On Crossover as an Evolutionary Viable Strategy

J. David Schaffer
Binghamton University
83 PUBLICATIONS 7,339 CITATIONS
SEE PROFILE

Larry J. Eshelman
Philips
22 PUBLICATIONS 2,761 CITATIONS
SEE PROFILE

ICGA -91
MORGAN KAUFMA
P 61-68

On Crossover as an Evolutionarily Viable Strategy

J. David Schaffer
Philips Labs
Briarcliff Manor, NY, USA
ds1@philabs.philips.com

Larry J. Eshelman
Philips Labs
Briarcliff Manor, NY, USA
lje@philabs.philips.com

Abstract

Genetic algorithm theory has long maintained that fitness proportional reproduction and crossover recombination yield a powerful search algorithm. The analyses behind this assertion has implied that highly disruptive crossover is likely to be inferior to a less disruptive one. Recent empirical results have provided two opposite challenges to these conclusions: one suggesting highly disruptive crossover can be more effective than less disruptive crossover, and one suggesting that crossover itself contributes nothing that mutation alone cannot provide.

To shed some light on these questions we present results from experiments in which a small subpopulation using crossover tries to invade a larger population that does not. We observe that crossover always takes over except when it is very disruptive. However, this takeover is not always to the benefit of the population; for some environments, mutation alone can eventually do better. We also find support for the assertion that crossover can exploit epistasis that mutation alone cannot and observe much faster search by crossover on tasks that mutation alone can solve.

1 Introduction

A popular model of evolutionary processes involves a repeated cycle of fitness biased selection and random mutation applied to a population of structures. Holland[10] has argued for the benefits of crossing over material from two parents. The claim is that there are non-linear interactions (epistasis) among the building blocks (genes, schemata) within these structures that mutation, proceeding as it does by single steps, could not properly exploit. Procedures employing these notions have become known as genetic algorithms (GAs). Nevertheless, to counter the premature convergence possibilities of crossover alone, GA prescription has been to use a low level of mutation in addition to crossover, but it has been believed that this was not the main driver of the search. Schaffer et al.[12] have suggested that contrary to this thinking, mutation was actually making a significant contribution to the search effectiveness, an observation supported by the higher mutation rates found by a meta-GA for a GA[8]. In addition, Davidor[2] has suggested that there are degrees of epistasis that are beyond even crossover's exploitive abilities. Fogel and Atmar[6] have gone so far as to argue that crossover cannot claim any generally useful advantage over mutation and selection alone, supporting their claim with experiments in which subpopulations using different combinations of genetic operators competed for dominance. Their results show mutation alone dominating the subpopulations using crossover.

Biological observations suggest that a very low level of mutation and crossover with low disruptiveness are what Nature uses and this is consistent with traditional GA theory. If one favors the Panglossian assumption — if Nature does it, it must be the best thing to do — then one must try to explain these new empirical observations.

To shed more light on these issues, we wish to address the question, "is crossover an evolutionarily viable strategy?" That is, suppose there is an evolutionary system in operation (i.e. a periodic reproduction and selection of the fittest cycle operating on a population of individuals), and suppose that a gene appears for recombination (crossover). Under what circumstances, if any, will this gene come to dominate in the population? And is the effect of crossover beneficial to the search process?

2 The Race

There will clearly be a race between the offspring of the crossover subpopulation and the rest of the population. Even in the case where the rest of the population has no method for introducing variation, selection will still be putting pressure on these offspring by increasing the instances of the best unchanging individuals, thus lowering everyone else's relative fitness. Once crossover produces an individual superior to the best unchanging individual, the takeover of the crossover gene is assured, so long as crossover is unlikely to quickly destroy him again. On the other hand, if crossover is highly disruptive of what has contributed to the survival of its possessors, they may not get enough tries to produce the wunderkind before their subpopulation dies out.

