Adapting Crossover in a Genetic Algorithm

William M. Spears
Naval Research Laboratory
Washington, D.C. 20375 USA
spears@aic.nrl.navy.mil

Abstract

Traditionally, genetic algorithms have relied upon 1 and 2-point crossover operators. Many recent empirical studies, however, have shown the benefits of higher numbers of crossover points. Some of the most intriguing recent work has focused on uniform crossover, which involves on the average $L/2$ crossover points for strings of length L. Despite theoretical analysis, however, it appears difficult to predict when a particular crossover form will be optimal for a given problem. This paper describes an adaptive genetic algorithm that decides, as it runs, which form is optimal.

Keywords: Genetic algorithm theory, recombination operators, adaptation

1 Introduction

One of the unique aspects of the work involving genetic algorithms (GAs) is the important role that recombination plays. In most GAs, recombination is implemented by means of a crossover operator which operates on pairs of individuals (parents) to produce new offspring by exchanging segments from the parents' genetic material. 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 [Holland, 1975; De Jong, 1975]. However, there continue to be indications that there are situations in which having a higher number of crossover points is beneficial [Syswerda, 1989;

Eschelman, 1989]. 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 [Syswerda, 1989].

Recent work by [Spears and De Jong, 1990] has extended the theoretical analysis of n-point and uniform crossover with respect to disruption of sampling distributions. However, they pointed out that disruption analysis alone is not sufficient in general to predict and/or select optimal forms of crossover. In particular, they have shown that the population size must also be taken into account [De Jong and Spears, 1990].

Since the theory is inadequate for selecting optimal crossover forms (*before* the genetic algorithm is initiated) it is reasonable to ask whether this selection can be achieved via other means *while* the genetic algorithm is running. Earlier work by Schaffer and Davis suggest that it is possible for the genetic algorithm itself to choose optimal crossover forms as it runs. This paper also investigates this possibility, in a fairly restricted form, by examining whether a genetic algorithm can choose between 2-point crossover and uniform crossover as the problem is being solved.

This paper is organized in the following manner. First, we review the current theory on disruption analysis, with an emphasis on the differences between 2-point and uniform crossover. Second, we provide some motivation for and historical background of adaptive mechanisms that manipulate genetic operators. Third, we introduce a simple adaptive mechanism, followed by experimental results and an analysis of these results. Finally, we conclude with a discussion and directions for future research in this area.

2 Disruption Analysis

Holland provided the initial formal analysis of the behavior of GAs by showing how they allocate trials in a near optimal way to competing low order hyperplanes if the disruptive effects of the genetic operators used is not too severe [Holland, 1975]. Since mutation is typically run at a very low rate, it is generally ignored as a significant source of disruption. However, crossover is usually applied at a very high rate. So, considerable attention has been given to estimating P_d, the probability that a particular application of crossover will be disruptive.

Holland's initial analysis of the sampling disruption of 1-point crossover [Holland, 1975] has been extended to n-point and uniform crossover [De Jong, 1975; Spears and De Jong, 1990]. These results are in the form of estimates of the likelihood that the sampling of a kth order hyperplane (H_k) will be disrupted by a particular form of crossover. It turns out to be easier mathematically to estimate the complement of disruption: the likelihood of a sample surviving crossover (which we denote as P_s). As one might expect, the results are a function of both the order k of the hyperplane and its defining length (see [Spears and De Jong, 1990] for more precise details).

We provide in Figure 1 a graphical summary of a typical instance of these results for the case of 3rd order hyperplanes. The non-horizontal curves represent the survival of 3rd order hyperplanes under n-point crossover (n = 1...6). The horizontal line represents the probability of survival under uniform crossover. Figure 1 highlights two important points. First, if we interpret the area *above* a particular curve as a measure of the cumulative disruption potential of its associated crossover operator, then these curves suggest that 2-point crossover is the least disruptive of the crossover operators, while uniform crossover is the most disruptive. Finally, unlike n-point crossover, uniform crossover disrupts all hyperplanes of order k with equal probability, regardless of how long or short their defining lengths are.

