92-4

A formal analysis of the role of multi-point crossover in genetic algorithms

Kenneth A. De Jong

Computer Science Department, George Mason University, Fairfax, VA 22030, USA

William M. Spears

AI Center – Code 5510, Naval Research Laboratory, Washington, DC, USA

On the basis of early theoretical and empirical studies, genetic algorithms have typically used 1 and 2-point crossover operators as the standard mechanisms for implementing recombination. However, there have been a number of recent studies, primarily empirical in nature, which have shown the benefits of crossover operators involving a higher number of crossover points. From a traditional theoretical point of view, the most surprising of these new results relate to uniform crossover, which involves on the average $L/2$ crossover points for strings of length L. In this paper we extend the existing theoretical results in an attempt to provide a broader explanatory and predictive theory of the role of multi-point crossover in genetic algorithms. In particular, we extend the traditional disruption analysis to include two general forms of multi-point crossover: n-point crossover and uniform crossover. We also analyze two other aspects of multi-point crossover operators, namely, their recombination potential and exploratory power. The results of this analysis provide a much clearer view of the role of multi-point crossover in genetic algorithms. The implications of these results on implementation issues and performance are discussed, and several directions for further research are suggested.

1. Introduction

One of the unique aspects of the work involving genetic algorithms (GAs) is the important role that recombination plays in the design and implementation of robust adaptive systems. In most GAs, individuals are represented by fixed-length strings and recombination is implemented by means of a crossover operator which operates on pairs of individuals (parents) to produce new strings (offspring) by exchanging segments from the parents' strings. Traditionally, the number of crossover points (which determines how many segments are exchanged) has been fixed at a very low constant value of 1 or 2. Support for this decision came from early work of both a theoretical and empirical nature [4,9].

However, there continue to be indications of an empirical nature that there are situations in which having a higher number of crossover points is beneficial

[6,11]. Perhaps the most surprising result (from a traditional perspective) is the effectiveness on some problems of uniform crossover, an operator which produces on the average $(L/2)$ crossings on strings of length L [11].

The motivation for this paper is to extend the theoretical analysis of the crossover operator to include the multi-point variations and provide a better understanding of when and how to exploit their power. We will concentrate on three different aspects of multi-point crossover operators: their disruptive effect, recombination potential, and exploratory power. Specifically, this paper will focus on two forms of multi-point crossover: n-point crossover and uniform crossover.

Sections 2 through 5 of this paper provide a thorough treatment of the disruptive effects of n-point and uniform crossover, while section 6 explains the role of disruption in the genetic algorithm. Section 7 describes, in detail, a new theory of recombination potential, based on the earlier disruption theory. Our thoughts on the exploratory power of n-point and uniform crossover are outlined in section 8. The implications of these theories are discussed in the final section.

2. Traditional analysis

Holland provided the initial formal analysis of the behavior of GAs by characterizing how they biased the makeup of new offspring in response to feedback on the fitness of previously generated individuals. By focusing on hyperplane subspaces of L-dimensional spaces (i.e., subspaces characterized by hyperplanes of the form "$-d_1-d_2-d_3-$", where the d_i are the defining positions of the hyperplane), Holland showed that the expected number of samples (individuals) allocated to a particular kth order hyperplane H_k at time $t+1$ is given by:

$$m(H_k, t+1) > m(H_k, t)\frac{f(H_k)}{\bar{f}}(1 - kP_m - P_c P_d(H_k)). \qquad (1)$$

In this expression, $f(H_k)$ is the average fitness of the current samples allocated to H_k, $\bar{f}$ is the average fitness of the current population, P_m is the probability of using the mutation operator, P_c is the probability of using the crossover operator, and $P_d(H_k)$ is the probability that the crossover operator will be "disruptive" in the sense that the children produced will not be members of the same subspace as their parents.

The usual interpretation of this result is that subspaces with higher than average payoffs will be allocated exponentially more trials over time, while those subspaces with below average payoffs will be allocated exponentially less trials. This assumes that there are enough samples to provide reliable estimates of hyperplane fitness, and that the effects of crossover and mutation are not too

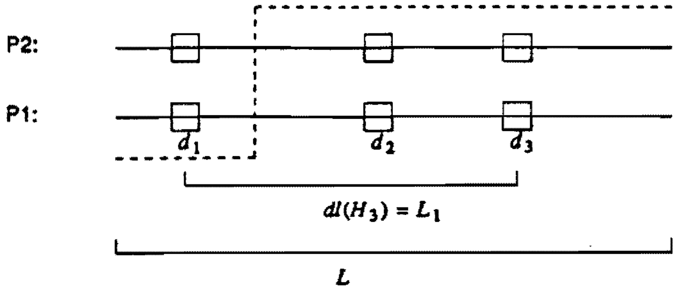

Fig. 1. A 3rd order hyperplane.

disruptive. Since mutation is typically run at a very low rate (e.g., $P_m = 0.001$), it is generally ignored as a significant source of disruption. However, crossover is usually applied at a very high rate (e.g., $P_c \geqslant 0.6$). So, considerable attention has been given to estimating P_d, the probability that a particular application of crossover will be disruptive.

To simplify and clarify the analysis, it is typically assumed that individuals are represented by fixed-length binary strings of length L, and that crossover points can occur with equal probability between any two adjacent bits. For ease of presentation these same assumptions will be made for the remainder of this paper. Generalizing the results to non-binary fixed-length strings is quite straightforward. Relaxing the other assumptions is more difficult.

Under these assumptions, Holland provided a simple and intuitive analysis of the disruption of 1-point crossover: as long as the crossover point does not occur within the defining boundaries of H_k (i.e., between any of the k fixed defining positions), the children produced from parents in H_k will also reside in H_k [9]. Figure 1 represents this graphically for a 3rd order hyperplane. Note that d_1, d_2, and d_3 represent the three defining positions of the 3rd order hyperplane, while P1 and P2 indicate the two parents. The dotted line represents 1-point crossover.

If crossover does occur inside the defining boundaries, disruption may or may not result. Disruption will depend on where the crossover point occurs inside the defining boundaries and on the alleles that the parents have in common on the k defining positions. Hence, P_d can be bounded by the probability that the crossover point will fall within the defining boundaries of H_k. Under the assumption of uniformly distributed crossover points, this yields:

$$P_d(H_k) \leqslant \frac{dl(H_k)}{L-1}, \tag{2}$$

where $dl(H_k)$ is the "defining length" of H_k, namely, the distance between the first and last of the k fixed defining positions of hyperplane H_k.

This analysis has led to considerable discussion of the "representational bias" built into 1-point crossover, namely, that crossover is much more disruptive to hyperplanes whose defining positions happen to be far apart. It also suggests a plausible role for inversion operators capable of effecting a change of representation in which the defining lengths of key hyperplanes are shortened.

De Jong [4] extended this analysis to n-point crossover by noting that no disruption can occur if there are an even number of crossover points (including 0) between each of the defining positions of a hyperplane. Hence, we have a bound for the disruption of n-point crossover:

$$P_d(n, H_k) \leqslant 1 - P_{k,even}(n, H_k). \tag{3}$$

Note that $P_{k,even}$ is defined to be the probability that an even number of crossover points will fall between each of the k defining positions. De Jong [4] provided an exact expression for $P_{k,even}$ for the special case of 2nd order hyperplanes (i.e., $k = 2$):

$$P_{2,even}(n, L, L_1) = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} \left(\frac{L_1}{L}\right)^{2i} \left(\frac{L - L_1}{L}\right)^{n-2i}. \tag{4}$$

$P_{2,even}(n, L, L_1)$ is the probability that an even number of crossover points will fall within the 2nd order hyperplane defined by L and L_1. Recall that L is the length of the string, while L_1 is the defining length of the hyperplane. The second term of the summation is the probability of placing an even number of crossover points within the two defining points. The third term is the probability of placing the remaining crossover points outside the two defining points. Finally, the combinatorial represents the number of ways an even number of points can be drawn from the n crossover points.

The family of curves generated by $P_{2,even}$ provide considerable insight into the change in disruptive effects on second order hyperplanes as the number of crossover points is increased. Figure 2 plots the curves for binary strings of length L. Notice how the curves fall into two distinct families depending on whether the number of crossover points is even or odd. Since $P_{2,even}$ guarantees no disruption, we are interested in increasing $P_{2,even}$ whenever possible. By going to an even number of crossover points, we can reduce the representational bias of crossover, but only at the expense of increasing the disruption of the shorter definition length hyperplanes.

