

---

# The Distributional Biases of Crossover Operators

---

**Soraya Rana**

Computer Science Department  
Colorado State University  
Fort Collins, CO 80523 USA  
email: rana@cs.colostate.edu

## Abstract

The choice of genetic operators is one way in which genetic algorithms can be tailored to specific optimization problems. For bit represented problems, the choice of crossover operator, or the choice not to use a crossover operator, can dramatically affect search performance. The efficacy of crossover for genetic search is governed by the relationship between the crossover biases and the search problem itself. Crossover operators have two forms of bias: positional bias and distributional bias. This paper analytically characterizes the distributional biases that exist for several commonly used crossover operators. The effects of the crossover biases are empirically examined for a Simple Genetic Algorithm applied to two types of NK-landscapes.

## 1 INTRODUCTION

Striking a balance between exploration and exploitation is essential for any search algorithm. Genetic operators are the means by which genetic algorithms attempt to achieve that balance. Selection is vital to the execution of the genetic algorithm because it culls the population and narrows the focus of search to regions with potentially high fitness. Crossover and mutation are the mechanisms for exploration; however, *both* crossover and mutation are not always necessary for exploration. In particular, the merits of crossover for genetic search have often been questioned (Fogel and Atmar, 1990, Spears, 1992, Jones, 1995, Eshelman and Schaffer, 1993).

Typically, mutation is a necessity for traditional genetic algorithms because it maintains diversity in a finite population and also ensures that every point in the

search space has a nonzero probability of being visited. On the other hand, crossover does not introduce allelic diversity into the population and does not always guarantee that every point in the search space can be visited. Since crossover requires two parent strings, its exploratory power depends on the differences between those parents. As the population converges, the exploratory power of crossover diminishes. However, even if crossover is taken out of a population-based search paradigm, its exploratory power is still limited by its inherent bias in how points are sampled.

There are two forms of crossover biases: positional and distributional (Eshelman et al., 1989). Positional bias refers to the frequency that crossover exchanges bits occurring in particular locations on the string. For instance, given the parent strings 01110 and 10001, the offspring 00000 cannot be produced under one point crossover because the values at both the first and the last bit can never be exchanged concurrently. Analytical results for the positional bias distributions for several crossover operators have been derived (Booker, 1992). Distributional bias refers to the number of bits that are swapped under a specific crossover operator. This paper presents mathematical descriptions for the distribution of Hamming distances between parents and offspring for four commonly used crossover operators. These distributions model how “large steps” are made by crossover in the first generation of a genetic algorithm with a randomly initialized population.

## 2 CROSSOVER OPERATORS AND LOCAL SEARCH NEIGHBORHOODS

Genetic algorithms can be viewed as a population-based form of local search (Ulder et al., 1990, Reeves, 1994). Local search is an iterative process whereby a point or population is continually modified and evalu-

ated until a goal state or stopping criterion is reached. Since one generation of a genetic algorithm consists of applying selection, crossover and mutation to a population to produce a new population, genetic search fits into the paradigm of local search. Furthermore, all three genetic operators can, in turn, be considered as individual local search operators (Jones, 1995).

Local search operators explore the search space by sampling points from a predefined *neighborhood* (Papadimitriou and Steiglitz, 1982). Given a discrete search problem defined over a set of points representing all possible inputs  $S$ , a *neighborhood*  $N$  is a mapping:

$$N : S \rightarrow \mathcal{P}(S)$$

Where  $\mathcal{P}(S)$  is the power set of  $S$ . So the neighborhood  $N$  is a function which takes a single point,  $s \in S$ , and generates a subset of points that are neighbors of  $s$ . For most local search operators,  $N$  is defined to be a small change to single solution. For instance, given an  $L$  bit string,  $x \in \mathcal{B}^L$ ,  $\mathcal{B} = \{0, 1\}$ , define:

$$N_1(x) = \{y : y \in \mathcal{B}^L \text{ such that } bc(x \oplus y) = 1\}$$

where  $bc(x)$  returns the number of 1-bits in the bit string  $x$ . So,  $N_1(x)$  is simply the single-bit flip or the 1-change neighborhood (Papadimitriou and Steiglitz, 1982). To define a  $k$ -change neighborhood about a bit string:

$$N_k(x) = \{y : y \in \mathcal{B}^L \text{ such that } bc(x \oplus y) = k\}$$

The local search neighborhood for the mutation operator is typically considered to be the 1-change neighborhood. The local search neighborhood for crossover cannot be immediately defined as a  $k$ -change neighborhood because crossover neighborhoods must be defined by pairs of input strings (Reeves, 1994, Jones, 1995, Culberson, 1992). However, a  $k$ -change neighborhood *can* be used to model the step-sizes taken by crossover.