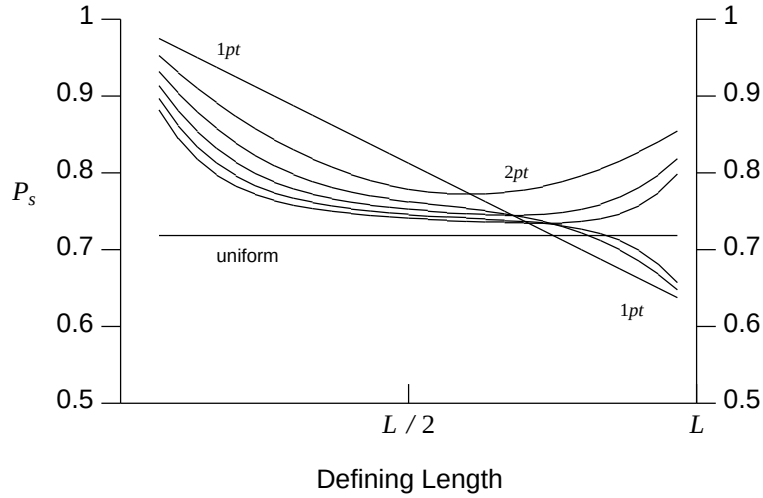

Figure 1: Survival of 3rd Order Hyperplanes

3 A Positive View of Crossover Disruption

A recurring theme in Holland's work is the importance of a proper balance between exploration and exploitation when adaptively searching an unknown space for high performance solutions [Holland, 1975]. The disruption analysis of the previous section implicitly assumes that disruption of the sampling distributions is a bad thing and to be avoided (e.g., a high disruption may stress exploration at the expense of exploitation). However, this is not always the case. There are important situations in which minimizing disruption hinders the adaptive search process by overemphasizing exploitation at the expense of needed exploration. One of the clearest examples of this is when the population size is too small to provide the necessary sampling accuracy for complex search spaces [De Jong and Spears, 1990].

To illustrate this we have selected a 30 bit problem with 6 peaks from [De Jong and Spears, 1990]. The measure of performance is simply the best individual found by the genetic algorithm. This is plotted every 100 evaluations. Since we are maximizing, higher curves represent better performance. Figures 2 and 3 illustrate the effect of population size on GA performance. Notice how uniform crossover dominates 2-point crossover on the 6-Peak problem with a small population, but just the opposite is true with a large population.†

There is another situation in which minimizing disruption hinders the adaptive search process, namely, when the population loses diversity. As the population of a genetic algorithm evolves over time, it generally loses genetic diversity and converges towards a common area of the search space. It is precisely at those times that exploration is needed, and a low amount of disruption will hinder the search.

† All the experimental results have been averaged over 30 runs.