If we interpret the area above a particular curve as measure of the cumulative disruption potential of its associated crossover operator, then these curves suggest that 2-point crossover is the best as far as minimizing disruption. These results together with early empirical studies were the basis for using 2-point crossover in many of the implemented systems. Since then, there have been several additional studies focusing on crossover.

Bridges and Goldberg [2] have extended Holland's analysis of 1-point crossover, deriving tighter bounds on the disruption by taking into account the

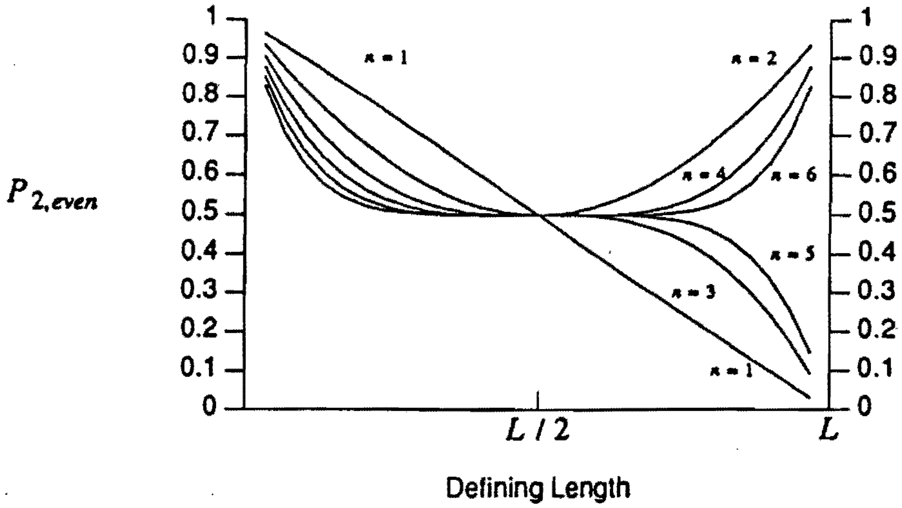

Fig. 2. n-point crossover disruption on 2nd order hyperplanes.

properties of the second parent and gains in samples in H_k due to disruption elsewhere.

Syswerda [11] introduced a "uniform" crossover operator in which P_0 specified the probability that the allele of any position in an offspring was determined by using the allele of the first parent, and $1 - P_0$ the probability of using the allele of the second parent. He provided an initial analysis of the disruptive effects of uniform crossover for the case of $P_0 = 0.5$, and compared it with 1 and 2-point crossover. He presented some provocative results suggesting that, in spite of higher disruption properties, uniform crossover can exhibit better recombination behavior, which can improve empirical performance.

Eschelman, Caruana, and Schaffer [6] analyze crossover operators in terms of "positional" and "distributional" biases, and present a set of empirical studies suggesting that no n-point, shuffle, or uniform crossover operator is universally better than the others.

These results and other empirical studies motivated us to attempt to clarify the effects of multi-point crossover by extending the current analysis. In the remaining sections we present a general analysis of crossover disruption, providing precise formulations for the disruption probabilities of both uniform and n-point crossover on kth order hyperplanes. Two other properties of crossover are also analyzed in detail: the recombination potential and exploratory power of both uniform and n-point crossover.

These theoretical extensions allow us to explain the apparent contradictory nature of recent empirical studies, and help us understand more clearly the interacting effects of crossover and other GA features such as population size. In addition, the analysis reemphasizes the important differences between the roles of crossover and mutation, and suggests the need for an adaptive crossover mechanism.

3. Crossover disruption for higher order hyperplanes

One possible explanation for the conflict between the early analysis and recent empirical results on the merits of having more crossover points is that the analysis involved only the special case of 2nd order hyperplanes. It is possible that, if properly extended to higher order hyperplanes, the theory would predict the benefits of a corresponding increase in the number of crossover points. In this section we attempt to resolve this issue by generalizing De Jong's disruption results to hyperplanes of arbitrary order.

As noted earlier, the disruption probability $P_d(n, H_k)$ of n-point crossover on a kth order hyperplane H_k can be conservatively bounded by $1 - P_{k,even}\ (n, H_k)$ where $P_{k,even}(n, H_k)$ is the probability that n-point crossover produces only an even number of crossover points between each of the defining positions of H_k.

De Jong's formula for calculating $P_{2,even}$ can be generalized by noting that $P_{k,even}$ can be defined recursively in terms of $P_{k-1,even}$. To see this, consider how $P_{3,even}$ can be calculated in terms of $P_{2,even}$. Figure 3 illustrates the approach graphically.

The probability of n-point crossover generating only an even number of crossover points between *both* d_1–d_2 and d_2–d_3 can be calculated by counting the number of ways an even number of crossover points can fall between d_1–d_3, and for each of these possibilities requiring an even number to fall in d_1–d_2 (a second order calculation involving L_1 and L_2). More formally, we have:

$$P_{3,even}(n, L, L_1, L_2) = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} \left(\frac{L_1}{L}\right)^{2i} \left(\frac{L - L_1}{L}\right)^{n-2i} P_{2,even}(2i, L_1, L_2). \tag{5}$$

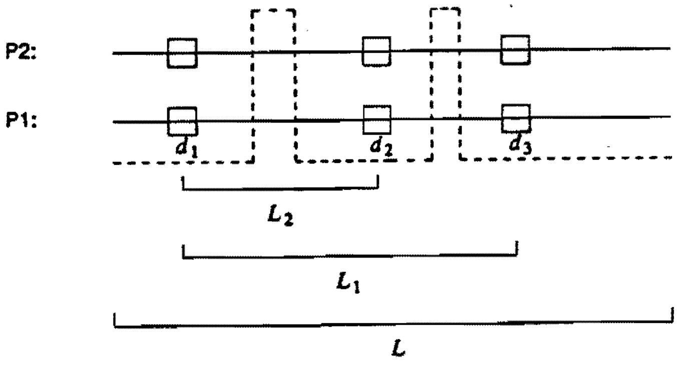

Fig. 3. Non-disruptive n-point crossover.

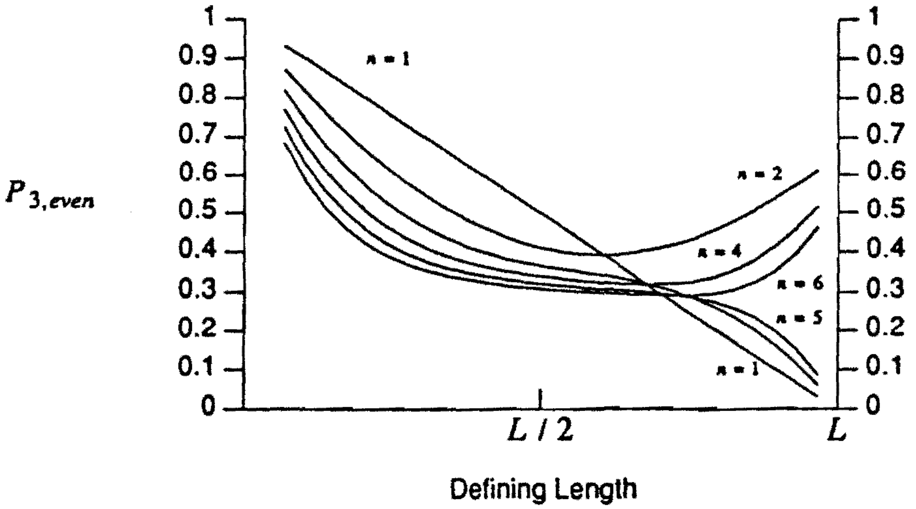

Fig. 4. $P_{k,even}$ on 3rd order hyperplanes.

