

See discussions, stats, and author profiles for this publication at: <http://www.researchgate.net/publication/220885629>

Biases in the Crossover Landscape.

CONFERENCE PAPER · JANUARY 1989

Source: DBLP

CITATIONS

269

3 AUTHORS, INCLUDING:



[Larry J. Eshelman](#)

Philips

21 PUBLICATIONS 2,231 CITATIONS

[SEE PROFILE](#)



[J. David Schaffer](#)

Binghamton University

67 PUBLICATIONS 5,043 CITATIONS

[SEE PROFILE](#)

PROCEEDINGS OF THE
THIRD INTERNATIONAL
CONFERENCE ON

GENETIC ALGORITHMS

George Mason University

June 4-7, 1989

Editor/Program Chair: J. David Schaffer

With support from:

Navy Center for Applied Research in Artificial Intelligence,
Naval Research Laboratory
Philips Laboratories, North American Philips Corporation

Conference Committee:

Kenneth A. De Jong, George Mason University (Conference Chair)
John J. Grefenstette, Naval Research Laboratory (Finance)
Lashon B. Booker, Naval Research Laboratory (Local Arrangements)
David E. Goldberg, University of Alabama (Publicity)
J. David Schaffer, Philips Laboratories (Program)
Lawrence Davis, Bolt, Beranek and Newman, Inc.
John H. Holland, University of Michigan
George G. Robertson, Xerox PARC
Stephen F. Smith, Carnegie-Mellon University
Stewart W. Wilson, Rowland Institute for Science

Morgan Kaufmann Publishers, Inc.
San Mateo, California

Biases in the Crossover Landscape

Larry J. Eshelman
Richard A. Caruana
J. David Schaffer

Philips Laboratories, North American Philips Corporation, 345 Scarborough Road, Briarcliff

Manor, New York 10510

Abstract

Effective search often makes use of biases to help direct the search. In this paper we sketch a bias landscape for the crossover operator used in genetic search. We identify two types of biases — positional bias and distributional bias — which we use to classify various alternative crossover operators. The one-point traditional crossover operator is characterized by high positional bias and low distributional bias. We provide experimental evidence suggesting that this may be a poor combination. In fact, most of the other crossover operators described in our study do consistently better than one-point traditional crossover.

1. Introduction

Inductive learning algorithms use bias in order to make learning more effective [13]. For example, biasing search towards certain regions of the search space is an important means of enabling efficient learning from a manageable set of examples. In the case of genetic algorithms (GA), trials are allocated to clusters of genes based on their observed fitness. The GA's general bias, as expressed in the *schema sampling theorem* [6], implies that an exponentially increasing number of trials will be allocated to sets of bits (schemas) occurring in the better performing individuals. This general bias is implemented by selection according to fitness and the crossover and mutation operators, which provide the specific biases that affect the performance of the GA. Some of the biases associated with these operations are well known and are necessary in order to direct search. Others, however, are not so well understood and, unless properly exploited, can hinder search. In this paper, we discuss two types of biases associated with crossover that fall into this latter category: (1) positional bias and (2)

distributional bias. We suggest that one way to understand the behavior of alternative crossover operators is in terms of these biases. We illustrate this by analyzing several different crossover operators.

2. Background

The GA combines the exploitation of past results with the exploration of new areas of the search space. The effectiveness of the GA depends upon an appropriate mix of exploration and exploitation. Selection according to fitness is the source of exploitation. Several researchers have studied mechanisms that affect selection bias. Baker [2] and Schaffer [7] studied how different selection mechanisms bias selection. Grefenstette showed the effect of different scaling mechanisms on selection pressure [5]. Whitley and Kauth introduced a parameter for directly controlling selection pressure [14]. Sirag and Weisser introduced a thermodynamic operator with a global temperature parameter for controlling convergence [12].

The mutation and crossover operators are the sources of exploration. In order to explore they must disrupt some of the schemata on which they operate. The tradeoff of exploration and exploitation is clearest with mutation. As the mutation rate is increased, mutation becomes more disruptive until the exploitative effects of selection are completely overwhelmed. Since mutation is a simple operator, flipping bits at random with a uniform distribution along the string, it would seem that there is not much room for bias. Caruana and Schaffer have shown, however, that a hidden bias is introduced by the binary coding of parameters and that it can be eliminated by Gray coding [3].

Crossover, in contrast to mutation, provides a much subtler form of exploration, and so is an even more fertile ground for unexpected biases.