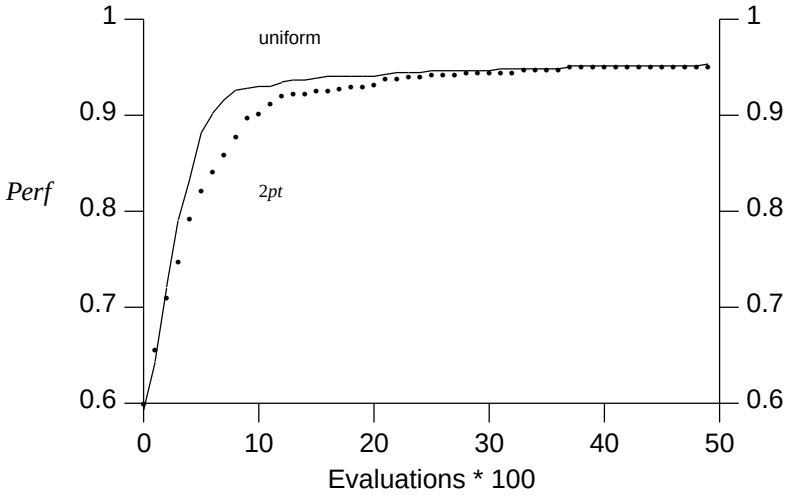

Figure 2: 6-Peak (30 bits) - Population 20

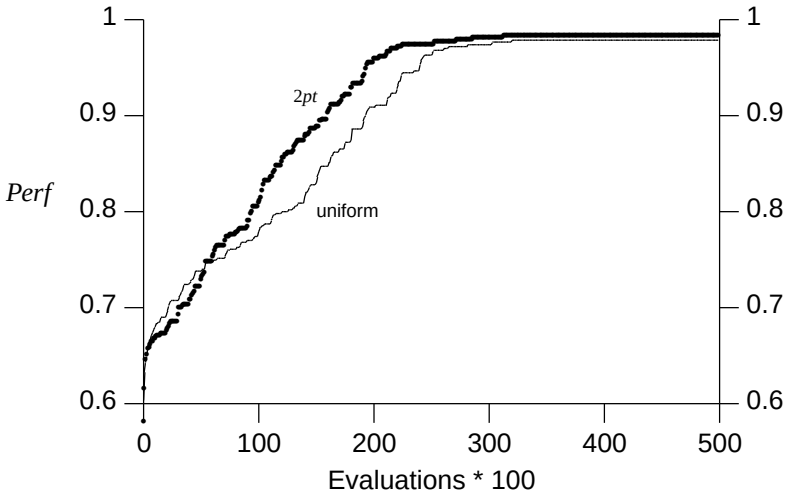

Figure 3: 6-Peak (30 bits) - Population 1000

To illustrate this effect we have selected a 45 bit Hamiltonian Circuit problem from [Spears, 1990]. Again, the measure of performance is the best individual found by the genetic algorithm, which is plotted every 100 evaluations. Figure 4 illustrates the behavior of 2-point crossover and uniform crossover on this particular problem. It is interesting to note that although 2-point crossover outperforms uniform in the beginning of the search (which is as expected, since the population is large), it is eventually

overtaken by uniform crossover. For this particular problem the added disruption of uniform crossover is beneficial towards the end of the runs.

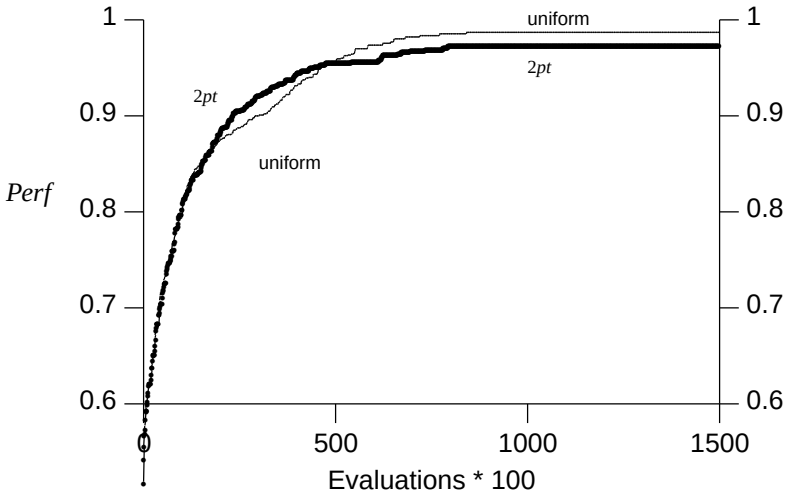

Figure 4: HC10 (45 bits) - Population 1000

It is obvious from these results that is is hard to predict (ahead of time) which crossover form is optimal for a given problem. Furthermore, it is hard to argue that any one form is optimal over the course of one run. This has motivated us to consider the possibility of adaptively choosing between crossover forms *during* a run. This paper specifically considers the restricted problem of choosing between only 2 crossover forms: 2-point crossover and uniform crossover. Before we can outline our method, however, some background on how adaptive mechanisms can be used in genetic algorithms is in order.