In general, we have:

$$P_{k,even}(n, L, L_1, \ldots, L_{k-1}) = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} \left(\frac{L_1}{L}\right)^{2i} \left(\frac{L-L_1}{L}\right)^{n-2i} \times P_{k-1,even}(2i, L_1, \ldots, L_{k-1}). \tag{6}$$

Figures 4 and 5 illustrate $P_{k,even}$ for hyperplanes of order 3 and 5. Note that each point on the graph represents an average over all hyperplanes of a particular defining length. Note that, apart from a skewing effect, the curves

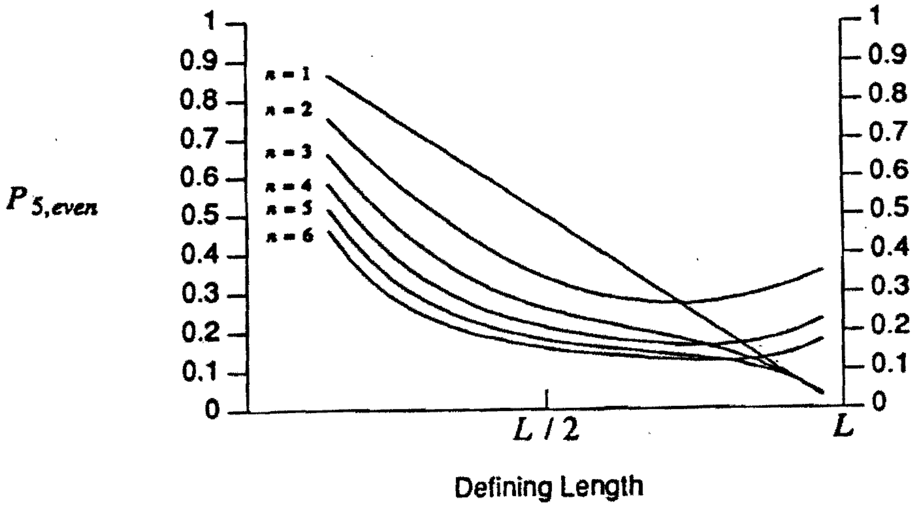

Fig. 5. $P_{k,even}$ on 5th order hyperplanes.

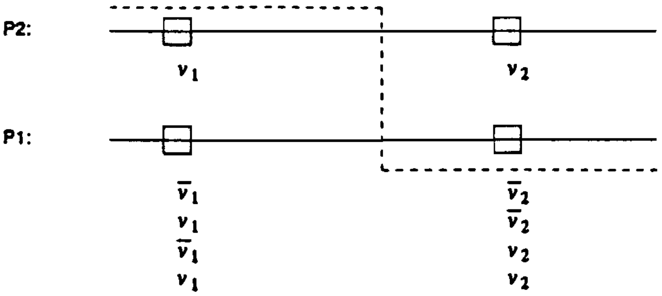

Fig. 6. Disruption in "odd" crossovers.

yield the same interpretation as De Jong's earlier curves for 2nd order hyperplanes: 2-point crossover minimizes disruption. So, extending the analysis thus far does not help in understanding the potential benefits of higher numbers of crossover points (seen in some empirical results).

4. Precise formulations of *n*-point disruption probabilities

Although the previous section generalized De Jong's initial disruption analysis, it did not explain the potential benefits of higher numbers of crossover points (seen experimentally). A second explanation for the conflicting results on the merits of a higher number of crossover points is that the $P_{k,even}$ curves are very weak bounds on P_d. It is possible that P_d itself, if computable, would yield different results. In this section we attempt to resolve this issue by providing exact equations for P_d.

The primary reason for the weakness of the $P_{k,even}$ bound is that it ignores the fact that many of the cases in which an odd number of crossover points fall between hyperplane defining positions are not disruptive to the sampling process. This occurs whenever the second parent happens to have identical alleles on the hyperplane defining positions which are exchanged by "odd" crossovers *. Figure 6 illustrates this in the simple case of 2nd order hyperplanes. Note that, in this figure, v_1 and v_2 represent the alleles (i.e., values) at those defining positions. Of the four possible combinations of matches on the defining positions of H_2, only the first $(-v_1-v_2-, -\bar{v}_1-\bar{v}_2-)$ actually results in a disruption.

Hence, in order to obtain precise disruption probabilities for an arbitrary hyperplane H_k, we must generalize $P_{k,even}$ to $P_{k,s}$ (i.e., the probability of *survival*) by including "odd" crossovers which are not disruptive. We achieve

* An "odd" crossover occurs when an odd number of crossover points falls within two adjacent defining positions of the hyperplane.

this in two steps: by first counting all possible odd crossovers, and then calculating which of those are not disruptive.

The probability that an odd number of crossover points will fall within a 2nd order hyperplane is given by:

$$P_{2,odd}(n, L, L_1) = \sum_{i=0}^{\lceil \frac{n}{2} \rceil} \binom{n}{2i-1} \left(\frac{L_1}{L}\right)^{2i-1} \left(\frac{L-L_1}{L}\right)^{n-(2i-1)}. \tag{7}$$

Let C be the probability that a particular odd crossover will not be disruptive. Then we can write:

$$P_{2,s}(n, L, L_1) = P_{2,even}(n, L, L_1) + CP_{2,odd}(n, L, L_1). \tag{8}$$

We can combine the contributions from the odd and even cases by defining C_s to be the probability that *any* particular crossover will not be disruptive. In this case:

$$P_{2,s}(n, L, L_1) = \sum_{i=0}^{n} \binom{n}{i} \left(\frac{L_1}{L}\right)^{i} \left(\frac{L-L_1}{L}\right)^{n-i} C_s. \tag{9}$$

Notice the similarity to eq. (4) defined earlier. In this case we are now considering all possible distributions of crossover points (both even and odd). In general:

$$P_{k,s}(n, L, L_1, \ldots, L_{k-1}) = \sum_{i=0}^{n} \binom{n}{i} \left(\frac{L_1}{L}\right)^{i} \left(\frac{L-L_1}{L}\right)^{n-i} \times P_{k-1,s}(i, L_1, \ldots, L_{k-1}). \tag{10}$$

We are left, then, with the problem of computing the correction factor C_s. We achieve this in the following manner. Given a particular distribution of crossover points, we can identify how many of the given hyperplane's defining positions are being exchanged by this particular crossover. If both parents match on these positions, no disruption occurs. For example, reconsider fig. 6. In this case the particular crossover will not disrupt the 2nd order hyperplane if the parents match on either the first or the second or both defining positions. If we let $P_{eq}(d)$ represent the probability that both parents have the same alleles on a particular defining position d, then

$$C_s = P_{eq}(d_1) + P_{eq}(d_2) - P_{eq}(d_1)P_{eq}(d_2) \tag{11}$$

specifies the probability that this crossover will not be disruptive. If we assume, for example, that $P_{eq}(d_i) = 0.5$ for both defining positions, then $C_s = 0.75$ which agrees with the prior discussion for fig. 6.

This same observation is true for kth order hyperplanes. We now must consider subsets X of the set of k defining positions. For the ease of presentation we denote K to be that set of k defining positions. Suppose that a

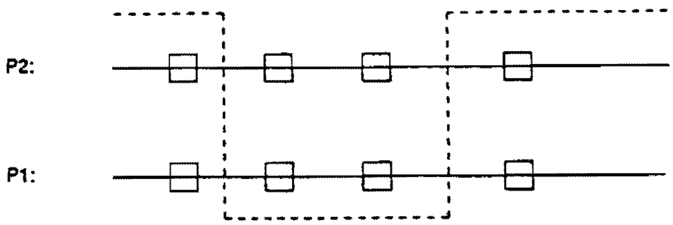

Fig. 7. Non-disruptive "odd" crossover on 4th order hyperplanes.

crossover results in a subset X of defining positions being exchanged. In this case no disruption will occur if: (1) the parents match on the subset X, or (2) if they match on the subset $K-X$, or (3) they match on the set K. Hence, the general form of the correction is:

$$C_s = \prod_{d \in X} P_{eq}(d) + \prod_{d \in K-X} P_{eq}(d) - \prod_{d \in K} P_{eq}(d). \tag{12}$$

If an even crossover occurs, then $|X| = 0$ and $C_s = 1$, as we would expect. However, for an odd crossover $C_s < 1$ (unless $P_{eq}(d) = 1$ for all $d \in K$). Figure 7 illustrates these ideas for one particular crossover on 4th order hyperplanes.

Deriving precise expressions for the values of $P_{eq}(d)$ at a particular point in time is difficult in general since they vary from generation to generation in complex, nonlinear, and interacting ways. We can, however, get considerable insight into the effects of shared alleles on disruption analysis by looking at several specific cases. Suppose, for simplicity, that for the hyperplane H_k of interest, $P_{eq}(d)$ is approximately the same for all the defining positions $d \in K$, i.e., $P_{eq}(d) = P_{eq}$. In this case the general form for C_s reduces to the simpler form:

$$C_s = P_{eq}^{|X|} + P_{eq}^{|K-X|} - P_{eq}^{|K|}. \tag{13}$$

In the case of the 4th order hyperplane in fig. 7, this becomes

$$C_s = P_{eq}^2 + P_{eq}^2 - P_{eq}^4. \tag{14}$$

If we assume $P_{eq} = 0.5$ (as is typical in the first few generations), then $C_s = (7/16)$ reflects the proportion of cases in which this particular crossover will not be disruptive.

