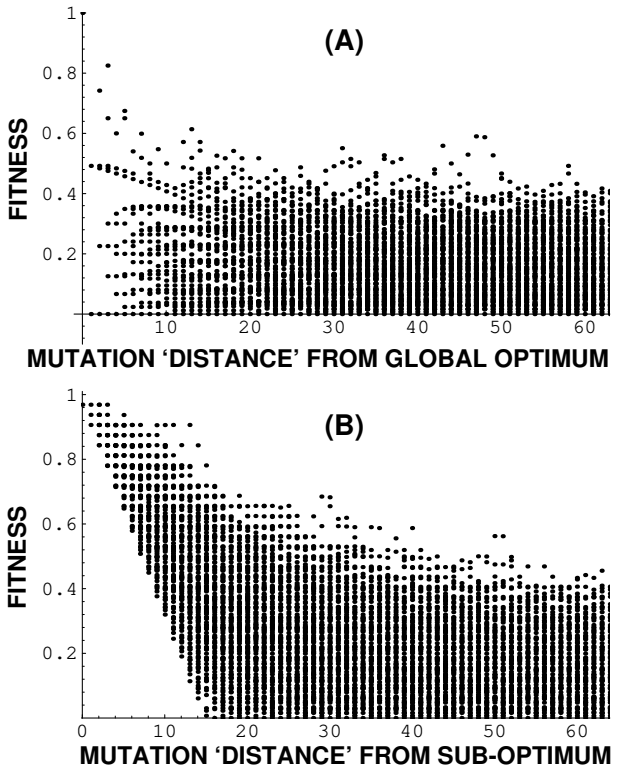

Figure 7: A plot of the test fitness function versus mutation distance—the number of generations of single bit-flip mutation. (A) Starting with the global optimum. (B) Starting with the sub-optimum genotype, a run of 32 zeros and 32 ones. Population size is 3,000, bitstring length is $L = 64$.

The second operator-based distance measure in Section 3.3.3 consists of applying the genetic operator iteratively to a population in reverse, beginning with the global optimum and evolving the bitstrings that would produce it. In the case of recombination, the validity of this reverse application of the operator relied on the symmetry in transition probabilities between parents and recombinant offspring. In the case of single bit-flip mutation, this symmetry also exists between parent and mutant offspring. Using mutation, two analogs to Figure 5 will be obtained: the first will use an initial population composed of the global optimum bitstring; the second will use an initial population composed of one of the sub-optimal bitstrings. Each iteration consists of replacing the population with single-bit mutants of its bitstrings. The fitnesses of the mutant genotypes are plotted versus the generation in which they occur. The results are shown in Figure 7.

The two plots (A) and (B) are revealing. When the population begins with the sub-optimum bitstring (B), the distribution of fitnesses moves steadily downward as the number of generations away from the sub-optimum grows. Figure 7(B) is reminiscent of Figure 5.

In contrast, when the global optimum bitstring is used to create the initial population (A), the distribution of fitnesses follows an uneven course. The fitnesses drop from 1.0 to 0 and 0.492 as one moves a single mutation away from the global optimum. At two mutations away, mutants with fitness 0.74 appear. At three mutations away, bitstrings with fitness 0.83 appear. Therefore, if one begins with a highly fit bitstring, the mutation operator cannot produce a path to the global optimum that has monotonically increasing fitnesses. During the run of the mutation-selection genetic algorithm, if the mean fitness of the population has evolved to greater than 0.492, then the bitstrings that are one mutation away from the global optimum will be unable to increase in frequency when they appear in the population. Thus Figure 7(A) can be interpreted as showing that the global optimum is surrounded by an 'adaptive valley' with respect to the mutation operator under the test fitness function.

These two mutation-based distance measures produce scatter plots that distinguish the global optimum from the sub-optimum bitstrings. Although the plots are not completely unambiguous in what they reveal, they are consistent with the prediction that a mutation-selection genetic algorithm (with no recombination) will find the sub-optimal bitstrings easily, but not the global optimum bitstring with this test function. Therefore, for both the mutation and crossover operators, the cases examined support the method of operator-based fitness distance correlation analysis.

4 Discussion

A fitness function is derived here that serves as a counterexample to one of the conjectures of fitness distance correlation analysis. The counterexample is derived by using the genetic operator itself—single-point crossover— to define the distance measure between bitstrings and the global optimum. The fitness function that is devised has zero fitness distance correlation, and shows no relationship between Hamming distance and fitness in a scatter plot, indicating that it should be difficult for a genetic algorithm to optimize under the FDC conjecture.

Contrary to the expectations of FDC analysis, the fitness function is found to be easily optimized by a GA using single-point crossover and roulette wheel selection, and the efficiency of the GA (as measured by the proportion of the search space sampled during the search before finding the global optimum) increases with the size of the search space.

The failure of Hamming-distance based FDC analysis on this test function poses the question of whether other candidate methods of GA analysis could do better. One of these methods, evolvability analysis, is investigated. The test function is revealed to exhibit high evolvability throughout its range of fitnesses, which is concordant with the test function being easy for a GA to optimize. Evolvability analysis also predicts that if mutation is the sole genetic operator, it should yield high fitnesses, yet be unable to find the global optimum. This prediction is borne out.