Crossover, like mutation, explores the search space by changing the value of some of the bits in a string. Unlike mutation, however, changes in the chromosome produced by crossover are constrained to those values that have been shown to be viable in so far as they have survived the selection process. Crossover is, in effect, a method for sharing information between two successful individuals. Although the amount of exploration provided by crossover will be affected by the crossover rate, it will also be affected by selection. As selection pressure is increased, variability is decreased in the population, and consequently there are fewer differences that crossover can explore by recombination. To the extent that the population has converged, there is less information that can be shared between individuals. In short, crossover is severely affected by the exploration-exploitation tradeoff. The amount of exploration performed by crossover is limited by the amount of exploitation performed by selection. Increased exploitation by selection leads to decreased exploration by crossover.

In this paper we examine how the specific method used by crossover for selecting material (bits) to be exchanged affects exploration. Throughout this paper we assume that two parents are selected according to fitness and material between them is exchanged to produce two children which replace them. (Other mating schemes will require re-analysis of the biases.) We categorize different crossover methods according to two kinds of biases that affect the exploratory power of crossover: (1) positional bias and (2) distributional bias. A crossover operator has positional bias to the extent that the creation of any new schema by recombining existing schemata is dependent upon the location of the bits in the chromosome. A crossover operator has distributional bias to the extent that the amount of material that is expected to be exchanged is distributed around some value or values as opposed to uniformly distributed ranging from 1 to $L-1$ bits.

One-point traditional crossover has a very strong positional bias, but no distributional bias. A consequence of its positional bias is that schemata with short defining length are less likely to be disrupted than schemata with long defining length. A consequence of its lack of distributional bias is that the expected number of schemata disrupted by crossover is relatively low. There is no reason, however, to believe that this is an optimum combination. In fact, the experimental evidence we present later in this paper indicates that the

combination of biases represented by one-point traditional crossover is far from optimal. In the next two sections we discuss in detail the two biases that we have identified. Then we describe how the various alternative crossover operators fall within this landscape. Finally, we present some experimental evidence with regard to the performance of these operators.

3. Positional Bias

One-point traditional crossover operates by (1) randomly pairing two individuals, (2) selecting a random position along the string with a uniform distribution, and (3) then exchanging the segments to the right of this position. A side effect of this operation is that interacting bits that are relatively far apart on the string are more likely to be disrupted (separated) by crossover than bits that are relatively close together. Conversely, non-interacting bits that are close together on the string are more likely to be treated by crossover as related components (i.e., not separated) than bits that are far apart. It is recognized that one-point traditional crossover's positional bias, like most biases, can be detrimental if not properly exploited, but it has always been assumed that there are ways of exploiting it. In particular, genes that are thought to interact should be placed near each other on the chromosome.

There are, however, problems with this solution. First, it may not be known which bits are related and which are not. In fact, the GA is most useful for large, complex, poorly understood search spaces where there is little or no a priori knowledge about which bits interact. Second, a linear string may not allow all interacting bits to be placed close to each other if some of them also interact with other bits. Third, the task is not simply to place interacting bits close to each other, but also to place non-interacting bits as far apart as possible. The first problem might be solved by letting the GA decide which bits are related and which are not. This could be done by including the inversion operator [6]. Unfortunately, there is often no way to position bits on a one-dimensional string so that their relative locations reflect their degree of expected interaction even when this is known. In other words, the positive effect of one-point traditional crossover's sampling bias in favor of short schemata may be offset by two undesirable side-effects. First, interacting clusters of bits that are far apart on the chromosome are less likely to propagate together. Second, short clusters of non-interacting genes (i.e., not causally related to good

performance, but occurring by chance in better individuals) are less likely to be disrupted, contributing to premature convergence of the gene pool to suboptimal individuals. The first effect is well known, but the latter is not. We call this later side-effect *spurious correlation*. Given the large number of schemata processed by the GA, some spurious correlations are inevitable. One way to enable the GA to avoid being misled by spurious correlations is to soften selection pressure. An alternative technique is to increase and to more uniformly distribute the disruptive effects of crossover.

Another way of understanding the impact of positional bias is to analyze its effect on the explorative power of crossover. One measure of the explorative power of a particular crossover operator is the portion of the points in the search space that are potentially reachable by a single crossover step. Of course, this will depend upon the diversity of the two parents. Assume that the parents are maximally diverse, i.e., they are complements of each other. Then, in principle, the two parents can be cut and spliced in one way or another to produce every possible point in the search space — all 2^L points where L is the length of the string (assuming binary alleles). The upper limit on the explorative power of a crossover operator would be the ability to potentially reach any of the 2^L points in a single crossover operation. One-point traditional crossover, for instance, can potentially reach any one of only $L-1$ points in the search space in a single crossover event. It is limited to $L-1$ crossover points and so can produce only $L-1$ recombinations from the two parents. This is not to say that one-point traditional crossover is limited to searching a small portion of the search space. In principle, any point in the search space can be reached with repeated crossovers, but those points that can only be reached by a long sequence of operations will have a much lower probability of being searched than those that can be reached with a short sequence of operations. In other words, crossover will be biased in favor of those points in the search space that are more readily reachable. This means that a GA using one-point traditional crossover is more likely to sample some points repeatedly while rarely sampling others. The stronger the selection pressure, the stronger this effect will be, but it will occur even if there is no selection taking place at all.