In order to formally describe crossover neighborhoods, additional terminology needs to be introduced. Let  $i, j \in \mathcal{B}^L$  and define  $i \subseteq j$  to denote  $i$  is a **submask** of  $j$  to indicate that  $i$  has a 0-bit wherever  $j$  has a 0-bit and  $i$  has a 0-bit or 1-bit wherever  $j$  has a 1-bit.

Crossover can be performed by applying a mask  $m$  to a string pair  $x$  and  $y$  (Syswerda, 1989). The crossover mask,  $m$ , contains a 1-bit where the parent strings  $x$  and  $y$  should exchange bit values. Consider the following example:

Let  $x = 11101010$ ,  $y = 00101000$ ,  $m = 11110000$   
Then  $m$  indicates the  $x$  and  $y$  should swap their first four bit values to produce 00101010 and 11101000.

If the crossover mask  $m$  is restricted by the distance between the parent strings  $x$  and  $y$  so  $m \subseteq x \oplus y$  where the string  $d = x \oplus y$  is called a **Hamming distance mask**, then the offspring can be produced using the exclusive-or operator. So, given  $m \subseteq d$  the possible offspring are  $x \oplus m$  and  $y \oplus m$ . Then the set of all potential offspring for a string pair whose Hamming distance mask is  $d$  can be generated for all  $m \subseteq d$  and the resulting distance between parents and offspring is  $bc(m)$ . For instance, in the previous example:

$$x = 11101010, y = 00101000, m = 11110000$$

However, if  $m$  is restricted by the distance between  $x$  and  $y$ , then  $m = 11000000$ . Then the pair of offspring associated with  $(x, y)$  are:

$$x \oplus m = 00101010 \text{ and } y \oplus m = 11101000$$

and the offspring differ from the parents by 2 bits. Note that Hamming distance between parents and offspring is measured conservatively so that if  $z$  is an offspring of  $(x, y)$ , then the parent-offspring distance is  $\min(bc(x \oplus z), bc(y \oplus z))$ .

Rather than explicitly defining and counting the crossover neighborhoods for specific pairs of strings, the task of counting all possible Hamming distances sampled by a particular crossover operator reduces to counting all possible crossover submasks of all possible Hamming distance masks subject to positional bias constraints. Over the space of all possible pairs of strings, each of the possible  $2^L$  Hamming distance masks will be encountered exactly  $2^L$  times. The set of all possible submasks with  $bc(m) = k$  for a specific Hamming distance mask,  $d \in \mathcal{B}^L$ , can be computed using a  $k$ -change neighborhood definition over Hamming distance masks:

$$N_k(d) = \{m : m \subseteq d \text{ such that } bc(m) = k\}$$

$$\text{where } 0 \leq k \leq bc(d)$$

This general form of the neighborhood of submasks can be redefined for each particular crossover operator in order to select only the submasks that meet the specific positional biases of that crossover operator.

## 2.1 HAMMING DISTANCE NEIGHBORS FOR ONE POINT AND TWO POINT CROSSOVER

**One point** crossover and **two point** crossover are two examples of crossover operators with positional biases. For an  $L$  bit string, one point crossover chooses a crossover point,  $c$ , somewhere between the first and last bit and creates offspring by concatenating the first  $c$  bits from one parent with the remaining bits from the

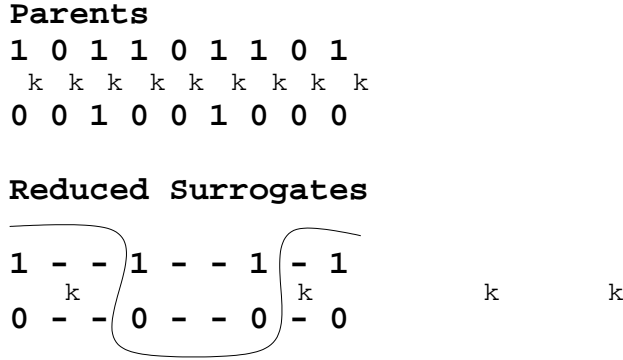


Figure 1: Sample of two point crossover with reduced surrogates.

second parent. Two point crossover uses two crossover points to create a segment that is swapped between the parents. In both cases, a contiguous block of bits must be exchanged between two parents.