4 Background

As pointed out earlier, the tension between exploitation and exploration is a recurring theme in genetic algorithms. There are a number of factors that influence this tension. These factors include: population size, selection mechanisms, and operator probabilities. A large population size can enhance exploration. A strong selection algorithm can stress exploitation. Finally, as indicated in previous sections, the choice of crossover operator also influences the tension between exploitation and exploration. Although all these factors are of interest, this particular paper is concerned mainly with the selection (and adaptation) of crossover operators.

There have been a number of attempts to couple adaptive mechanisms with genetic algorithms. It is convenient to divide the mechanisms into two categories, which we refer to as the *offline* approach and the *online* approach. The *offline* approach attempts to adapt the genetic algorithm over the course of many runs. In contrast, the *online* approach adapts the GA as it solves one problem.

A good example of the offline approach is provided by [Grefenstette, 1986]. In this paper a separate *meta*-GA is used to adapt a genetic algorithm as it solves a suite of

problems. Unfortunately, this does not satisfy our purposes since our goal is to adapt the GA *as* it solves a problem. For this reason we concentrate on online approaches.

Online approaches can be further divided into three categories, that we refer to as: *tightly* coupled, *loosely* coupled, and *uncoupled*. Any adaptive mechanism requires some information on which to base the adaptations. Frequently, this information is in the form of histories or trails of statistics (i.e., bookkeeping). The genetic algorithm is an adaptive mechanism that uses a population to maintain statistics about the space being searched. This population is used by the genetic algorithm to adaptively guide the search through the space (via genetic operators). It seems reasonable, then, to consider using the GA as an adaptive mechanism for adjusting *itself* as it solves a problem. We refer to this approach as *tightly* coupled. In an uncoupled approach, a totally separate adaptive mechanism adjusts the GA. In the *loosely* coupled approach the GA is partially used for the adaptive mechanism (i.e., either the population or the genetic operators are used in some fashion).

It should be stressed that any of these approaches involve the search of two spaces. One space is the original space being searched by the GA. The other is a space of GAs, which is being searched by the adaptive mechanism. In the tightly coupled approach, the GA is simultaneously searching both spaces. This approach is elegant and straightforward, since no new adaptive mechanism is needed (see [Schaffer, 1987] for an example of this approach). The approach we describe in this paper is tightly coupled. One possible problem with tight coupling is that the coupling of the two search spaces could hinder the adaptive mechanism. As the population evolves, diversity will be lost, in *both* spaces! It is precisely at those times that the adaptive mechanism may be most useful. If diversity has been lost, that adaptation may be difficult to accomplish. This issue will be discussed again in this paper.

Another concern is that this mechanism may only work well with large population sizes, since the population is being used to sample the space of GAs. As with any other GA (or statistical sampling) application, small populations (samples) may be misleading, and lead to wrong conclusions. This issue also will be discussed further in this paper.

An uncoupled approach does not rely upon the GA for the adaptive mechanism (see [Davis, 1989] for an example of this approach). While this may alleviate the problems associated with coupling, such mechanisms appear to be difficult to construct, and involve a lot of separate bookkeeping. The result is an inelegant system that lacks generality. One possible compromise would be to have an adaptive mechanism that uses the GA for maintaining statistics, but has a separate set of operators. This mechanism could be fairly elegant, robust, and general, yet avoid the problems associated with coupling. This is a topic for future papers.

5 Implementation

As mentioned earlier, we are interested in having the GA be self-selective with respect to its choice of crossover operator. Since 2-point crossover is the least disruptive, and uniform the most disruptive, it is reasonable to have the GA select from only those two possibilities.