In order to increase the number of points in the search space reachable by a single crossover event, the number of crossover points needs to be

increased, or to be more precise, the crossover operator needs to be able to cut the string in a variable number of places ranging from 1 up to $L-1$ cuts. In terms of how many points can be reached by cutting and splicing, exploratory power can be restricted in two ways: (1) by limiting the number of cuts allowed and (2) by biasing where the cuts are made or not made on the string. One-point traditional crossover is restricted in both these ways. Only one segment is crossed over, and one of the end points of this segment must be one of the end points of the chromosome.

4. Distributional Bias

In this section we take a look at crossover in terms of its effect on the disruption/preservation of schemata. Our analysis, however, will be somewhat different from that usually taken [6]. In particular, having separated out positional bias, we need a way of analyzing the disruptive effect of crossover that does not include defining length. We begin by considering how many schemata are preserved when two parents of length L are crossed over. With one-point traditional crossover, a child will receive a segment of length x from one parent and a segment of length $L-x$ from the other parent. (For the moment x can be interpreted as the crossover point.) Any schema that has defined positions in point) Any schema that has defined positions in

$$2^x + 2^{(L-x)} \quad (1)$$

schemata will be preserved. Since crossover explores by recombining schemata from the two parents, the schemata that are recombined are thus preserved (and in the process, exploited) but only at the cost of disrupting other schemata. Exploration in the form of new recombinations comes at the cost of disruption. Keeping this in mind, note that formula (1) has its greatest value when x is either 1 or $L-1$ and its lowest value when x is equal to $L/2$. In other words, crossover is most explorative when $x = L/2$. This is because there are potentially up to

$$2^L - (2^x + 2^{(L-x)}) \quad (2)$$

new schemata introduced by crossover. Thus, the choice of x , or more generally, the choice of the distribution of x , will have a significant impact on how explorative or exploitative crossover is. An x that is biased toward $L/2$ will be more disruptive, and thus more explorative, than an x biased toward 1 or $L-1$.

It should be stressed that this analysis is not dependent upon the positional bias of one-point traditional crossover. Although we have been interpreting x as the crossover point, x can be more generally interpreted as a measure of the amount of material (number of bits) which is crossed over to the child from one parent. In other words, this material does not have to be contained in a single segment. If the string were divided into 10 segments, for example, then x would be the number of bits in the 5 segments that a child receives from one of the parents, and $L-x$, the number of bits in the 5 segments that the same child receives from the other parent.

The distributional bias of crossover can also be analyzed in terms of its effect on crossover's exploratory power. To the extent that the amount of material being crossed over is concentrated toward the mean value rather than uniformly distributed, certain points in the search space will be less reachable than others by crossover. For example, if crossover is biased toward exchanging 50% of the material in the chromosome, then search will be biased against points that can be reached by only exchanging a smaller portion (e.g., 10%) of the material. On the other hand, since more points in the search space can be reached by exchanging 50% of the material than 10% of the material, assuming that there is no restriction on which bits are exchanged, a crossover operator with a 50% bias is more exploratory than one with a 10% bias.

One should be careful to distinguish what we are calling the exploratory power of a crossover operator from its disruptive effect. Points in the search space are reached by exchanging material, but not all exchanges are equally disruptive. For example, a crossover operator biased toward exchanging 50% of the material is less exploratory (as measured by how much of the search space is potentially reachable) than one that has no distributional bias (i.e., has a uniform distribution); yet, it is more disruptive since the points it can reach can only be reached by disrupting a large number of schemata.

Because the one-point traditional crossover operator chooses its crossover point randomly along the length of the string with a uniform distribution, it is unbiased with respect to the distribution of material exchanged. No value of x , where x is the number of bits exchanged, is any more likely than any other value. What limits one-point traditional crossover's exploratory power is its positional bias, not its distributional bias. It is an open question to what extent the exploratory power of crossover

should be limited, and whether the biases of one-point traditional crossover are the best way of doing this. We have already expressed reservations about one-point traditional crossover's positional bias. As far as distributional bias is concerned, it is unlikely that the ideal value of x can be fixed a priori. On the other hand, this does not mean that it is therefore optimal to allow it to vary uniformly as in the case of one-point traditional crossover. It might turn out that it is better to let it vary around some value rather than let it vary uniformly from 1 to $L-1$. Perhaps at values near 0 and L it is not explorative enough and at values near $L/2$ it is too disruptive. We doubt if there is anyway of determining this except experimentally, and probably the optimum distributional bias will be problem dependent.