Reduced surrogates are a modification to traditional crossover which ensures there is an equal chance of producing any possible offspring (Booker, 1987). Figure 1 illustrates the use of reduced surrogates with two point crossover. Normally, two point crossover can select from any of the 9 possible crossover points. Reduced surrogates constrain the set of available crossover points by Hamming distance so that a crossover point occurs between a pair of differing bits. In this example, two point crossover must choose 2 out of 4 possible crossover points. Reduced surrogates also guarantees that when the parent strings are at least 2 bits different, the offspring will always differ from the parents under one or two point crossover.

Reduced surrogates restrict the neighborhood of crossover which directly impacts the count of parent-offspring distances. In Table 1, the bit strings represent the Hamming distance masks for all possible parent pairs for a 4-bit problem. The table represents the Hamming distance between the parents and offspring if we were to choose any of the possible crossover points for either one point crossover or one point crossover using reduced surrogates. Regardless of the mask  $d_i$ , traditional one point crossover always has 3 potential crossover points. The number of crossover points varies from 0 to 3 for one point crossover with reduced surrogates. The Hamming distances (HD) are enumerated for each possible crossover point. The  $d_0$  represents the case where both parents are identical. For one point crossover, we can mate two identical parents three ways to produce offspring that do not differ from the parent pair. For reduced surrogates, there are no crossover points so the offspring are simply copies of

HD-Mask	One-Pt			Red.Sur			
	Xover Point Locations						
	1	2	3	0	1	2	3
$d_0=0000$	0	0	0	0			
$d_1=0001$	0	0	0	0			
$d_2=0010$	0	0	0	0			
$d_3=0011$	0	0	1		1		
$d_4=0100$	0	0	0	0			
$d_5=0101$	0	1	1		1		
$d_6=0110$	0	1	0		1		
$d_7=0111$	0	1	1		1	1	
$d_8=1000$	0	0	0	0			
$d_9=1001$	1	1	1		1		
$d_{10}=1010$	1	1	0		1		
$d_{11}=1011$	1	1	1		1	1	
$d_{12}=1100$	1	0	0		1		
$d_{13}=1101$	1	1	1		1	1	
$d_{14}=1110$	1	1	0		1	1	
$d_{15}=1111$	1	2	1		1	2	1

Table 1: Enumeration of 4-bit space to compute Hamming distance results of one point crossover with and without reduced surrogates.

the parents (i.e. there is one way to generate the case of  $HD_0$ ). Also note that for  $d_3$ , there are two ways in which traditional one point crossover can result in no change to the parent strings.

The goal of this paper is to compute the distributions of Hamming distances between parents and all possible offspring. The number of available crossover points will affect the distribution, particularly if there are cases where the same pair of offspring can occur in multiple ways. While this type of calculation can certainly be performed, the use of reduced surrogates simplifies the calculations. It is assumed throughout the remainder of the paper that crossover is performed using reduced surrogates.

### 2.1.1 Hamming Distance Counts

The function,  $N_k(d)$ , computes all possible submasks of a particular Hamming distance mask. Every crossover operator that uses reduced surrogates will have a neighborhood defined that is a subset of  $N_k(d)$ . The subsets are determined by the specific constraints for the crossover operators. Formalizing the constraints for one point and two point crossover require additional functions to be defined.

The function **pack** is a mapping:

$$pack : \mathcal{B}^L \times \mathcal{B}^L \rightarrow \mathcal{B}^M \quad \text{where } M \leq L$$

The **pack** function takes an  $L$ -bit string as its first argument and an  $L$ -bit mask containing exactly  $M$  1-bits as its second argument. The result of **pack** is an  $M$ -bit string extracted from the  $L$ -bit input string by

the bit mask. So, for example:

$$\text{pack}(011000, 011010) \implies 110$$

Now define two additional functions,  $\text{span}$  and  $b_i$ :

$$\text{span} : \mathcal{B}^L \rightarrow \mathbb{N}$$

where  $\text{span}$  counts the number of bits (inclusive) between the outermost 1 bits in the string. For instance  $\text{span}(0010010) = 4$ . Also, define the function  $b_i : \mathcal{B}^L \rightarrow \{0, 1\}$  to return the value at bit position  $i$  in the input string where  $b_0(x)$  returns the rightmost bit in the string and  $b_{L-1}(x)$  returns the leftmost bit in the string.