When mutation is also operating, the race is between the (possibly mutated) offspring of crossover, the offspring of mutation alone, and the unchanged individuals. All the while, selection will be tending to reduce the gene pool variation, mutation will be tending to maintain it, and crossover will be tending to redistribute it.

A critical factor in determining the outcome of this race seems to be the *riskiness* of the change operator (mutation or crossover). We define a *safety ratio* as the ratio of the probability that an offspring will be better than the parent(s) to the probability that it will be worse. These probabilities will be influenced by several factors, but unfortunately we know of no theory that identifies what they are. As a start toward building such a theory, we have chosen to examine empirically the epistasis in the problem[1], the disruptiveness of the operators to the schemata in the parents, and the rate of mutation.

3 Experimental Methods

Simulations of the invasion of a population by a small subpopulation of individuals possessing a crossover gene were conducted under varying conditions of problem epistasis, crossover disruptiveness and level of background mutation.

Following majority practice in GA work, the structures manipulated by our genetic search procedures were bit strings (chromosomes). All chromosomes contained 100 bits used to compute fitness and one more bit for a crossover gene. The simulations began with a random initial population of 500. This population size is larger than usually used for optimization, but was chosen to reduce the variance in the outcomes from stochastic effects (genetic drift). Furthermore, we are not interested here in discovering good optimization strategies per se, where minimizing the total computational effort to an answer of a given quality is the issue. We are assuming a Nature-like model in which all population members are evaluated in parallel and the race is conducted on a generation basis.

After evaluation of the initial population, the fittest members were selected using Baker's unbiased selection procedure[9]. A randomly selected 10% of this population had their crossover gene turned on and the others had theirs turned off. Thus, the invading subpopulation begins with 50 individuals, a size often used for genetic optimization tasks. A smaller subpopulation may not contain enough initial variation to be viable. No further direct manipulations of the crossover genes were done. Then followed generation cycles consisting of the following steps:

> Randomly mate the crossing individuals among themselves[2], and replace them by their offspring.
>
> If (mutation rate > 0) perform mutation on the entire gene pool (including the crossover gene).
>
> Evaluate the new offspring.
>
> Select the fittest.
>
> Terminate if (population converged, 2 generations with no new offspring (spinning) or 50,000 trials).

Twenty independent runs were executed with different random seeds, using each combination of the following experimental conditions.

Epistasis: We chose five problems with known amounts and types of epistasis, Onemax, Plateau, Plateau-d, Trap and Trap-d.

Onemax is a simple linear problem (i.e. zero epistasis). The evaluation returned for each 100-bit string is just the number of ones it contains. The optimum (minimum) value is zero and occurs for the all-zero string, the worst possible score is 100 and the expected score for a random 100-bit string is 50. The Plateau problem contains the maximum amount of epistasis within small segments (genes), and no epistasis between segments. For each five-bit segment, a value of five is given for the "00000" pattern (the optimum allele) and zero for all others.

The score for a string is 100 minus the sum of the segment scores. Thus the optimum score is zero and the worst possible score is 100. The expected score for a random 100-bit string is 100 – 20(5)(1/32) = 96.875.

[1] The problem is taken to be the composite of the representation, usually a mapping of the instance space to be searched onto strings of some sort, and the evaluation function that is computed from these strings. It has been common to think of the problem as only the evaluations function, but recently more research attention has begun to be focused on the mapping as well[11,14,15].

[2] If the number of crossing individuals is odd, then one mating with a non-crossing individual will occur with probability 0.5.

The Trap problem is like the Plateau problem, but is deceptive[3]. A value of five is given for the "00000" allele, but for other patterns, a value of 0.5 times the number of ones is given. Unless the optimum allele is present, there is pressure away from it towards its complement. The optimum score is again zero, the worst possible score is 90 (for a string where all 20 segments have a single one and four zeros), and the expected score for a random string is

$$100 - 20\left[\frac{5}{32} + \binom{5}{1}\frac{0.5}{32} + \binom{5}{2}\frac{1.0}{32} + \binom{5}{3}\frac{1.5}{32} + \binom{5}{4}\frac{2.0}{32} + \frac{2.5}{32}\right]$$

$$= 71.875$$

Note that the epistasis in the Plateau and Trap problems is very local; it includes only the five adjacent bits of the segment. This situation favors the bias of the two-point crossover operator, but as Eshelman et al. [4] point out, this locality cannot generally be assumed. So, we included a variant of each of these problems, Plateau-d and Trap-d, in which the segments were uniformly intermixed along the chromosome. That is, the bits of segment 1 were at loci 1,21,41,61, and 81, segment 2 at 2,22,42,62 and 82, etc.