5. The Bias Landscape

We have discussed two dimensions by which we can classify any crossover operator: (1) its positional bias, and (2) its distributional bias. Neither of these dimensions can be characterized by single scalar quantities. The distribution of the amount of material crossed over may be described by a number of measures, including mean, variance, skew, and kurtosis. Positional bias is perhaps even harder to characterize on a single scale. Does a crossover operator biased toward making cuts at parameter boundaries have more positional bias than one point crossover? This would depend upon such factors as the length of the string and the number of parameters.

With these caveats in mind, we have attempted to provide a two dimensional sketch of the crossover landscape in Figure 1. The horizontal axis represents distributional bias indirectly in terms of the expected maximum number of schemata that will be disrupted by a single crossover event. At the left there is a minimum amount of disruption since one, and only one, bit is exchanged. At the right there is a maximum amount of disruption since $L/2$ bits are always exchanged. (Note that an operator may lie in the left side of the landscape, even though the mean amount of material exchanged is near $L/4$, because of high variance around this mean.) The vertical axis represents positional bias. At the top there is no positional bias and at the bottom there is maximum positional bias. At the four corners are four limiting cases. The two limiting cases at the far left involve minimum disruption since only 1 bit is exchanged. In the upper case, a single bit is chosen at random, whereas in the lower case, the bit

exchanged is specified by position. The two limiting cases at the far right represent maximum disruption since exactly $L/2$ bits are exchanged. In the upper case, $L/2$ bits are chosen at random, whereas in the lower case, the bits exchanged are always in the same $L/2$ loci. One-point traditional crossover (1pt-tr) falls near the bottom left corner since it has high positional bias and low distributional bias. The relative position of several other crossover operators are shown in Figure 1.[†] In the next few subsections we analyze a number of alternative crossover operators in terms of positional bias and distributional bias. We have not attempted to uniformly sample the bias landscape, but have picked points of historical interest as well as a few points that are pivotal from the point of view of this study.

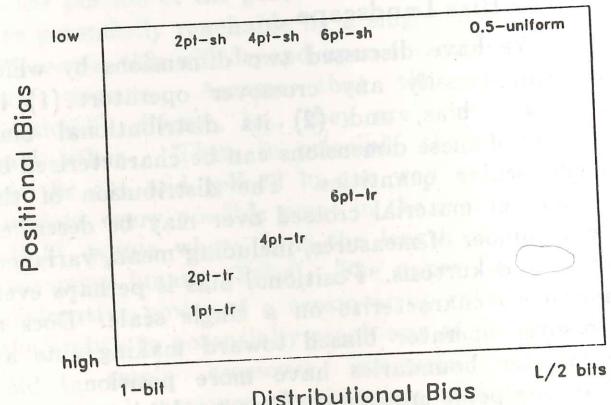


Figure 1. The crossover landscape

5.1. Two-point traditional crossover

Two-point traditional crossover is a step in the direction of reducing, but not eliminating, the positional bias of one-point traditional crossover. Thus in our crossover landscape illustrated in Figure 1, two-point traditional crossover (2pt-tr) appears directly above one-point traditional crossover. Two-point crossover treats the chromosome string as a ring. Two unique points are selected at random, breaking the ring into two segments that are exchanged between the parents to produce two offspring. In order to see how this reduces positional bias, it is useful to view one-point traditional crossover as also performed on a ring. The only difference from this perspective between one-point

[†] Uniform crossover is discussed in section 5.4, and shuffle crossover (sh) is discussed in section 5.5.

and two-point crossover is that in the case of the former one of the points is fixed (i.e., where the end points of the string are joined). To illustrate the impact of this difference, consider a schema of order two (i.e., has only two defined positions) whose two points determine a segment that covers 25% of the ring. Then in the case of two-point crossover the probability of disruption is the probability of one crossover point lying within the segment and one without ($2 * 0.25 * 0.75 = 0.375$). In the case of one-point crossover, on the other hand, the probability of disruption depends upon whether or not the fixed crossover point lies within the segment. If it does, the probability of disruption is the probability of the randomly chosen point falling outside of the segment (0.75), otherwise it is the probability of it falling within the segment (0.25). Thus certain schemata have a privileged status over other schemata with the same pattern but a different position on the string. More generally, whereas one-point crossover can potentially reach only one of $L-1$ points in the search space when two complementary parents are crossed over, two-point crossover can potentially reach any one of $L(L-1)/2$ points in the search space.