The two neighborhood submask descriptions for one point crossover,  $N_k^o(M)$ , and two point crossover,  $N_k^t(M)$  can be expressed as:

$$N_k^o(d) = \left\{ \begin{array}{l} m : m \in d \quad \text{such that} \\ bc(m) = k \quad \text{and} \\ \text{span}(\text{pack}(m, d)) = k \quad \text{and} \\ b_0(\text{pack}(m, d)) \neq b_{bc(d)}(\text{pack}(m, d)) \end{array} \right. \quad (1)$$

where  $1 \leq k \leq \lfloor L/2 \rfloor$  and  $2k \leq bc(d) \leq L$  and

$$N_k^t(d) = \left\{ \begin{array}{l} m : m \in d \quad \text{such that} \\ bc(m) = k \quad \text{and} \\ \text{span}(\text{pack}(m, d)) = k \end{array} \right. \quad (2)$$

where  $1 \leq k \leq \lfloor L/2 \rfloor$  and  $2k \leq bc(d) \leq L$

The case for string pairs resulting in 0 crossover points, namely when  $bc(d) = 0$  and  $bc(d) = 1$ , are handled separately.

The number of neighbors with a specific Hamming distance,  $HD_k$ , can be computed by generating and counting all possible neighbors for each crossover operator. Starting with one point crossover, for  $1 \leq k \leq \lfloor L/2 \rfloor$ , the only strings that can produce  $HD_k$  offspring must differ by at least  $2k$  bits. Parent strings,  $x, y$  are considered to be  $k$  bits away from an offspring  $z$  if  $bc(x \oplus z) = k$  or  $bc(y \oplus z) = k$  and  $bc(x \oplus y) \geq 2k$ .

Since we are considering one point crossover, there are only 2 ways to produce  $HD_k$  neighbors when the Hamming distance masks contain at least  $2k$  1-bits: to choose crossover submasks with exactly  $k$  1-bits to the right or left of the crossover point. We can count both cases by summing over all possible strings of Hamming distance  $2k$  to  $L$  for both cases. When there is a Hamming distance of  $2k$ , both offspring will differ by  $k$  bits on both the left and right so this case should only be counted once.

The formula for computing the total number of  $HD_k$  neighbors produced under a one point crossover operator with reduced surrogates is:

When  $k = 0$

$$HD_0 = \binom{L}{0} + \binom{L}{1} = 1 + L$$

when  $1 \leq k \leq \lfloor L/2 \rfloor$ .

$$HD_k = \left[ 2 \sum_{i=2k}^L \binom{L}{i} \right] - \binom{L}{2k}$$

For two point crossover, the number of  $HD_0$  neighbors is exactly the same as for one point crossover because there are the same number of Hamming distance masks that result in no crossover points. Given a Hamming distance mask,  $d$ , with at least  $2k$  1-bits, there are  $bc(d)$  pairs of crossover points that will produce submasks with exactly  $k$  1-bits. Thus the counting formula for the  $HD_k$  neighbors for two point crossover with reduced surrogates is:

When  $k = 0$

$$HD_0 = \binom{L}{0} + \binom{L}{1} = 1 + L$$

when  $1 \leq k \leq \lfloor L/2 \rfloor$ .

$$HD_k = \sum_{i=2k}^L i \binom{L}{i}$$

## 2.2 HAMMING DISTANCE NEIGHBORS FOR UNIFORM CROSSOVER

Two crossover operators which have no positional bias are **uniform** crossover and **HUX** (Eshelman, 1991). Uniform crossover can be thought of as  $n$  point crossover because each bit that differs between parents can be flipped independently of one another. It is assumed that uniform crossover will also be applied with reduced surrogates. HUX is a variant of uniform crossover that, by definition, toggles exactly half of the differing bits between parents.

The two neighborhood submask descriptions for uniform crossover,  $N_k^u(d)$ , and HUX,  $N_k^h(d)$  can be expressed as:

$$N_k^u(d) = \{ m : m \in d \quad \text{such that} \quad bc(m) = k \} \quad (3)$$

where  $0 \leq k \leq \lfloor L/2 \rfloor$  and  $2k \leq bc(d) \leq L$  and

$$N_k^h(d) = \{ m : m \in d \quad \text{such that} \quad bc(m) = k \} \quad (4)$$

where  $0 \leq k \leq \lfloor L/2 \rfloor$  and  $2k \leq bc(d) \leq 2k + 1$

For uniform crossover, the only way to produce an  $HD_k$  neighbor is to have a Hamming distance mask with at least  $2k$  1-bits. Given Hamming distance masks with  $i$  1-bits where  $2k \leq i \leq L$ , there are  $\binom{i}{k}$  ways to generate crossover submasks with  $k$  bits set. When the Hamming distance mask has exactly  $2k$  bits set, then both parents will produce offspring with a distance of  $k$ , so this case should only be counted once.