Figures 8 and 9 show the effects of counting the non-disruptive "odd" crossovers in these special cases. Figure 8 assumes a value of $P_{eq} = 0.5$, which is likely to hold in the early generations when matches are least likely. Figure 9 assumes a value of $P_{eq} = 0.75$ to get a feeling of the effect as the population becomes more homogeneous. Note that in both cases, the amount of expected disruption has been significantly reduced and the relative difference in disruption among different numbers of crossover points is reduced as well. At the same time, note that the curves for the various number of crossover points have held their relative position with respect to one another.

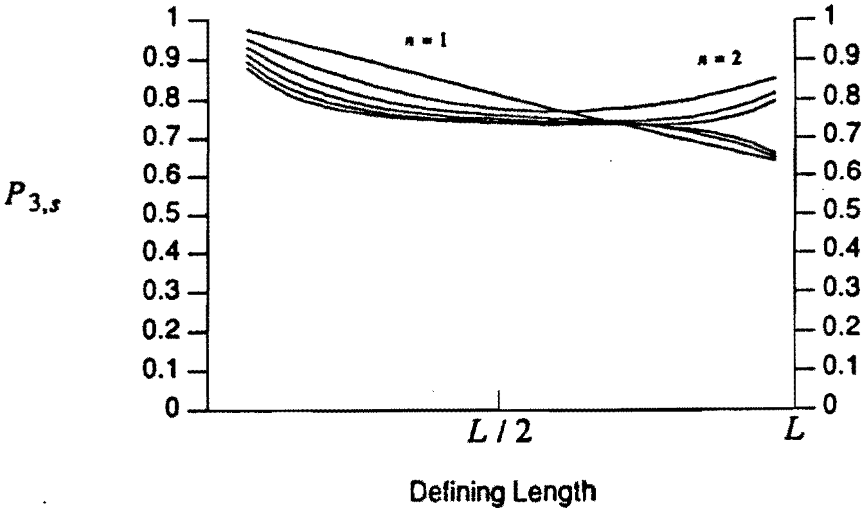

Fig. 8. $P_{k,s}$ on 3rd order hyperplanes with $P_{eq} = 0.5$.

These results help explain the fact that in some empirical studies little or no difference in effect is seen by varying the number of crossover points used in n-point crossover between, say, 1 and 16. It does not help explain why more crossover points might be significantly better, nor does it provide any insight into the relationship between n-point crossover and uniform crossover.

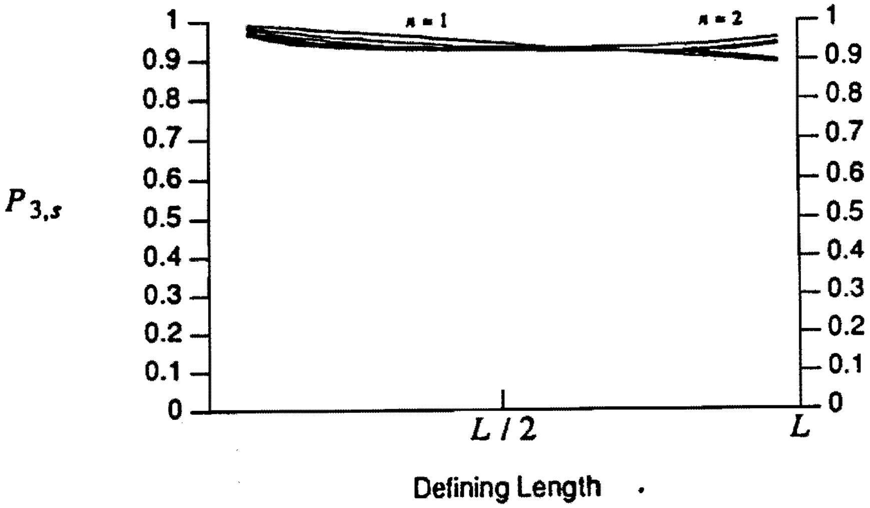

Fig. 9. $P_{k,s}$ on 3rd order hyperplanes with $P_{eq} = 0.75$.

5. Analyzing uniform crossover

Syswerda [11] defined a family of "uniform" crossover operators which is a variant of a notion that has been informally experimented with in the past: to produce offspring by randomly selecting at each locus the allele of one of the parents. By defining P_0 to be the probability of using the first parent's allele, offspring can be produced by flipping a P_0 biased coin at each position *.

A good way of relating uniform crossover to the more traditional n-point crossover is to think of uniform crossover as generating a mask of 0s and 1s, indicating which parent's allele is to be used at each position. As we scan the mask from left to right, a switch from 0 to 1 or from 1 to 0 represents a crossover point. For example, the mask 0011100 defines a 2-point crossover operation. If $P_0 = 0.5$, all masks are equally likely. If we examine the n-point crossover operations defined by this set of masks, we see immediately that they are binomially distributed around $((L-1)/2)$. For example, the set of all 8-bit masks results in the following distribution:

0-pt crossover: 0.008
1-pt crossover: 0.055
2-pt crossover: 0.164
3-pt crossover: 0.273
4-pt crossover: 0.273
5-pt crossover: 0.164
6-pt crossover: 0.055
7-pt crossover: 0.008

resulting in an average of 3.5 crossover points for an 8-bit mask.

If $P_0 \neq 0.5$, the masks are no longer uniformly distributed, but contain on the average longer runs of 0s or 1s. From the point of view of n-point crossover, the effect is to skew the binomial distribution toward 0. For example, setting $P_0 = 0.1$ changes the distribution of 8-bit masks to:

0-pt crossover: 0.430
1-pt crossover: 0.108
2-pt crossover: 0.323
3-pt crossover: 0.064
4-pt crossover: 0.064
5-pt crossover: 0.008
6-pt crossover: 0.003
7-pt crossover: 0.000

resulting in an average of 1.26 crossover points for an 8-bit mask.

* Other informal studies viewed the process as a random walk and defined P_0 as the probability of switching between parents. The two views are equivalent if and only if $P_0 = 0.5$.

We are now in a position to analyze the disruption properties of uniform crossover in the same manner as the analysis of n-point crossover in the preceding sections. The notion of an even number of crossover points between the defining positions of hyperplane H_k corresponds to masks which have either all 0s or all 1s on the defining positions of H_k. Hence, the corresponding conservative bound on the disruption of uniform crossover is given by:

$$P_d(H_k) \leqslant 1 - P_{k,even}(H_k), \tag{15}$$

where

$$P_{k,even}(H_k) = (P_0)^k + (1 - P_0)^k. \tag{16}$$

If $P_0 = 0.5$ for example, then

$$P_{k,even}(H_k) = (\tfrac{1}{2})^{k-1} \tag{17}$$

for all hyperplanes of order k. Notice that, unlike the traditional n-point crossover, there is no representational bias with uniform crossover in the sense that all hyperplanes of order k are equally disrupted regardless of how long or short their defining lengths are.

As before, we can obtain an exact formulation of P_d if we include non-disruptive "odd" crossovers. For uniform crossover this corresponds to those masks which are not either all 0s or all 1s on the hyperplane defining positions, but are non-disruptive because the parents share common alleles on those particular positions. If we let $|K|$ be the cardinality of the set K, and $PS(K)$ be the power set of K, then the probability of survival for uniform crossover is:

$$\begin{aligned} P_{k,s}(H_k) = {} & P_{k,even}(H_k) \\ & + \sum_{I \in PS(K) - K - \{\ \}} (P_0)^{|I|}(1 - P_0)^{|K-I|} \\ & \times \left[\prod_{d \in I} P_{eq}(d) + \prod_{d \in K-I} P_{eq}(d) - \prod_{d \in K} P_{eq}(d) \right], \end{aligned} \tag{18}$$

where P_{eq} is the probability of matching alleles, as before. Note that the last term in the expression is identical to the correction C_s defined earlier for the n-point crossover analysis. If the above is rewritten more concisely by combining the "even" and "odd" terms, $P_{k,s}$ can be expressed in a form similar to that derived for the n-point crossover analysis:

$$\begin{aligned} P_{k,s}(H_k) = {} & \sum_{I \in PS(K)} (P_0)^{|I|}(1 - P_0)^{|K-I|} \\ & \times \left[\prod_{d \in I} P_{eq}(d) + \prod_{d \in K-I} P_{eq}(d) - \prod_{d \in K} P_{eq}(d) \right]. \end{aligned} \tag{19}$$

As before, we can get considerable insight into these equations by examining the disruption on hyperplanes for which $P_{eq}(d) = P_{eq}$ for all $d \in K$. Figure 10 graphically illustrates this for 3rd order hyperplanes, and shows the relationship