Since the tightly coupled approach is the most straightforward, we have chosen that approach for our initial experiments. Recall that, with the tightly coupled approach, the GA simultaneously searches both the original problem space and a space of GAs. Since we are only interested in choosing between 2-point crossover and uniform crossover, how do we allow the GA to decide which crossover operator to perform?

One obvious solution is to append one bit to the end of every individual in the GA population. Suppose a '1' refers to uniform crossover, and a '0' to 2-point crossover. Then the last column of the population (i.e., the last bit of every individual) is used to sample the space of GAs. If it is better to use uniform crossover, more '1's will appear in the last column as the GA evolves. If it is better to use 2-point crossover, more '0's will appear. Also, since the approach is tightly coupled, all genetic operators are allowed to manipulate this extra column of bits (including crossover!).

How are these extra bits used when crossover is about to be performed on individuals? It is clear that the extra bits should somehow influence which operator is performed. There are two possible techniques, which we refer to as *local* and *global* adaptation. In *local* adaptation the last bit of each individual is important for choosing which crossover operator is performed on that individual. For example, suppose two individuals are chosen for crossover. Then we examine the last bit of each individual. If the two bits are '1's, we perform uniform crossover. If the two bits are '0's, we perform 2-point crossover. If the two bits are different, we randomly choose which crossover operator to perform.

In *global* adaptation, the last bits of the population probabilistically determine which crossover operator to perform on each individual. In the first case, the choice of crossover operator is tied to a particular individual. In the second case, the choice of crossover operator is not tied to a particular individual, but to a population as a whole. For example, suppose the last column has 75% '1's, and 25% '0's. Then, when crossover is performed on the population, uniform crossover will be called 75% of the time, and 2-point crossover will be called 25% of the time.

Note that, from a probabilistic viewpoint, local and global adaptation are roughly equivalent, since the odds of firing a particular operator are the same, for a given generation. However, they are not equivalent in the sense of coupling, since local adaptation stresses individuals, while global adaptation stresses the population as a whole. For the experiments described below, the local adaptation method is used. However, the distinction between local and global adaptation is important, and will be reexamined later in this paper.

6 Results

To test our system, we investigated a set of *n*-Peak and Hamiltonian Circuit (HC*n*) problems from [Spears, 1990]. For the sake of brevity we will only present the results from the problems presented earlier in this paper: 6-Peak and HC10. The results for all the problems were similar.

Figure 5 illustrates the success of the adaptive mechanism on the 6-Peak problem with a population size of 20. The solid line represents runs using the adaptive operator. The dotted lines represent the runs with the fixed operators (2-point and uniform). Note that although the adaptive operator performs moderately well at the beginning, it loses to both fixed operators at the end of the runs.

Figure 6 illustrates the success of the adaptive operator on the 6-Peak problem with a population size of 1000. The thin solid line represent the runs with the adaptive operator. In this case the adaptive operator performs roughly as well as the 2-point operator.

Figure 7 illustrates the success of the adaptive mechanism on the HC10 problem with a population size of 1000. The thin solid line represents the runs with the adaptive operator. Note that the adaptive operator performs as well as 2-point in the beginning of