The formula for counting the possible  $HD_k$  neighbors under uniform crossover is:

$$HD_k = \left[ 2 \sum_{i=2k}^L \binom{L}{i} \binom{i}{k} \right] - \binom{L}{2k} \binom{2k}{k}$$

when  $0 \leq k \leq \lfloor L/2 \rfloor$ .

HUX randomly flips exactly half of the differing bits. In order to produce offspring that are  $HD_k$  from a parent under HUX, the parents need to differ by either  $2k$  or  $2k + 1$  (when  $k < \lfloor L/2 \rfloor$ ) bits.

The formula for computing the number of offspring under HUX is:

when  $0 \leq k < \lfloor L/2 \rfloor$ .

$$HD_k = \binom{L}{2k} \binom{2k}{k} + \binom{L}{2k+1} \binom{2k+1}{k}$$

when  $k = L/2$  (i.e.  $k$  is exactly half of the string length and  $L$  is even),

$$HD_k = \binom{L}{k}$$

### 2.3 THE DISTRIBUTIONS OF HAMMING DISTANCES FOR CROSSOVER

Normalizing the counting formulas for Hamming distances 0 through  $L/2$  forms the distribution of all possible Hamming distance neighbors for crossover. The distributions represent the exact distribution of parent-offspring distances for crossover if all strings are paired uniformly at random and also represent the exact distributional biases that exist for this set of four crossover operators.

Figure 2 shows the distributions for parent-offspring distances for one and two point crossover using reduced surrogates. The values on the horizontal axis represent the set of possible Hamming distances. The values on the vertical axis represent the proportion of crossover events resulting in specific parent-offspring

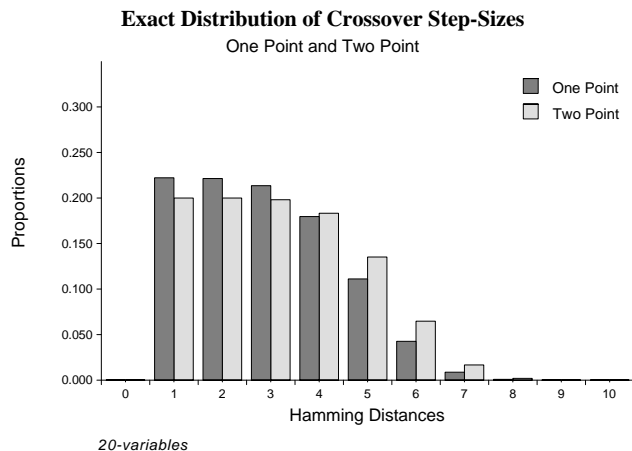


Figure 2: Exact sampling distributions for one and two point crossover with reduced surrogates.

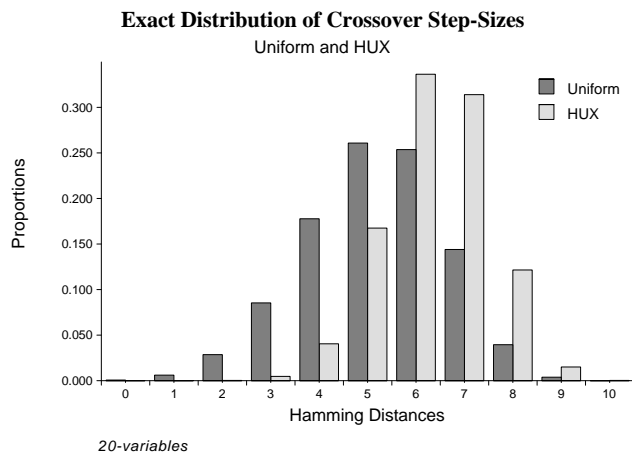


Figure 3: Exact sampling distributions for uniform crossover and HUX.

distances. Due to the conservative calculation of distance, the maximum Hamming distance between parents and offspring is  $\frac{L}{2}$ . Note that this distribution is heavily weighted to the left which means that these two operators are biased to make moves that are of low Hamming distances. In practice, a genetic algorithm would not sample the distribution in this manner. The genetic algorithm will tend to bias the sampling even more heavily towards the lower Hamming distances because selection drives the population to become more uniform. This sampling distribution illustrates that there are practical limits to the exploratory power of crossover operators.