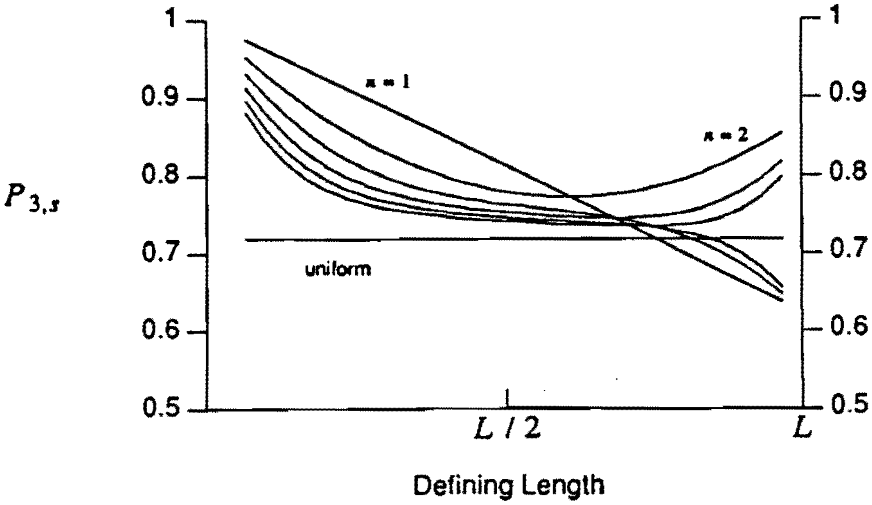

Fig. 10. $P_{k,s}$ on 3rd order hyperplanes with $P_{eq} = 0.5$.

between uniform crossover ($P_0 = 0.5$) and n-point crossover. Note that uniform crossover is much more disruptive than n-point crossover, but has the interesting property that its disruption is *independent* of hyperplane defining length.

This analysis also allows us to see clearly the effect of setting P_0 to something other than 0.5. Note that because of the symmetry of uniform crossover, P_0 and $1 - P_0$ produce the same effects. So, it is sufficient to consider the effects of decreasing P_0. As noted earlier, this has the effect of reducing the expected number of crossover points. One would therefore expect a reduction in disruption as well. Figure 11 illustrates this effect for 3rd order hyperplanes. Notice how the disruption of uniform crossover can be controlled by lowering P_0,

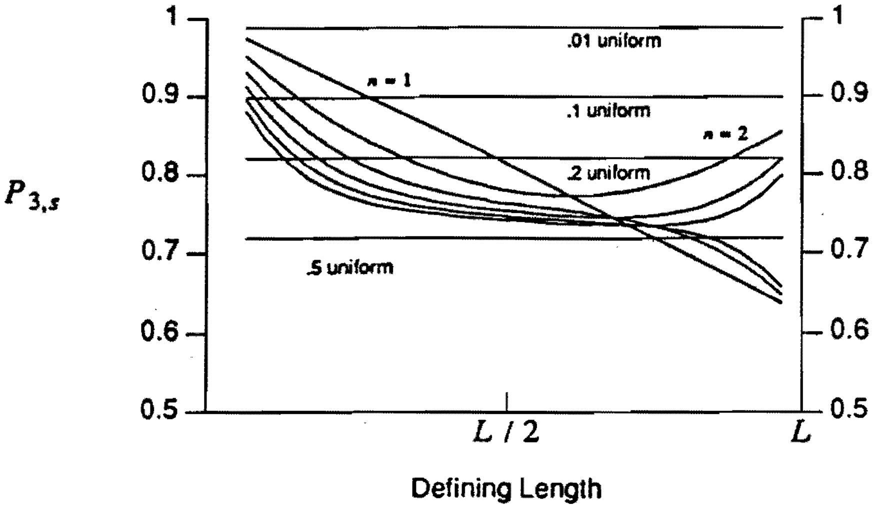

Fig. 11. Disruption of uniform crossover ($P_{eq} = 0.5$).

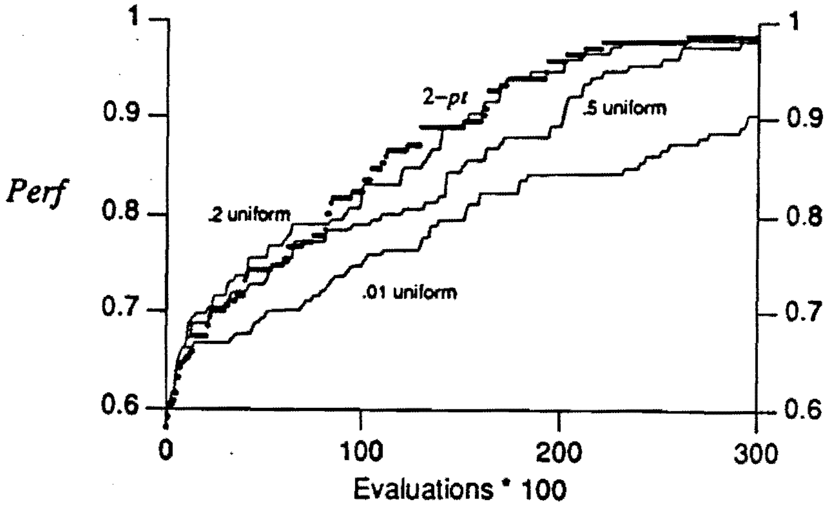

Fig. 12. 6-peak (30 bits) – population 1000.

without affecting the property that the disruption has no defining length bias. In particular, note that by simply lowering P_0 to 0.1, uniform crossover is less disruptive (overall) than 2-point crossover and has no defining length bias! This suggests a much more positive view of the potential of uniform crossover, namely, as an unbiased recombination operator whose disruption potential can be easily controlled by a single parameter P_0.

To illustrate the effect of disruption on performance, fig. 12 presents the results of running a GA on a 30-bit problem with 6 peaks (from [5]). The measure of performance is simply the best individual found at every generation. Since we are maximizing, higher curves represent better performance *. It is clear that the level of disruption provided by the standard form of uniform crossover ($P_0 = 0.5$) is too high for this problem and performance suffers relative to 2-point crossover. However, with uniform crossover we can increase and decrease performance on a given problem with a fixed population size simply by varying P_0. Note that in this particular case, a value of $P_0 = 0.2$ for uniform crossover results in performance similar to 2-point crossover. Referring back to fig. 11, we can now see why. On this particular problem uniform crossover with $P_0 = 0.2$ provides approximately the same level of disruption as 2-point crossover, but without the corresponding length bias.

One might be tempted to conclude that further reductions in disruption ($P_0 < 0.2$) would result in even better performance. However, fig. 12 indicates that, for this particular problem, performance is significantly degraded when $P_0 = 0.01$. This suggests that having a proper (but not necessarily minimal)

* All experimental results are averaged over 10 independent runs.

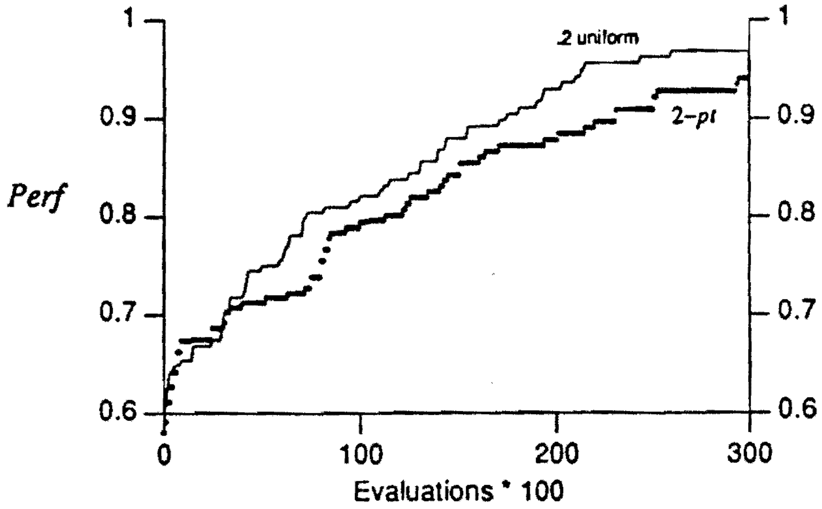

Fig. 13. 6-peak (30 + 270 fake bits) – population 1000.

crossover disruption rate is an important component in achieving good performance. We explore this theme in some detail in the next section.

The fact that the disruption rate of uniform crossover is independent of hyperplane defining length can be shown to be helpful in overcoming certain representational dependencies. Suppose an important hyperplane has a defining length of 30. If the string length L is 60, then the defining length of the hyperplane is $L/2$. However, if the string length is 300, then the defining length of the hyperplane is $L/10$. Figure 11 indicates that the disruptive effect of uniform crossover is the same in either situation, while the disruptive effect of n-point crossover changes.