Disruptiveness of crossovers: We studied two crossover operators that differ on this dimension. Two-point crossover (2X) selects at random two loci and exchanges the parental segments between them. This operator shares the local bias of traditional one-point crossover, but is less disruptive and exhibits generally superior search behavior[3,12]. Uniform crossover (UX), simply exchanges the parental bit at each locus with an even probability. It is considerably more disruptive than 2X of short schemata, but does not share its local bias which makes it less exploitive of spurious correlations[13]. It is also a more vigorous recombiner of parental schemata[5].

Mutation rate: We chose two rates, zero (no mutation), and 0.0005. The latter rate was selected after some preliminary experiments, and represents an attempt to balance the expected number of improvers from the mutation and crossover subpopulations in the initial generation of the race. This balance could not be perfect for all of the problems; however, rates much higher than this lead to very poor search by mutation, and rates much lower lead to intolerably slow search.

4 Baseline Tests

Anticipating that insight into the riskiness of each operator would be useful for interpreting the race results, some baseline experiments were conducted. Each problem was run 20 times using selection with each change operator alone (i.e. mutation @ .0005, 100% 2X, or 100% UX). The population size was 500 in all cases. During these runs all offspring produced were compared to their parents and statistics were kept on the fraction of them that were better and worse than their parents (both parents in the case of crossover). The ratio of P(better)/P(worse) can be thought of as a *safety ratio* for the operator. The higher this ratio, the safer it is for a population to use the operator to produce offspring.

4.1 Mutation alone

The safety ratio curves for mutation are displayed in Figure 1. The Plateau-d and Trap-d were not tested for mutation or for UX since these operators have no local bias making these problems the same as their locally coded counterparts. These curves all reflect the increasing risk of using mutation as the population becomes more attuned to the problem (environment). On Onemax and Trap the ratio begins quite high, indicating that the likelihood of improving by random mutation is almost as high as the likelihood of worsening. Improvements are easy in the nearly random populations. The population on Trap converges more slowly than on Onemax because the selection pressure is less, particularly after the first 100 generations or so. Mutating an optimal allele yields a worse offspring, but mutating a non-optimal allele can still make small improvements. The Plateau problem represents a much riskier environment. After the very first selection step the population will consist only of individuals with one or more optimal alleles (those with none score zero and produce no offspring). Mutation is unlikely to add a new one, but is likely to destroy one. After 50 generations or so, mutation is very dangerous.

The view of the riskiness of using mutation from these curves is strictly local (i.e. offspring compared to parents). Showing how the population performs relative to the true global optimum, Table 1 presents the mean and variance (among the 20 runs) of the best individual found and the mean number of generations needed to find it. Note that in spite of the increasing risk associated with mutation, this is a fairly effective search algorithm. The global optimum is found every time for Onemax and most of the time (16/20) for Plateau and in a number of generations well beyond the point when the risk of destroying the parent's genetic material is very high. Even on Trap, the final solutions contain

[3] Deceptiveness is a concept introduced by Goldberg[7] and refers to a situation in which the average fitnesses of low order schemata lead away from the optimal solution.