Figure 3 shows the distributions for parent-offspring distances for uniform crossover and HUX. In this case, this distribution represents the step-sizes that would be taken by uniform crossover and HUX if they were

applied to a population with all strings in equal proportion and if all strings were randomly paired. Note that these distributions produce offspring with much larger Hamming distances (on average) than one or two point crossover.

### 3 EXAMPLE: SGA APPLIED TO NK-LANDSCAPES

The distributions computed in the previous section assume that all strings are paired with equal probability. While this may seem impractical, given a simple genetic algorithm (SGA) (Goldberg, 1989) with a large population, the distributions are a very close approximation to the distribution of parent-offspring distances sampled in the first generation. To illustrate how the initial population of an SGA samples offspring, the distances between parent strings and resulting offspring for an SGA with a population size 500 were measured on two forms of NK-landscapes. The SGA was run for 250 generations. The SGA used elitism and tournament selection with a tournament size of 2. Selection pressure was kept relatively low and mutation ( $p_m=0.0125$ ) was used so the population would maintain some diversity throughout all 250 generations.

Kauffman’s NK-landscapes are a class of problems used in theoretical biology to study rugged fitness landscapes (Kauffman, 1993). NK-landscapes require two input parameters,  $N$  and  $k$ .  $N$  represents the number of bits and  $k$  controls the *epistatic interactions*, or bit interactions, in the problem. For each of the  $N$  variables, a set of  $k$  distinct variables are chosen to interact with that variable. So, there are  $N$  sets of  $k + 1$  variable combinations constrained so that each variable occurs in at least one set. In order to ensure that the variable combinations interact epistatically, a uniform random fitness between 0 and 1 is assigned to all possible  $2^{k+1}$  values for each set of variable combinations. These  $N2^{k+1}$  values are stored in a lookup table. An example of the lookup table for an NK-landscape with  $N=4$ ,  $k=1$  is:

Variables ( $v_0, v_1$ )	Settings for ( $v_0, v_1$ )			
	0 0	0 1	1 0	1 1
(0,1)	0.1	0.2	0.5	0.8
(1,2)	0.3	0.7	0.2	0.6
(2,0)	0.9	0.3	0.7	0.4
(3,1)	0.4	0.6	0.9	0.1

So each row in the table corresponds to a function that is enumerated over all possible values of the variables occurring as an ordered pair in the leftmost column. To evaluate an input string, the fitnesses of specific bit combinations are retrieved from the lookup table and averaged together. For example,

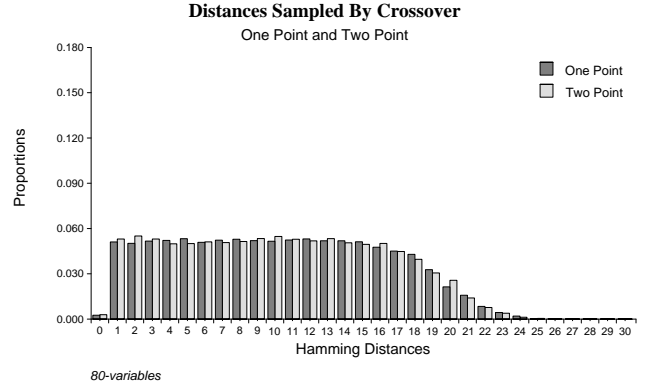


Figure 4: Distribution of step-sizes taken by one and two point crossover in the first generation of an SGA.

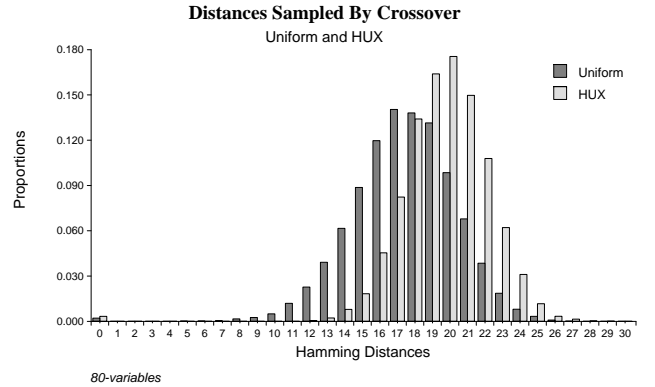


Figure 5: Distribution of step-sizes taken by uniform crossover and HUX in the first generation of an SGA.

$$f(0101) = \frac{1}{4}(0.5 + 0.7 + 0.4 + 0.4) = 0.5.$$

The bit interactions for NK-landscapes can be chosen randomly or fixed so all interactions occur at adjacent bits. Since the set of four crossover operators have different forms of positional biases, the method for choosing bit interactions can affect the relative performance of crossover. For this reason, two sets of 100 instances of NK-landscapes, with  $N = 80$  and  $k = 3$ , were used for the experiments.