Syswerda illustrated how this can clearly affect performance with his "sparse 1-max" problem in which 270 fake bits were appended to a 30-bit problem [11]. One can show similar results with almost any problem. Figure 13 illustrates this on our 6-peak problem problem appended with 270 fake bits. Note that, in comparison to the original 30-bit problem shown in fig. 12, the performance of uniform crossover ($P_0 = 0.2$) remains unchanged, while the performance of 2-point crossover is considerably worse after the 270 fake bits have been added. Referring back to fig. 11, we see that the effect of appending 270 fake bits is that the defining lengths of the hyperplanes involving the 30 important bits are now all $\leqslant L/10$ (L is the length of the string). 2-point crossover is less disruptive within this range (0 to $L/10$) of defining lengths and, in this case, the result is a less than optimal balance of exploration and exploitation.

In summary, we note that the analysis in this section has highlighted two important properties of uniform crossover. The first is the ease with which the disruptive effect of uniform crossover can be precisely controlled by varying P_0. The second important property is that the disruptive potential of uniform crossover does not depend on the defining length of hyperplanes. This allows uniform crossover to perform effectively regardless of the distribution of important alleles.

6. A more positive view of disruption

An implicit assumption of the previous sections was that disruption is a bad thing and should be minimized. If we take this point of view, that performance of GAs is inversely related to the amount of sampling disruption of crossover, then we have not achieved our goal of extending the existing theoretical analysis to explain the recent empirical results involving the benefits of crossover operators involving larger numbers of crossover points. The disruption analysis of the previous sections clearly shows that increasing the number of crossover points for both uniform and n-point crossover increases the disruption rates.

The escape from this apparent dilemma can be found in a recurring theme in Holland's work: the importance of a proper balance between exploration and exploitation when adaptively searching an unknown space for high performance solutions [9]. Because GAs involve finite populations, because selection pressure comes from observed (rather than actual fitness differentials), and because no a priori assumptions are made about the space being searched, a certain level of disruption is required (beneficial) to balance the pressure to exploit current observations.

This suggests the need for a broader theory which captures the effects of crossover in conjunction with other key aspects of GAs such as population size, mutation rate, and selection mechanisms. The initial pieces of such a theory are beginning to emerge, but are not as well developed as the disruption analysis. In this section we analyze two important situations in which crossover disruption interacts strongly with other aspects of GAs. The first relates to the evolving makeup of the population and the need for different levels of disruption at different points in the evolutionary process. The second involves the need for varying disruption levels as a function of population size and problem complexity.

6.1. CROSSOVER PRODUCTIVITY

Sampling disruption is important for understanding the effects of crossover when populations are diverse (typically early in the evolutionary process). However, when a population becomes quite homogeneous, another factor becomes important: whether the offspring produced by crossover will be different than their parents in some way (thus generating a new sample) or the same (clones).

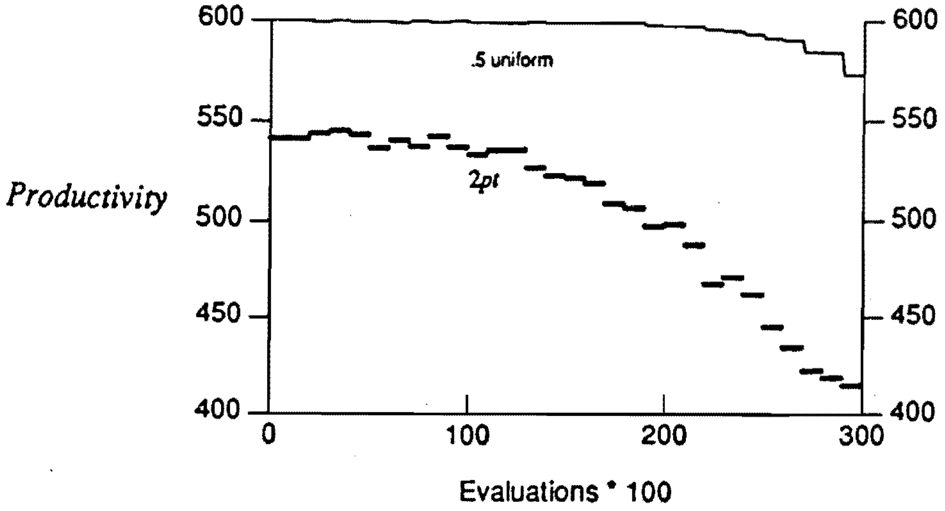

Fig. 14. Productivity on 6-peak (30 bits) – population 1000.

If we try to formally compute the probability that the offspring will be different than their parents, the computation is precisely the same as the previous disruption computations. To see this, consider two parents whose alleles differ on only four loci. In order for crossover to produce new offspring, some but not all of those alleles must be exchanged. The probability of this occurring is just $P_d(H_4)$. In other words, those operators that are more disruptive are also more likely to create new individuals from parents with nearly identical genetic material.

This property of crossover has been dubbed 'crossover productivity" and is easy to measure experimentally. Figure 14 illustrates how significantly the "productivity" of 2-point crossover can drop off as evolution proceeds on the 6-peak problem. The vertical axis indicates the number of crossovers, per 100 evaluations, that produced offspring different from their parents. Since $P_c = 0.6$, and the population size is 1000, the maximum productivity is 600. Note that, as expected, uniform crossover ($P_0 = 0.5$) is more productive than 2-point crossover, especially as the population loses diversity.

This observation helps explain some of the experimental results in which higher crossover rates performed well. Figure 12 illustrates how the added productivity of uniform crossover ($P_0 = 0.5$) helps performance at the end of a run (i.e., when the population loses diversity). A more dramatic example is shown in fig. 15. Notice that, in this case, 2-point crossover is surpassed by uniform crossover.

This analysis strongly suggests the need for an adaptive crossover operator whose disruption rate changes with the context of the evolving population dynamics. We have some empirical results to support this (e.g., [1]). However, a

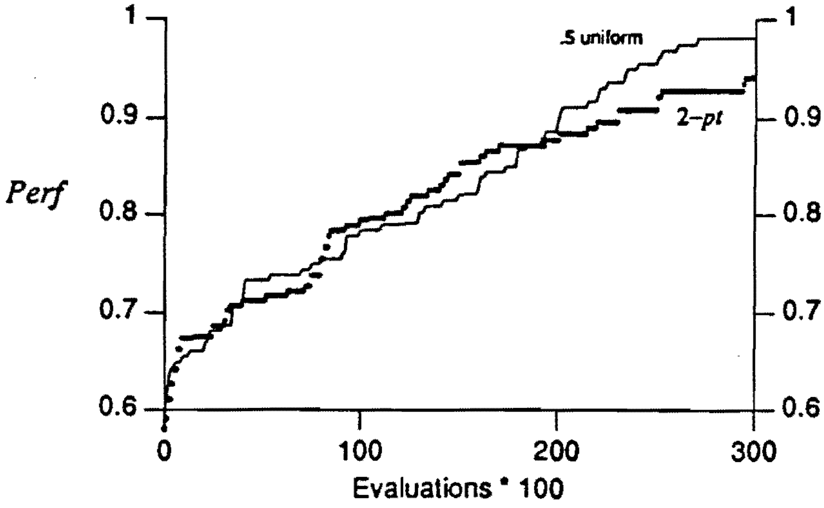

Fig. 15. 6-peak (30+270 fake bits) – population 1000.

formal analysis specifying the "balance equations" involving crossover disruption and population homogeneity is not yet available.

6.2. POPULATION SIZE EFFECTS

It is also quite straightforward to show that additional exploration is required when the population size is too small to provide the necessary sampling accuracy for complex search spaces [5]. Figures 12 and 16 illustrate this effect on the 6-peak problem. Notice how uniform crossover dominates 2-point crossover on

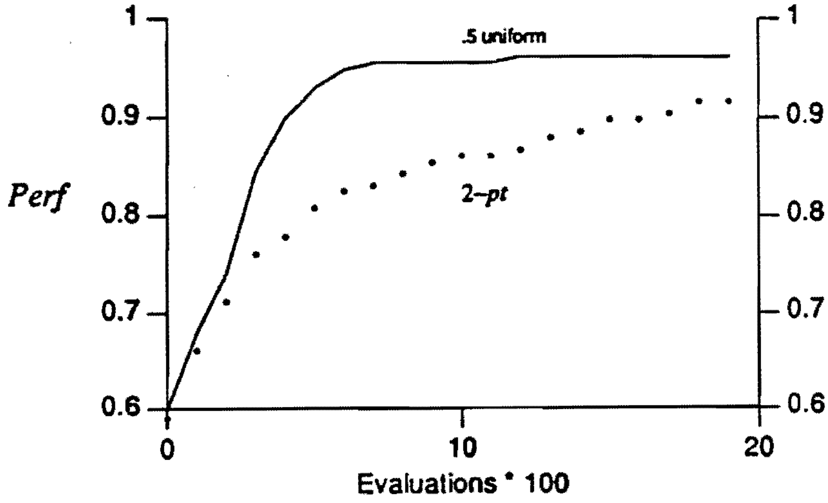

Fig. 16. 6-peak (30 bits) – population 20.