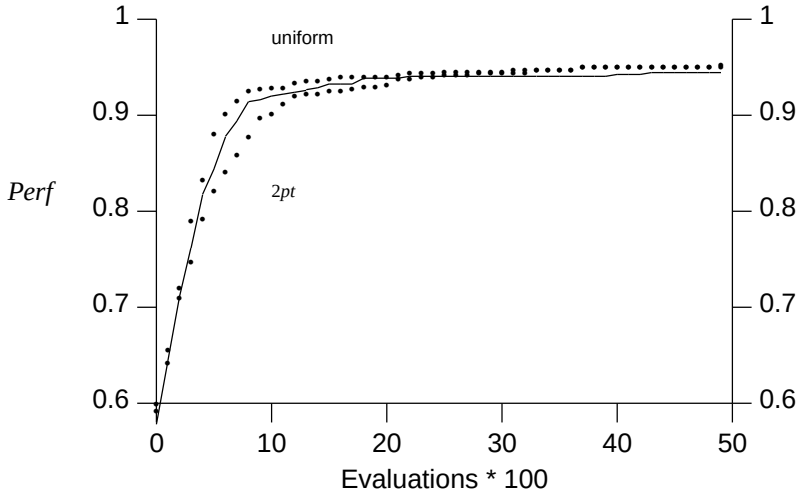

Figure 5: 6-Peak (30 bits) - Population 20

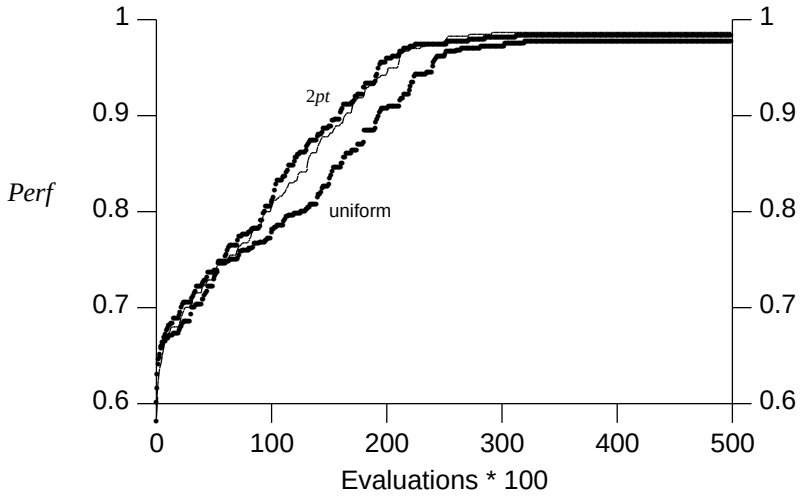

Figure 6: 6-Peak (30 bits) - Population 1000

the runs, and then performs as well as uniform crossover at the end of the runs.

Figure 8 illustrates the probability of each operator during a typical experiment (Hamiltonian Circuit) using the adaptive mechanism. The vertical axis represents the probability that uniform crossover will be used (and one minus the probability that 2-point crossover will be used). Since the population is initialized randomly, the initial

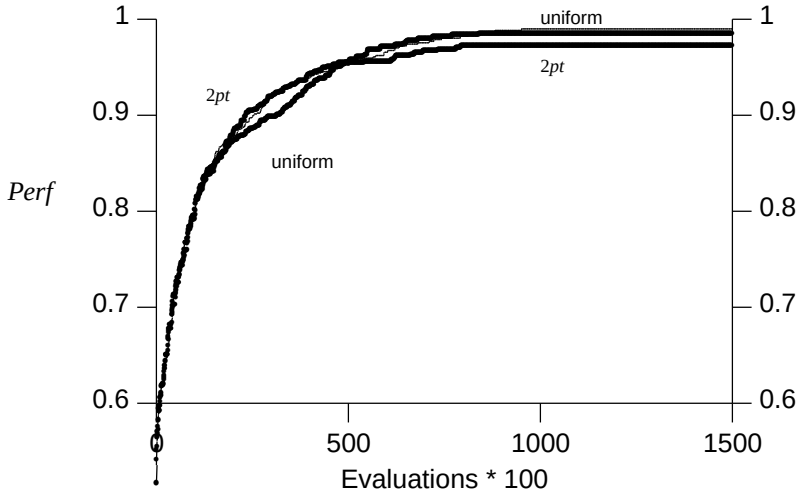

Figure 7: HC10 (45 bits) - Population 1000

probability of using uniform crossover is 50%. Note that when the population size is 20, the probability of using uniform remains at 50%, averaged over 30 runs. When individual runs are analyzed it is apparent that the probability of using uniform crossover rapidly approaches 1 or 0 during the course of the run. However, when averaged over 30 runs it is clear that no consistent adaptation is occurring.