Traditional two-point crossover reduces positional bias without introducing any distributional bias. The positional bias, however, is still quite strong. Schemata with defining lengths near $L/2$ are more likely to be disrupted than schemata of the same order but with shorter or longer defining lengths. In terms of exploratory power, $L(L-1)/2$ points in the search space are reachable by two-point crossover as opposed to $L-1$ in the case of one-point crossover. This, however, is far short of the 2^L points that are potentially reachable.

5.2. Multi-point traditional crossover

Multi-point crossover is the natural extension of two-point crossover [4]. Increasing the number of crossover points reduces positional bias, but unlike the move from one point to two-point crossover, it also introduces some distributional bias. Like two-point crossover, multi-point crossover treats the chromosome as a ring which the crossover points cut into segments. Since the segments of the child must alternate between the two parents, there must be an even number of segments, and hence an even number of crossover points. Alternatively, if there is an odd number N of crossover points, either one point must be ignored, or the end point of the string must be treated as the $N+1$ crossover point. Increasing the number of pairs of crossover points

has two effects. First, it decreases the positional bias, and, secondly, it introduces a distributional bias. The first effect follows from the fact that as the number of crossover points is increased, the expected length of the segments decreases, and thus the probability of shorter schemata being disrupted increases whereas the probability of longer schemata being disrupted approaches 0.5. Secondly, the distribution of the number of bits crossed over is no longer uniform. As the number of crossover points increases, the distribution of material from the one of the parents (e.g., the first parent) approaches a binomial distribution with a mean of $L/2$ [4] [9]. Figure 1 represents this by showing the path of the multi-point traditional crossover operators (4pt-tr, 6pt-tr) moving towards the upper right hand corner as the number of crossover points increases.

5.3. Segmented Crossover

Strictly speaking, in order for multi-point to approach a binomial distribution with a probability of 0.5, the number of crossover points would need to be variable with a mean of $L/2$. Segmented crossover is a variant of multi-point crossover, which allows the number of crossover points to vary. Instead of choosing in advance a fixed number of unique crossover points, a segment switch rate is specified. The switch rate specifies the probability that a segment will end at any point in the string. For example, if the switch rate is 0.25, then once a segment is begun, there is a 0.25 probability of ending it at each bit and beginning a new segment by making a cut. With a 0.25 switch rate the expected number of segments would be $L/4$. Unlike $L/4$ -point crossover, however, the number of crossover points will vary.

By using two switch rates, one for ending segments that are crossed over and one for ending segments that are not crossed over, the expected length of the segments being crossed over no longer needs to be equal to the expected length of the segments not being crossed over. For example, switch rates of 0.1 and 0.3 would segment the string (treated as a ring) so that the expected segment lengths (and so the expected amount of material) from one parent would be three times as long as from the other parent. Furthermore, by setting the switch rates so that they total 1.0, segmented crossover becomes equivalent to uniform crossover with an exchange rate equal to the smaller of the switch rates. For example, switch rates of 0.25 and 0.75 would be equivalent to uniform crossover with an exchange rate of 0.25.[†]

[†] Since segmented crossover is a variant of multi-point traditional crossover, it is not shown in Figure 1.

5.4. Uniform Crossover

Uniform crossover exchanges bits rather than segments [1]. For each bit position in the string the bits from the two parents are exchanged with fixed probability p . Since the probability of exchanging two bits in each position is independent of the choice made with regard to any other position, uniform crossover has no positional bias. It has a strong distributional bias, however. The expected number x of bits exchanged will be $p \cdot L$, and x will have a binomial distribution. Hence, uniform crossover reverses the biases of one-point traditional crossover. Its disruptive effect, and hence where it will lie in the landscape depicted in Figure 1, depends upon p , the exchange rate. If p is 0.5, uniform crossover (0.5-uniform) will be highly disruptive and lie in the opposite corner of our bias landscape from one-point traditional crossover.

5.5. Shuffle crossover

Shuffle crossover is similar to traditional crossover except that it randomly shuffles the bit positions of the two strings in tandem before crossing them over and then unshuffles the strings after the segments to the right of crossover point have been exchanged. Thus, crossover no longer has a single, consistent positional bias because the positions are randomly reassigned each time crossover is performed. Bits that are far apart are no more likely to be disrupted than bits that are close together. The shuffle crossover operator is designed to eliminate positional bias by having a schema disruption probability that is independent of schema defining length. More generally, shuffle crossover provides for increased sampling of interacting bits that are far apart (i.e., long defining length) and more disruption of neighboring bits that are only spuriously correlated (as well as bits that are truly correlated).