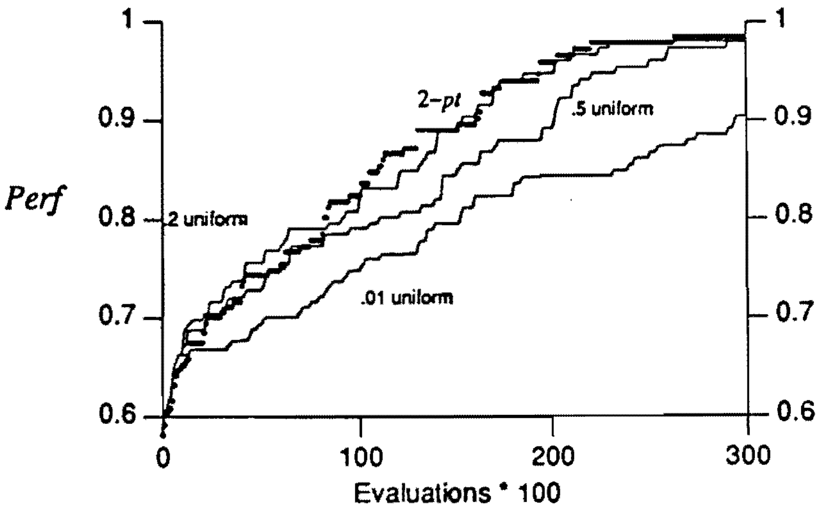

Fig. 17. 6-peak (30 bits) – population 1000.

the 6-peak problem with a small population, but just the opposite is true with a large population.

Figure 16 illustrates the importance of high disruption when small population sizes are used. Since small populations tend to converge quickly, high disruption helps to maintain exploration. Since larger populations converge less quickly, disruption is less useful. However, despite a larger population, it is still possible to have too little disruption. Consider our 6-peak problem in which uniform crossover is run with a P_0 of 0.01. The results are shown in fig. 17. Notice that a reduction of P_0 from 0.2 to 0.01 is detrimental to performance, although the reduction from 0.5 to 0.2 is beneficial. This reemphasizes the need for a *balance* between exploration and exploitation in order to achieve efficient search, and points out that problem complexity and population size can change the balance point.

These results go a long way towards explaining the sometimes contradictory results of many empirical studies concerning the use of uniform and *n*-point crossover. It does not, however, provide a strong predictive theory relating specific population sizes to appropriate crossover disruption rates. One of the difficulties in developing such a theory is that the properties of the space being searched affect the relationships which need to be expressed. If the important properties of the search space are assumed to be unavailable a priori, these results strongly suggest the need for adapting population size, disruption rates, etc. to the characteristics of the search space. Again, we have some tantalizing empirical evidence of the benefits of such an approach (e.g., [3]), but a theoretical analysis at this level is still a long way off.

6.3. MUTATION AS A SOURCE OF DISRUPTION

An obvious question at this point is how the sampling disruption provided by mutation affects the balance and choice of crossover operators. There are a significant number of empirical studies in which GA performance is improved by increasing and/or adapting the mutation rate (see, or example, [7]). Currently, there is no theory which captures the interrelationship precisely. Clearly, all of the issues raised earlier come into play, including population size, selection pressure, and the properties of the search space.

However, it is important to understand that the nature of the disruption produced by crossover is significantly different than that of mutation. Crossover disruption is better viewed and understood as a side effect of two important properties of crossover, namely its recombination potential and its exploratory power. Focusing on these two "positive" aspects of crossover gives a clearer picture of its role in GAs. We analyze these features in the next two sections.

7. Crossover recombination potential

The previous sections have described, in detail, the disruptive effects of n-point and uniform crossover. A more positive approach to analyzing crossover operators is to focus on their recombination potential, which is defined as the ability of crossover to create higher order hyperplanes when the parents contain the necessary lower order hyperplanes (building blocks). Syswerda pointed out that recombination can be considered to be a specialized form of survival, in which two lower order hyperplanes survive onto the same string, resulting in a higher order hyperplane [11]. This observation allowed Syswerda to construct a recombination analysis from his survival analysis. His analysis shows uniform crossover ($P_0 = 0.5$) to have a higher recombination potential than 1 and 2-point crossover.

However, since this survival analysis was limited to 1 and 2-point crossover, and to uniform crossover with a P_0 of 0.5, the recombination analysis was similarly limited. This motivated us to create a new recombination analysis in a similar vein, since our survival analysis includes all of n-point crossover and a parameterized uniform crossover. In the following discussion we will consider the creation of a kth order hyperplane from two hyperplanes of order m and n. We will restrict the situation such that the two lower order hyperplanes are non-overlapping, and $k = m + n$. Each lower order hyperplane is in a different parent. We denote the probability that the kth order hyperplane will be recombined from the two hyperplanes as $P_{k,r}$.

7.1. RECOMBINATION ANALYSIS FOR n-POINT CROSSOVER

Since recombination is a restricted form of survival, it is appropriate to reexamine our previous survival equations (eqs. (9) and (10)). Recall that the

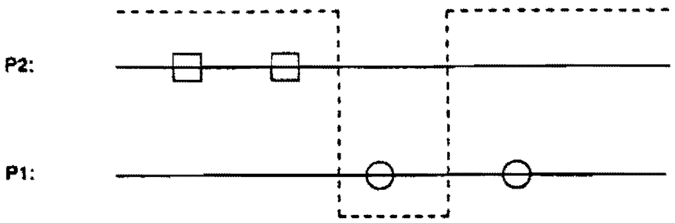

Fig. 18. Recombination of two 2nd order hyperplanes.

equations correctly consider all possible crossover distributions (both even and odd), while the correction factor C_s computes the probability that disruption will not occur, given a particular distribution. An analysis of recombination under n-point crossover is simple if one modifies C_s. Recall that recombination will occur if both lower order hyperplanes survive in the same individual. Suppose that an n-point crossover results in a subset X of the k defining positions surviving in the same individual. In this case recombination will occur if: (1) the parents match on the subset X, or (2) if they match on the subset $K-X$, or (3) they match on the set K. Hence, the general form of the correction is:

$$C_r = \prod_{d\in X} P_{eq}(d) + \prod_{d\in K-X} P_{eq}(d) - \prod_{d\in K} P_{eq}(d). \tag{20}$$

As an example, consider fig. 18. In this figure, we represent the recombination of two 2nd order hyperplanes. One hyperplane is depicted with circles, and the other with squares. Since three of the defining positions will survive onto the same individual (under the 2-point crossover shown), the probability of recombination, assuming $P_{eq}(d) = P_{eq}$ for all $d \in K$, is:

$$C_r = P_{eq}^1 + P_{eq}^3 - P_{eq}^4. \tag{21}$$

Note the similarity in description with the survival correction factor C_s (the only difference is in how X is defined). In other words, given a kth order hyperplane, and two hyperplanes of order n and m, $P_{k,r}$ is simply $P_{k,s}$ with the correction factor redefined as above.

7.2. RECOMBINATION ANALYSIS FOR UNIFORM CROSSOVER

The analysis of recombination under uniform crossover also involves the analysis of the original survival equation (see eq. (19)). Note that, due to the independence of the operator (each allele is swapped with probability P_0), eq. 19 can be divided into three parts. The first part expresses the probability that a hyperplane will survive in the original string:

$$P_{k,s,orig}(H_k) = \sum_{I\in PS(K)} (P_0)^{|I|}(1-P_0)^{|K-I|} \prod_{d\in K-I} P_{eq}(d). \tag{22}$$

The second part expresses the probability that a hyperplane will survive in the other string:

$$P_{k,s,other}(H_k) = \sum_{I \in PS(K)} (P_0)^{|I|}(1-P_0)^{|K-I|} \prod_{d \in I} P_{eq}(d). \tag{23}$$

The final part expresses the probability that a hyperplane will exist in both strings:

$$P_{k,s,both}(H_k) = \sum_{I \in PS(K)} (P_0)^{|I|}(1-P_0)^{|K-I|} \prod_{d \in K} P_{eq}(d). \tag{24}$$

Then:

$$P_{k,s}(H_k) = P_{k,s,orig}(H_k) + P_{k,s,other}(H_k) - P_{k,s,both}(H_k). \tag{25}$$

This formulation allows us to express recombination under uniform crossover. Again, assuming the recombination of two non-overlapping hyperplanes of order n and m into a hyperplane of order k:

$$\begin{aligned} P_{k,r}(H_k) = {} & P_{m,s,orig}(H_m)P_{n,s,other}(H_n) + P_{m,s,other}(H_m)P_{n,s,orig}(H_n) \\ & - P_{m,s,both}(H_m)P_{n,s,both}(H_n). \end{aligned} \tag{26}$$

This equation reflects the decomposition of recombination into two independent survival events. The first term is the probability that H_m will survive on the original string, while H_n switches (i.e., both hyperplanes survive on one parent). The second term is the probability that both hyperplanes survive on the other parent. The third term reflects the joint probability that both hyperplanes survive on both strings, and must be subtracted.