But simple evolvability analysis clearly has its shortcomings as well. The long path problem (Horn, Goldberg, and Deb, 1994) offers a good counterexample. Even though the chance of parents producing fitter offspring through mutation remains high as fitnesses increase, single-bit mutation in the long path problem can be inefficient at finding the optimum because the path it travels to the optimum can be quite long.

In the evolution strategy literature (Rechenberg, 1973; Beyer, 1993), operators that produce the highest rates of progress toward the global optimum do not generically produce the highest evolvability values. Rechenberg (1973), in the formulation of his '$1/5$ rule', found that the maximum rate of progress for a broad class of cases occurs for evolvability values near $1/5$. When the search space is $\Re^n$ and the fitness function is continuous, the Gaussian mutation operator with small amplitude will produce evolvability values near $1/2$. Larger amplitudes produce lower evolvability values, but larger fitness increases when they do occur, so there is an intermediate optimum. This tradeoff has been explored in variety of cases by Fogel and Ghozeil (1996), Therefore, a refinement of evolvability analysis needs to take into account the *amount* of fitness increase that occurs in an offspring, not just that it occurs.

Evolvability analysis is also no simpler computationally than running the genetic algorithm itself, because it de-

pends on the GA to find the sample points of high fitness that it uses in the analysis. Evolvability is potentially invertible, however. Which is to say, it has the potential to be used in constructing well-performing genetic algorithms. It suggests that one should choose, wherever feasible, to use genetic operators and representations that maintain a high likelihood of producing fitter variants as the fitness of parents increases.

Jones's (1995) suggestion that an operator-based distance would produce a more sound foundation for fitness distance correlation analysis is investigated. Two different distance measures based on single-point crossover are defined, and fitness-distance scatter plots are produced using these distance measures. With these measures, FDC correctly predicts that the test function will be straightforward for a GA to optimize. Two different distance measures based on single bit-flip mutation are also defined, and fitness-distance scatter plots are produced using these distance measures. With these measures, FDC is concordant with the result that a GA using mutation as its only genetic operator will find the sub-optimal bitstrings, but not the global optimum. Hence Jones's conjecture about the greater soundness of operator-based distances is supported.

This study confirms Jones's (1995) concerns regarding the apparent lack of connection between the actual dynamics of the genetic algorithm and Hamming-distance based FDC analysis. A counterexample is derived to one of the FDC conjectures, the case of zero fitness distance correlation. However, the question remains as to whether counterexamples can be found to Hamming-distance based FDC in the cases of large negative or positive FDC coefficients. Work on this question could reveal whether a deeper connection exists between Hamming distance and recombination in genetic algorithm dynamics.

Acknowledgements

This research was supported in part by the Santa Fe Institute and the Maui High Performance Computing Center, which have graciously hosted me as a visiting researcher. I thank the Santa Fe Institute for creating the research community which facilitated my investigation of this topic. Thanks to Terry Jones and Melanie Mitchell for their helpful comments.

References

Altenberg, L. 1995. The Schema Theorem and Price's Theorem. In D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 23–49. Morgan Kaufmann, San Mateo, CA.

Beyer, H.-G. 1993. Toward a theory of evolution strategies: Some asymptotical results from the $(1,^{+}\lambda)$-theory. *Evolutionary Computation* 1(2): 165–188.

Cockerham, C. C. 1954. An extension of the concept of partitioning hereditary variance for analysis of covariances among relatives when epistasis is present. *Genetics* 39: 859–882.

Collard, P. and C. Escazut. 1996. Fitness distance correlation in a dual genetic algorithm. In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI-96)*, pages 218–222, Budapest, Hungary. John Wiley and Sons.

Fogel, D. and A. Ghozeil. 1996. Using fitness distributions to design more efficient evolutionary computations. In *IEEE International Conference on Evolutionary Computation*, pages 11–19, New York. IEEE Press.

Grefenstette, J. J. 1995. Predictive models using fitness distribution of genetic operators. In D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 139–161. Morgan Kaufmann, San Mateo, CA.

Horn, J., D. E. Goldberg, and K. Deb. 1994. Long path problems. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature—PPSN III*, volume 866, Berlin. Springer-Verlag.

Jones, T. *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, March 1995.

Jones, T. and S. Forrest. 1995. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192, San Francisco, CA. Morgan Kaufmann. Santa Fe Institute Working Paper 95-02-022. URL: http://www.santafe.edu/sfi/publications/Working-Papers/95-02-022.ps.

Liepins, G. and M. Vose. 1991. Deceptiveness and genetic algorithm dynamics. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 36–50, San Mateo, CA. Morgan Kaufmann.

Manderick, B., M. de Weger, and P. Spiessens. 1991. The genetic algorithm and the structure of the fitness landscape. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150, San Mateo, CA. Morgan Kaufmann Publishers.

Prügel-Bennett, A. and J. L. Shapiro. 1994. Analysis of genetic algorithms using statistical mechanics. *Physical Review Letters* 72(9): 1305–1309. February 28.

Radcliffe, N. J. and P. D. Surry. 1995. Fitness variance of formae and performance prediction. In D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*. Morgan Kaufmann, San Mateo, CA.

Rechenberg, I. 1973. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart.

Reeves, C. and C. Wright. 1995. An experimental design perspective on genetic algorithms. In D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*. Morgan Kaufmann, San Mateo, CA.

Weinberger, E. D. 1990. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics* 63: 325–336.