The shuffle operation is independent of the number of crossover points. Thus, shuffle crossover can be combined with multi-point and segmented crossover as well as a one-point crossover. (Since there is no fixed end point, there is no distinction between N-point shuffle crossover where N is an even number and (N-1)-point shuffle crossover.) In our crossover landscape every N-point traditional crossover operator has an N-point shuffle crossover operator (2pt-sh, 4pt-sh, 6pt-sh) directly above it. Because the shuffle operation has already eliminated any positional bias, an increase in the (even) number of crossover points cannot reduce positional bias, but it will introduce distributional bias. If one-point (or two-point) crossover is used, then

shuffle crossover will have no distributional bias. As the number of crossover points approach $L/2$, shuffle crossover will approximate uniform crossover with an exchange rate of 0.5. Again, to be equivalent to uniform crossover, the number of crossover points would have to be allowed to vary in the appropriate manner.

6. Empirical Tests

In this section we attempt to determine how one-point traditional crossover's combination of no distributional bias and a strong positional bias affects search in practice, and whether or not some other combination of biases is more effective. To do this, we compare the performance of twelve different crossover operators on minimizing a set of functions. The crossover operators include traditional and shuffle crossover using different numbers of crossover points, uniform crossover at 0.25 and 0.5 exchange rates, and segmented crossover using a 0.25 switch rate. The functions are the five scalar-valued functions used by De Jong [4] to test the GA's performance on search spaces with a variety of characteristics. These test functions are summarized in Table 1.

TABLE 1. Test Functions

Fcn	Dim	Space Size	Description
f1	3	1.0×10^9	parabola
f2	2	1.7×10^6	Rosenbrock's saddle
f3	5	1.0×10^{15}	step function
f4	30	1.0×10^{72}	quadratic with noise
f5	2	1.7×10^{10}	Shekel's foxholes

Grefenstette [5] found a set of GA parameters (crossover rate, mutation rate, population size, generation gap, and scaling window) that performed consistently better than any previously known set with regard to online performance on functions f1-f5 using one-point traditional crossover. We use the parameter settings he discovered for our first set of experiments. Our experiments differ from Grefenstette's, however, in three respects. First, Grefenstette's, demonstrated the superiority of Gray coding over binary coding for these functions; we now use Gray coding in all of our experiments. Second, we use an improved selection procedure devised by Baker [2] that was not available at the time Grefenstette ran his experiments. Third, Grefenstette compared performance at a fixed number of evaluations (5000). Gray coding improves performance enough that comparison at 5000 evaluations is no longer

interesting: on some of the functions too many searches find the optimum before 5000 evaluations. Rather than try to determine a new, fixed number of evaluations at which to compare performance, we use the mean number of evaluations required to find the global optimum [1].

Given these differences, the question arises whether the parameters discovered by Grefenstette may no longer be optimal, and, therefore, could accidentally bias our results towards some crossover operator other than one-point traditional crossover. We have done extensive tests that show the Grefenstette parameter settings are optimal or near optimal for one-point traditional crossover with regard to online performance on these functions with Gray code and Baker selection [10]. Unfortunately, there is no reason to believe that these settings will be optimal for our performance criterion — the mean number of evaluations required to find the global optimum. In order to make sure that our experiments were not biased in favor of any crossover operator other than one-point traditional, we adopted Grefenstette's strategy of setting a genetic algorithm to optimize a genetic algorithm in order to locate a good set of parameter values for our new performance criterion. Each evaluation of the meta-GA comprised running one GA against each of the five test functions until either the optimum value was found or a maximum number of trials was reached, normalizing each result with respect to the number of trials that are required when Grefenstette's parameter settings are used, and then summing the normalized results. The parameter set searched at the meta-level included population size, mutation rate, and crossover rate.

The meta-search revealed that the GA worked well for the suite of five test functions at the following parameter settings: population size = 20, mutation rate = 0.016, crossover rate = 0.70. (Compare these to Grefenstette's settings for online performance: population size = 30, mutation rate = 0.010, crossover rate = 0.95.) Our second set of experiments use these new settings.

Table 2 shows the average number of function evaluations required for the various crossover operators tested to find the global optimum of each of the five test functions using Grefenstette's parameter settings. Table 3 shows the same thing using the new parameter settings. Each mean in both sets of experiments is based on 50 runs. Lower mean values suggest faster, more effective search. The representative standard errors for each function are shown at the bottom of the two tables.