To empirically verify that the distributional bias formulae are correct, the parent-offspring distances were tracked in the first generation and were accumulated for each problem in the set. The distances are tracked solely based on crossover before mutation was applied. Since the population size is 500 and there are 100 problems in each set, the number of samples per problem set is approximately  $50000 \times p_c$  and  $p_c = 0.6$  for all four crossover operators (where  $p_c$  is the probability that crossover occurs).

Figures 4 and 5 show the sampling distributions of the

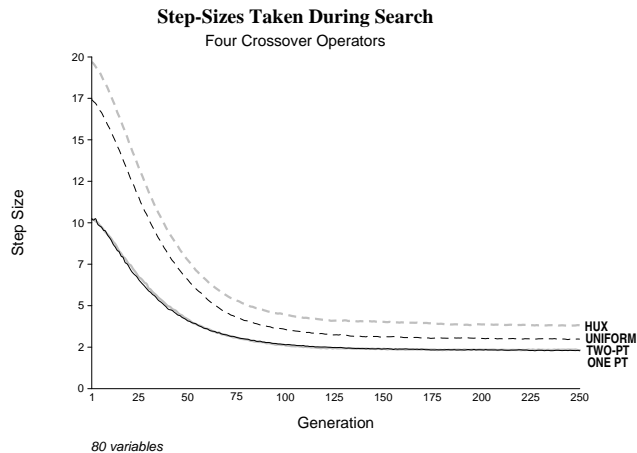


Figure 6: Average step sizes taken by crossover over 250 generations.

resulting parent-offspring distances. Again, the horizontal axis represents the set of Hamming distances and the vertical axis represents the proportion of the occurrence of the particular Hamming distances. For both graphs, the range of Hamming distances was truncated to 30 because Hamming distances larger than 30 were not encountered. Although the problem size is 80 bits rather than 20 bits, the distributions generated for the 20 variable exact sampling distributions are similar to those found empirically for 80 variable problems. These distributions were generated from the set of NK-landscapes with adjacent bit interactions; however, it should be noted that the distributions generated from the NK-landscapes with randomly chosen bit interactions were indistinguishable from these results.

Figure 6 tracks the average parent-offspring Hamming distances (or step-sizes) over 250 generations. Each point is the average of 100 problem instances. Again, the plot represents results from the set of NK-landscapes with adjacent bit interactions but the results are illustrative of the results for NK-landscapes with random bit interactions. Syswerda has noted that the parent distances in an initial population are approximately 50% of the string length, so crossover should exchange approximately 25% of the total number of bits (Syswerda, 1992). This graph illustrates that the actual number of *useful* swaps can be much lower than 25% of the string length and drops off quickly as search progresses. The use of a large population, small tournament size and mutation were intended to maintain diversity throughout execution, so the results are not an unreasonable indication of how quickly the exploratory power of crossover diminishes over time.

Operator	Random		Adjacent	
	$\mu$	$\sigma$	$\mu$	$\sigma$
Mutation	0.7528	0.01585	0.7474	0.01519
OnePt	0.7602	0.01454	0.7587	0.01503
TwoPt	0.7607	0.01546	0.7585	0.01488
Uniform	0.7622	0.01634	0.7548	0.01522
HUX	0.7629	0.01553	0.7540	0.01524

Table 2: Final results of SGA using four crossover operators on two types of NK-landscapes.

The relative performance of the various crossover operators varied between the two types of NK-landscapes as is illustrated in Table 2. The fitness values for NK-landscapes fall in the range  $[0, 1)$  and the SGA maximized fitness. Table 2 shows the mean and standard deviation for the best fitness found after 250 generations on both sets of 100 NK-landscapes. For the NK-landscapes with randomly chosen bit interactions, all crossover operators performed similarly. There were no statistically significant differences between the crossover operators. For the NK-landscapes with adjacent bit interactions, a one-way ANOVA statistical test confirmed that the crossover operators with positional bias performed significantly different from the crossover operators with no positional bias ( $p \leq 0.01$ ). A one-tailed t-test confirmed there is no significant difference between one and two point crossover; however, both uniform crossover and HUX performed significantly worse than either one or two point crossover ( $p \leq 0.05$ ).