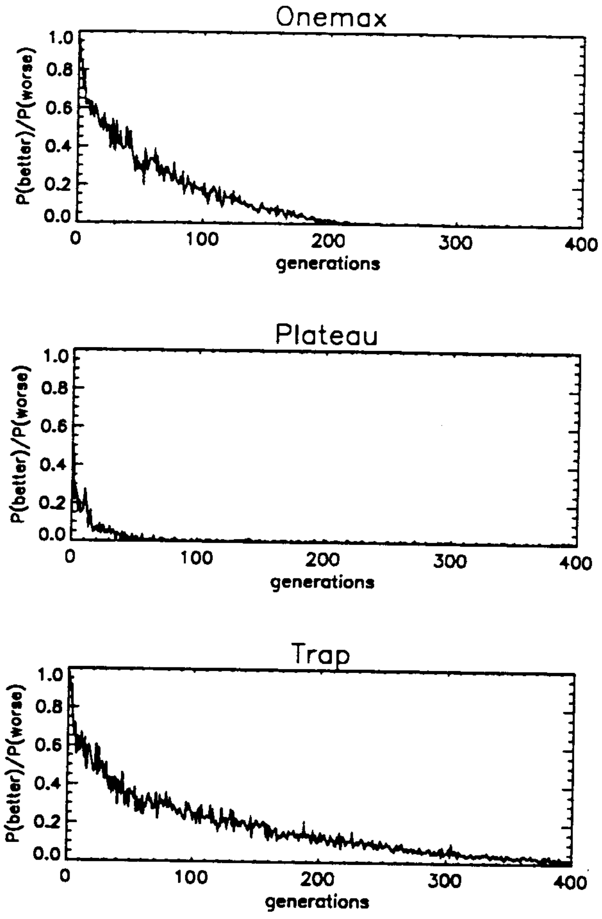

Figure 1. The *safety ratio* of using mutation alone

several optimal alleles (about 7/20) and the penoptimal alleles (the false peak) for the rest of the segments. This algorithm is somewhat analogous to simulated annealing or stochastic hillclimbing. In the early generations, improvements are easy and the survival threshold is low. Even offspring worse than their parent can survive so long as they're better than the worst in the population. Later, after much selection, only improvers are likely to survive (strictly descending).

Table 1: Performance of mutation alone

prob	mean best	sd best	mean generations @ best
Onemax	0	0	298
Plateau	0.015	0.033	1732
Trap	0.334	0.049	450

The good performance on the non-epistatic Onemax is no surprise, and climbing the false peaks of Trap is just a small version of Onemax, but Plateau seems to contradict the assertion of Holland[10, p. 110] that a reproductive plan using mutation alone is little better than an enumerative process. Plateau is highly epistatic, although the epistasis is low order (only 5 bits). There is a mechanism operating here that is not immediately obvious in Figure 1. Even after P(better)/P(worse) has dropped near zero, there is still a small chance of producing an improved offspring that is just perceptibly larger than for Onemax after it becomes essentially zero. To achieve this, mutation must be creating optimal alleles. For this to happen at the rates just visible in the figure, there must be a mechanism that favors 0s over 1s in the gene pool. This mechanism is the breakup of optimal alleles which selection has already concentrated, thus increasing the likelihood that they may be reconstructed by mutation. This experiment illustrates that, as Fogel and Atmar[6] observe, reproductive plans using only mutation can be very effective on surprisingly difficult problems.

4.2 2X alone

Risk curves for 2X are given in Figure 2. A curve for Onemax is not included, because its shape is obvious; by the nature of Onemax whenever an offspring is produced that is better than both parents, its sibling will be worse. Hence the P(better)/P(worse) ratio is forever 1.0. Note that these searches terminate much sooner than do those with mutation. As Holland has predicted, crossover is less effective than mutation at maintaining gene pool variation in the face of selection pressure to converge. On Plateau, the population after initial selection contains only individuals with at least one optimal allele and few with more (individuals with none must be the worst in the population, and do not reproduce). Hence disruption by 2X is not too likely relative to recombination, so the ratio is quite high. Very soon selection begins increasing the number of individuals with more than one optimal allele and the likelihood of disruption goes up so the ratio drops. Then convergence begins to make disruption less likely and creation by recombination more likely (0's are proliferating). The ratio climbs. A similar story is evident for Plateau-d, but now initial disruption is much more likely, so the ratio starts much lower. An important point here is that there are phases of the search when the risk of engaging in crossover may be high or low.