TABLE 2. Mean Number of Evaluations to Find the Global Optimum

Grefenstette's Parameter Settings					
	f1	f2	f3	f4	f5
1pt-tr	2487	15148	2952	4024	4683
2pt-tr	2323	16394	2381	3432	4379
4pt-tr	2283	14316	2676	2976	4534
8pt-tr	2210	14229	2259	3070	4334
12pt-tr	2400	13700	2517	2909	4312
2pt-sh	2498	13709	2801	4144	4057
4pt-sh	2387	14636	2865	3948	4628
8pt-sh	2152	12666	2733	3741	4683
12pt-sh	2448	12309	2473	4080	4219
0.25-un	2199	14672	3131	3705	4233
0.50-un	2372	14569	2350	3422	4280
0.25-sg	2321	14809	2573	3762	4523
SE	101	1270	244	148	242

Let us first examine the results shown in Table 2. Although it may not be clear at first which crossover operator is the winner, there is little question that one-point traditional crossover is the loser. Of the twelve crossover operators tested, one-point traditional crossover ranks as one of the worst performers on all five functions. Of the remaining eleven operators, only four ever perform worse than one-point traditional crossover: two-point traditional crossover on f2, two-point shuffle crossover on f1 and f4, twelve-point shuffle crossover on f4, and 0.25 uniform crossover on f3. Eight-point traditional crossover appears to be the best crossover operator. It is the only crossover operator which always ranks in the top half for all five functions, and on three of the functions its performance is more than two standard errors better than that of one-point traditional.

More generally, it appears that increasing the number of crossover points improves performance for both traditional and shuffle crossover, although beyond some maximum number performance begins to fall. Four-point shuffle seems to be the exception to this latter trend, being worse than two-point shuffle three out of five times. (We have no explanation why four-point shuffle does worse.) Finally, it should be noted that uniform crossover with a 0.5 exchange rate is one of the better performers.

The results shown in Table 3 also do not bode well for one-point traditional crossover. The new parameter settings help improve one-point traditional crossover's performance on four of the functions (at the expense of f4), but the other

TABLE 3. Mean Number of Evaluations to Find the Global Optimum

New Parameter Settings					
	f1	f2	f3	f4	f5
1pt-tr	1700	12439	1790	4955	3362
2pt-tr	1683	10623	2085	4712	3156
4pt-tr	1726	10063	1787	5346	3162
8pt-tr	1538	9477	1740	4137	3004
12pt-tr	1684	10042	1873	4488	3006
2pt-sh	1721	9695	1921	5949	2838
4pt-sh	1528	10134	1700	5054	2698
8pt-sh	1629	9763	1718	5384	3063
12pt-sh	1762	9827	1821	4892	2763
0.25-un	1707	10053	1930	4911	2877
0.50-un	1618	10000	1887	5904	2922
0.25-sg	1574	11199	1875	4989	2892
SE	77	737	135	341	149

crossover operators also show the same pattern of improvement. Relative to the other operators, one-point traditional crossover's performance has improved on functions f1, f3, and f4, but it has slipped to last place on f2 and f5. Eight-point traditional crossover is still a strong performer. It performs better than one-point traditional on all five functions and on four of them by more than two standard errors. Uniform crossover (0.5) is no longer such a strong performer, mainly because of its poor performance on f4.[†] On the other hand, the shuffle crossover operators (with the exception of two-point) become viable contenders.

We have summarized the results for both sets of parameters in Table 4. For each parameter set we have provided two methods of ranking. The first method, shown in the 'R' columns, is simply the ranking of the sum of the ranks across the five functions. For example, when Grefenstette's settings are used, one-point traditional ranks eleventh on four of the functions and tenth on the remainder. So the sum of its rankings is fifty-four. This sum places it in twelfth place relative to the other eleven crossover operators. The second

[†] We hypothesize that the higher mutation rate used by the new parameter settings is too disruptive for longer chromosomes such as f4 (the longest in the suite). This would explain why all the crossover operators perform worse on f4 when the new parameter set is used, and it might also explain why the performance of the uniform crossover operator, whose operation is more disruptive (and "mutation-like") than the other operators, deteriorates with respect to the other operators. This suggests that the mutation rate should be a function of the chromosome length.

method, shown in the 'N' columns, is the ranking of the sums of the normalized mean number of evaluations to find the global optimization. Each mean in Tables 2 and 3 is normalized by determining how much worse it is than the best mean for that function and then dividing it by the difference between the worst and best means for that function. For example, the normalized value of the f1 mean for one-point traditional crossover when Grefenstette's parameter settings are used is $\frac{2487-2152}{2498-2152} = 0.968$. These normalized values then are summed over the functions and ranked.