For both sets of NK-landscapes, the crossover operators used in conjunction with mutation significantly outperform mutation alone. Since using crossover and mutation together always outperformed mutation alone regardless of the specific crossover operator or specific problem set, the distributional biases for each crossover were categorically beneficial for either set of NK-landscapes. The relative performance of the crossover operators may change if the algorithm or algorithm parameters were tuned to the specific problem sets; however, an assessment of the utility of crossover for any other purpose than a means of making "large steps" was beyond the scope of this paper.

## 4 CONCLUSION

It is not uncommon for algorithms to adapt search step-sizes so that a large step-size is used early in execution and decreases as search progresses. The most common means for a genetic algorithm to adapt its search step-size is through the use of crossover. When initialized with a random population, the genetic algorithm will sample points that are distant in Hamming

space according to the neighborhood sampling distribution for the specific crossover operator. The degree to which crossover continues to explore the search space depends on the amount of diversity in the population. In addition to population diversity, the exploratory power of crossover is limited by its inherent positional and distributional bias.

This paper presents exact calculations for the Hamming distances sampled by: one point, two point, uniform and HUX crossover operators. The calculations are representative of the sampling that occurs in the early generations for a genetic algorithm with a large population. The distributional bias for crossover is the Hamming distances between parents and offspring, so these calculations can be used to characterize the distributional bias for practical crossover operators.

The analytical results were verified empirically by examining the distances sampled by crossover during the execution of an SGA on two variants of NK-landscapes. The NK-landscapes results illustrate that the analytical formulae for computing the distribution of distances between parents and offspring accurately models the distributions of distances occurring in the first generation of a simple genetic algorithm. Additionally, the results illustrate that the large-steps taken by crossover resulted in improved performance over using mutation alone for both variants of NK-landscapes.

## Acknowledgments

Soraya Rana was funded by a National Physical Science Consortium fellowship.

## References

- [1] Booker, L. (1987). Improving search in genetic algorithms. In Davis, L., editor, *Genetic Algorithms and Simulated Annealing*, chapter 5, 61–73. Morgan Kaufmann.
- [2] Booker, L. B. (1992). Recombination distributions for genetic algorithms. In Whitley, D., editor, *Foundations of Genetic Algorithms - 2*, 29–44. Morgan Kaufmann.
- [3] Culberson, J. (1992). Genetic invariance: A new paradigm for genetic algorithm design. Technical Report TR-92-02, University of Alberta, Edmonton, Alberta, Canada.
- [4] Eshelman, L. (1991). The chc adaptive search algorithm. how to have safe search when engaging in nontraditional genetic recombination. In Rawlins, G., editor, *Foundations of Genetic Algorithms*, 265–283. Morgan Kaufmann.
- [5] Eshelman, L. J., Caruana, R. A., and Schaffer, J. D. (1989). Biases in the crossover landscape. In Schaffer, J. D., editor, *Proceedings of the Third International Conference on Genetic Algorithms*, 10–19.
- [6] Eshelman, L. J. and Schaffer, J. D. (1993). Crossover’s niche. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, 9–14.
- [7] Fogel, D. B. and Atmar, J. W. (1990). Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems. *Biological Cybernetics*, 63(2):111–114.
- [8] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing, Co., Reading, MA.
- [9] Jones, T. (1995). *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, Albuquerque, New Mexico.
- [10] Kauffman, S. A. (1993). *Origins of Order*. Oxford Press.
- [11] Papadimitriou, C. H. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall.
- [12] Reeves, C. R. (1994). Genetic algorithms and neighbourhood search. In T.C.Fogarty, editor, *Evolutionary Computing: AISB Workshop*, Leeds, UK. Springer-Verlag.
- [13] Spears, W. M. (1992). Crossover or mutation? In Whitley, L. D., editor, *Foundations of Genetic Algorithms - 2*, 221–238. Morgan Kaufmann.
- [14] Syswerda, G. (1989). Uniform crossover in genetic algorithms. In Schaffer, J. D., editor, *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann.
- [15] Syswerda, G. (1992). Simulated crossover in genetic algorithms. In Whitley, L. D., editor, *Foundations of Genetic Algorithms - 2*, 239–255.
- [16] Ulder, N., Aarts, E., Bandelt, H., Laarhoven, P. V., and Pesch, E. (1990). Genetic local search algorithms for the traveling salesman problem. In *Parallel Problem Solving from Nature*, 109–116, Dortmund, Germany.