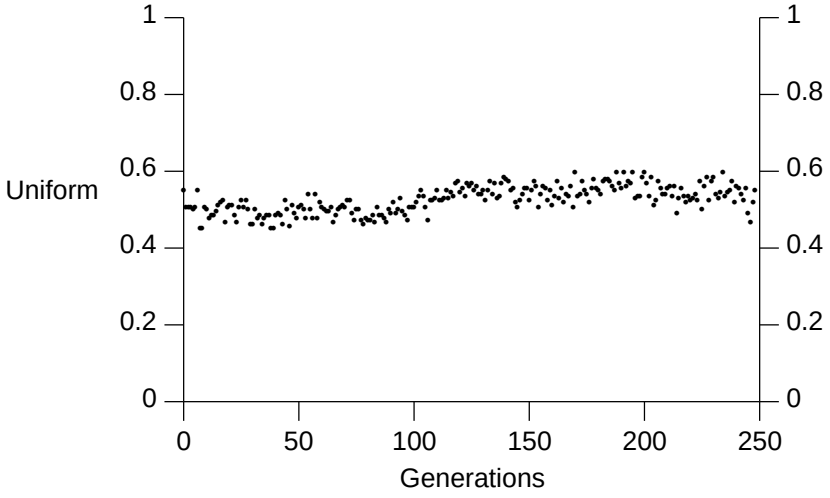

Figure 8: Operator Adaptation - Population 20

Figure 9, however, illustrates a typical experiment (Hamiltonian Circuit) of the adaptive mechanism when the population size is 1000. In this case the system gradually begins to favor 2-point crossover, with uniform crossover being used less than 20% of the time toward the end of the runs. Individual runs also confirm that this adaptation is consistently occurring.

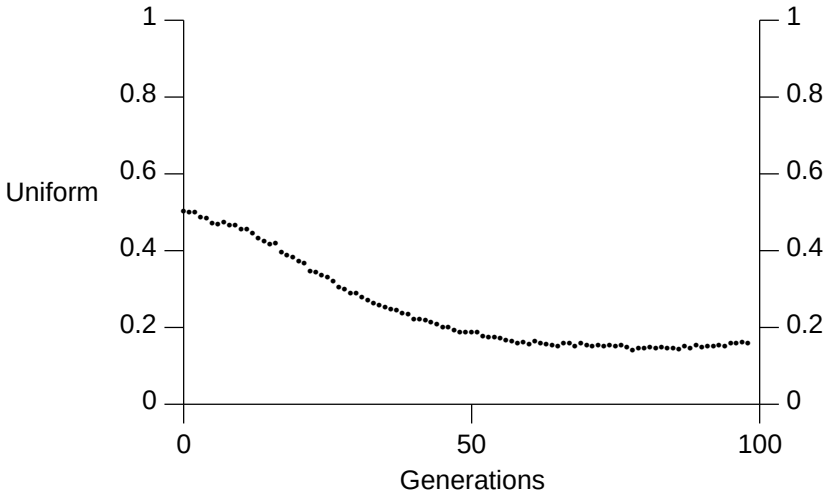

Figure 9: Operator Adaptation - Population 1000

What conclusions can be drawn from these experiments? First, Figure 8 indicates that the adaptive mechanism is not working well with small population sizes. However, this is not surprising, if we recall that small populations may result in misleading sampling. Second, Figure 9 indicates that the adaptive mechanism is working when the population is larger. This is confirmed by the experimental results, in which the adaptive operator did as well as the better fixed operator (see Figures 6 and 7).

7 Discussion

Since the adaptation does not work well with small population sizes, how do we explain the fairly good experimental results (with small population sizes)? We reran the system with the adaptation turned off (thus maintaining equal probabilities that either crossover operator will be used), and the results are similar. This indicates that simply having the two operators available to the system is useful. The non-adapting system, however, does not perform as well as the adaptive operator when the population size is larger (of size 1000). For the larger populations, then, it is apparent that the adaptation contributes to the experimental success.