It can be seen from Table 4 that according to either of these methods of ranking the crossover operators, one-point traditional is the clear loser and traditional eight-point is the clear winner, even when the parameter settings are optimized for one-point traditional crossover. We do not interpret this to mean that eight-point traditional crossover should now be the crossover operator of choice. The fact that its performance on f5, the only multi-modal function in the suite, is rather mediocre should give one pause. Although eight-point traditional does better than any of the other traditional crossover operators on f5 when the new parameter settings are used, all of the non-traditional crossover operators, except eight-point shuffle, do better than eight-point traditional.[†]

Parms	Gref.		New	
	R	N	R	N
Method				
1pt-tr	12	12	11	12
2pt-tr	8	7	10	11
4pt-tr	4	4	11	9
8pt-tr	1	1	1	1
12pt-tr	2	2	5	5
2pt-sh	10	10	9	10
4pt-sh	11	11	2	2
8pt-sh	6	5	4	3
12pt-sh	4	6	3	4
0.25-un	7	8	8	7
0.50-un	3	3	7	8
0.25-sg	9	9	6	6

[†] We have compared the twelve crossover operators discussed in this paper on four other multi-modal functions using Grefenstette's settings. Eight-point traditional's performance was rather disappointing — it ranked sixth (rank of the sum of the ranks). On the other hand, one-point traditional crossover, along with two-point traditional, continued to place last.

7. Conclusions

The GA community has been aware of the positional bias of one-point traditional crossover, but has assumed that this bias could be exploited to aid search. The GA community has also been aware of the low distributional bias of one-point traditional crossover, claiming that it is a virtue that the expected number of schemata disrupted is relatively low. This paper calls into question both of these assumptions about crossover bias. Of the family of crossover operators discussed in this paper, one-point traditional crossover emerges as the lowest achiever. Although we are reluctant to pick a winner on the basis of our test suite of five functions, our experimental results indicate that it is better to have more distributional bias toward higher disruption rates and perhaps less positional bias than has been traditionally assumed. It may turn out that there is no best function-independent crossover operator, but viewing crossover in terms of the bias landscape sketched in this paper may be helpful in learning how to adjust the exploration-exploitation tradeoff to various types of problems. Furthermore, it may suggest new ways in which the search mechanism can dynamically make these tradeoffs much in the spirit of Punctuated Crossover [8] and ARGOT [11].

References

1. D. H. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, Boston, MA, 1987.
2. J. E. Baker, Reducing Bias and Efficiency in the Selection Algorithm, *Genetic Algorithms and Their Applications: Proceedings of the Second*, MIT, July 28-31 1987, 14-21.
3. R. A. Caruana and J. D. Schaffer, Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms, *Proceedings of the 5th International Conference on Machine Learning*, Morgan Kaufmann, Los Altos, CA, June 12-14 1988, 153-161.
4. K. A. De Jong, Analysis of the Behavior of a Class of Genetic Adaptive Systems, Ph.D. Dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, 1975.
5. J. J. Grefenstette, Optimization of Control Parameters for Genetic Algorithms, *IEEE Transactions on Systems, Man & Cybernetics SMC-16,1* (January-February 1986), 122-128.
6. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
7. J. D. Schaffer, Some Effects of Selection Procedures on Hyperplane Sampling by Genetic Algorithms, in *Genetic Algorithms and Simulated Annealing*, L. D. Davis (editor), Morgan Kaufmann, Los Altos, CA, 1987, 36-40. Philips Laboratories, Technical Report-86-034, Sept 20, 1986.
8. J. D. Schaffer and A. Morishima, An Adaptive Crossover Distribution Mechanism for Genetic Algorithms, *Genetic Algorithms and Their Applications: Proceedings of the Second*, MIT, July 28-31 1987, 36-40. Philips Laboratories, Technical Report-86-052, Dec 30, 1986.
9. J. D. Schaffer and A. Morishima, Adaptive Knowledge Representation: A Content Sensitive Recombination Mechanism for Genetic Algorithms, *International Journal of Intelligent Systems 3,3* (March 1988).
10. J. D. Schaffer, R. A. Caruana, L. Eshelman and R. Das, A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization, in *these proceedings*.
11. C. G. Shaefer, The ARGOT Strategy: Adaptive Representation Genetic Optimizer Technique, *Genetic Algorithms and Their Applications: Proceedings of the Second*, MIT, July 28-31 1987, 50-58.
12. D. J. Sirag and P. Weisser, Toward a Unified Thermodynamic Genetic Operator, *Genetic Algorithms and Their Applications: Proceedings of the Second*, MIT, July 28-31 1987, 116-122.
13. P. E. Utgoff, Adjusting Bias in Concept Learning, *8th International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, August 1983. parisons with a standard genetic algorithm on a standard set of test functions for genetic algorithms.
14. D. Whitley and J. Kauth, GENITOR: A Different Genetic Algorithm, *Proceedings 1988 Rocky Mountain Conference on Artificial Intelligence*, 1988. Tech. Rpt. Computer Science-88-101, Colorado State University.