Figure 19 illustrates the relationships of the crossover operators in terms of their recombination potential. Note specifically that there is evidence to support

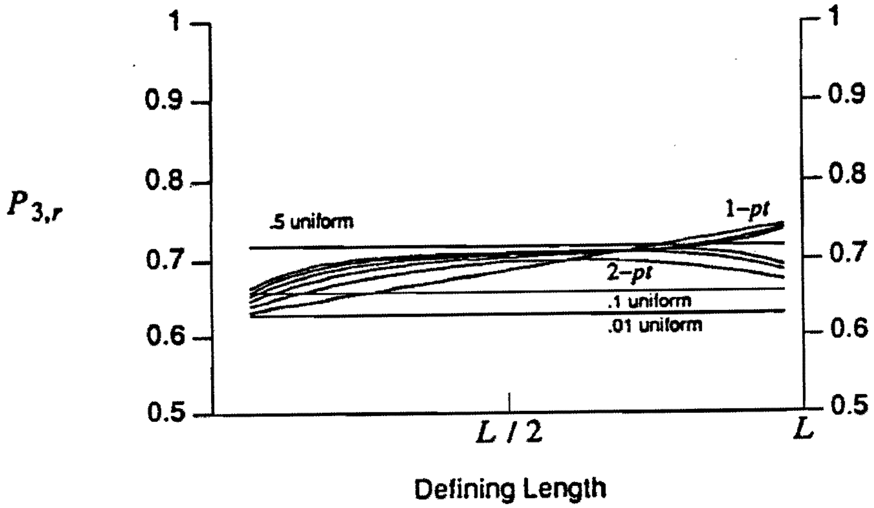

Fig. 19. Recombination of 3rd order hyperplanes ($P_{eq} = 0.5$).

the claim that uniform crossover ($P_0 = 0.5$) has a higher recombination potential than the other crossover operators. However, it is even more interesting to note that these relationships are qualitatively identical to those shown in fig. 11. In other words, if one operator is better than another for survival, it is worse for recombination (and vice versa). This observation appears to hold for all k, and suggests very strongly that one cannot increase the recombination power without a corresponding increase in disruption. It also suggests that, because of this direct relationship, both recombination and crossover disruption rates can be controlled by a single mechanism.

8. Crossover exploration power

Throughout this paper we have viewed disruption as a means for maintaining a balance between exploration and exploitation during adaptive search. It has also been pointed out that disruption does not necessarily imply useful exploration. The disruption caused by crossover and mutation simply implies that a hyperplane sample has been modified in some way so as to no longer be a member of that hyperplane, without any indication as to the possible forms that change might take. The potential number of ways in which a genetic operator can effect a change has been called its exploratory power. It is clear that crossover, in general, can make significantly broader changes to an individual than mutation can, thus providing for a more radical, but more performance-biased form of disruption.

Recently, it has been pointed out that uniform crossover has the additional property that it has more exploratory power than n-point crossover [6]. To see that this is true, consider the extreme case in which one parent is a string of all 0s and the other all 1s. Clearly uniform crossover can produce offspring anywhere in the space while 1 and 2-point crossover are restricted to rather small subsets. In general, uniform crossover is much more likely to distribute its disruptive trials in an unbiased manner over larger portions of the space.

The difficulty comes in analyzing whether this exploratory power is a virtue. If we think of exploitation as the biased component of the adaptive search process, it makes sense to balance this with unbiased exploration. Clearly, this exploratory power can help in the early generations, particularly with smaller population sizes, to make sure the whole space is well sampled. It can help avoid some "deceptive" traps presented by certain classes of problems. At the same time, some of this same exploratory power can be achieved over several generations via repeated applications of 1 and 2-point crossover. Unfortunately, our current analysis tools do not allow us to make comparisons of properties which span generations and are strongly affected by selection. Hopefully we will develop such tools and resolve questions of this type in the near future.

9. Conclusions and further work

The extensions to the analysis of n-point and uniform crossover presented in this paper provide considerable additional insight into the role of multi-point crossover in genetic algorithms. One of the unanticipated results of this effort is the surprisingly positive view of uniform crossover that emerges. There appear to be three potentially important features of uniform crossover. First, the disruption/recombination of hyperplane sampling under uniform crossover does not depend on the defining length of the hyperplanes. This reduces the possibility of representation effects, since there is no defining length bias. Second, the disruption potential is easily controlled via a single parameter P_0. This suggests the need for only one crossover form (uniform crossover), which is adapted to different situations by adjusting P_0. Finally, when a disruption does occur, uniform crossover results in a minimally biased exploration of the space being searched.

The paper, however, raises another issue: the need for a broader theory capable of indicating how to achieve a proper balance of exploration and exploitation for a particular problem by appropriate choice of population size, genetic operator rates, and selection pressure. A full theoretical analysis of this appears to be quite far off. The authors are currently working on extending the results presented here in that direction. Our goal is to understand these interactions well enough so that GAs can be designed to be self-selecting with respect to such decisions as optimal population size and the proper balance between exploration and exploitation.

Acknowledgements

We would like to thank the reviewers and the many other members of the GA community for their helpful comments and criticisms as the research presented here evolved to its current state. We would also like to thank Diana Gordon for her help in our preliminary recombination analysis. Finally, I would like to thank my father, William C. Spears, for creating within me a desire to understand intelligence, from his knowledge of and interest in all facets of psychology.

References

[1] L.B. Booker, Improving search in genetic algorithms, *Genetic Algorithms and Simulated Annealing*, ed. L. Davis (Morgan Kaufmann, 1987).

[2] C. Bridges and D. Goldberg, An analysis of reproduction and crossover in a binary-coded genetic algorithm, *Proc. 1st Int. Conf. on Genetic Algorithms*, ed. J.J. Grefenstette (Lawrence Erlbaum, 1985).

[3] L.D. Davis, Adapting operator probabilities in genetic algorithms, *Proc. 3rd Int. Conf. on Genetic Algorithms*, ed. J.D. Shaffer (Morgan Kaufmann, 1989).

[4] K.A. De Jong, An analysis of the behavior of a class of genetic adaptive systems, Doctoral Thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor (1975).

[5] K.A. De Jong and W. Spears, An analysis of interacting roles of population size and crossover in genetic algorithms, *Proc. 1st Int. Conf. on Parallel Problem Solving from Nature*, eds. H.P. Schwefel and R. Manner (Springer, 1990).

[6] L. Eschelman, R. Caruana and D. Schaffer, Biases in the crossover landscape, *Proc. 3rd Inf. Conf. on Genetic Algorithms*, ed. J.D. Shaffer (Morgan Kaufmann, 1989).

[7] T.C. Fogarty, Varying the probability of mutation in genetic algorithms, *Proc. 3rd Int. Conf. on Genetic Algorithms*, ed. J.D. Shaffer (Morgan Kaufmann, 1989).

[8] D.E. Goldberg, Sizing populations for serial and parallel genetic algorithms, *Proc. 3rd Int. Conf. on Genetic Algorithms*, ed J.D. Shaffer (Morgan Kaufmann, 1989).

[9] J.H. Holland, *Adaptation in Natural and Artificial Systems* (The University of Michigan Press, 1975).

[10] W. Spears and K.A. De Jong, An analysis of multi-point crossover, *Proc. Foundations of Genetic Algorithms Workshop*, ed. G. Rawlins (Morgan Kaufmann, 1990).

[11] G. Syswerda, Uniform crossover in genetic algorithms, *Proc. 3rd Int. Conf. on Genetic Algorithms*, ed. J.D. Shaffer (Morgan Kaufmann, 1989).