As mentioned earlier, one concern is whether a tightly coupled adaptive system can adapt near the end of runs, since the population has lost diversity. Figure 9 indicates that adaptation is occurring for the first 50 generations (i.e., the operator probabilities are still changing). However, there is no evidence of adaptation towards the end of the runs. Unfortunately, absence of adaptation is not necessarily proof that the adaptation can not occur. Furthermore, there is also no evidence that adaptation was necessary for these

problems (at the end of the runs).

The mechanism outlined above uses local adaptation. Is this necessary? If one views the genetic algorithm as a source of statistics about the space being searched (i.e., the emphasis is on the population as a whole), it would appear that there should be little difference between the local and global approaches. To test this, we reran our experiments using the global adaptation mechanism outlined earlier. We found the local adaptation mechanism to consistently outperform the global mechanism. Closer inspection revealed that adaptation did not occur, in the sense that the operator probability curves looked like Figure 8, regardless of population size! This indicates that it is important to tie the adaptation directly to individuals, and also lends some support to an alternative view of the genetic algorithm as a search procedure that supports multiple trajectories throughout the search space. In this view, each trajectory may have different requirements (e.g., amount of disruption).

8 Conclusions and Future Research

In this paper, we have explored one possible adaptive mechanism for selecting optimal crossover forms for a genetic algorithm. We have shown that a tightly coupled adaptive mechanism appears to work well, especially with larger population sizes. Also, we have evidence to support the view that the adaptation should be local (i.e., tied to particular individuals).

Future work will need to consider in more detail the advantages and disadvantages of tightly coupled adaptive mechanisms, especially in comparison with more loosely coupled mechanisms. The differences between local and global adaptation must be resolved. Finally, we would like to have the GA be self-selective with respect to the other factors that influence the tension between exploration and exploitation.

Acknowledgements

We would like to thank Diana Gordon and the INFT910 class for many helpful comments.

References

Davis, Lawrence (1989). Adapting Operator Probabilities in Genetic Algorithms, *Proc. 3rd Int'l Conference on Genetic Algorithms*, Morgan Kaufman Publishing.

De Jong, K. A. & Spears, W. (1990). An Analysis of of the Interacting Roles of Population Size and Crossover in Genetic Algorithms, *Proceedings of the First Int'l Conf. on Parallel Problem Solving from Nature*, Dortmund, Germany, October 1990.

De Jong, Kenneth A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Doctoral thesis, Dept. Computer and Communication Sciences, University of Michigan, Ann Arbor.

Eschelman, L., Caruana, R. & Schaffer, D. (1989). Biases in the Crossover Landscape, *Proc. 3rd Int'l Conference on Genetic Algorithms*, Morgan Kaufman Publishing.

Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization & Machine*

Learning, Addison-Wesley Publishing Company, Inc.

Grefenstette, John J. (1986). Optimization of Control Parameters for Genetic Algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-16, No. 1, Jan/Feb 1986.

Holland, John H. (1975). *Adaptation in Natural and Artificial Systems*, The University of Michigan Press.

Schaffer, J. David & Amy Morishima (1987). An Adaptive Crossover Distribution Mechanism for Genetic Algorithms, *Proc. 2nd Int'l Conference on Genetic Algorithms*, Morgan Kaufman Publishing.

Spears, William M. & De Jong, K. A. (1990). An Analysis of Multi-point Crossover, *Proceedings of the Foundations of Genetic Algorithms Workshop*, Indiana, July 1990.

Spears, William M. (1990). *Using Neural Networks and Genetic Algorithms as Heuristics for NP-Complete Problems*, Masters Thesis, Department of Computer Science, George Mason University, Fairfax, Virginia, 1990.

Syswerda, Gilbert. (1989). Uniform Crossover in Genetic Algorithms, *Proc. 3rd Int'l Conference on Genetic Algorithms*, Morgan Kaufman Publishing.