On Trap, the situation begins like Plateau, but here there is a force tending to proliferate ones instead of zeros. Even when convergence makes 2X less disruptive, it is also less creative, because the gene pool is not so rich in the building blocks of the optimal alleles. Unlike Plateau, there is no assurance that all the individuals in the initial population will have at least one optimal allele (not all those with none will be the population's worst individuals). If crossover does not disrupt any optimal alleles (either because there are none or because they're missed by crossover) the result is exactly like Onemax, one offspring is better than both parents and one is worse giving an instance of the ratio=1.0 that's added into the ratio sum. This occurs less often in Plateau where such events yield offspring that are equal to their parents unless there is recombination of optimal alleles. This accounts for the ratio initially being higher for Trap than Plateau and for Trap-d than Plateau-d. The curves for both Trap and Trap-d initially fall because of disruption of optimal alleles, but this disruption is more vigorous in Trap-d soon yielding a population in which there are few optimal alleles. Then the problem becomes a Onemax as the false peaks are climbed. The ratio looks like things are improving, but performance is poor. (See below.) On Trap, more optimal alleles are preserved than on Trap-d, so crossover remains risky.

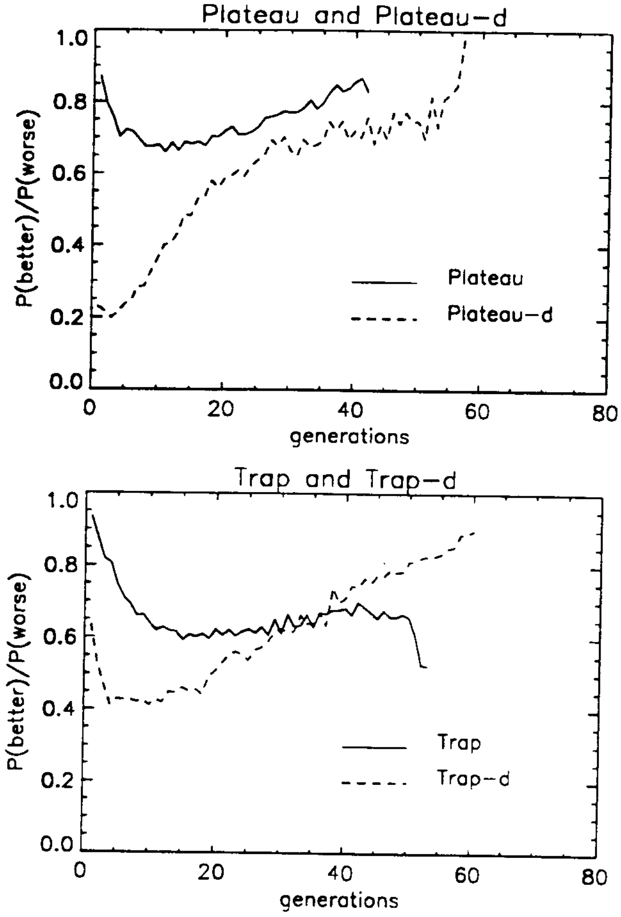

Figure 2. The *safety ratio* of using 2X alone

The performance is shown in Table 2. We see that 2X alone yields a very effective search algorithm when its local bias is beneficial. This algorithm finds the global optimum every time for Onemax and Plateau and most of the time (18/20) for Trap. Furthermore it does so more quickly than mutation alone. The poor performance on Trap-d is expected, but good performance on Plateau-d is a bit surprising. While 2X is highly disruptive of the distributed optimal alleles, the selection preference for them will yield a gene pool that is littered with their fragments even though there is no explicit selection preference for the fragments themselves. We observed above that mutation could exploit this situation, and here we observe that crossover, by recombining these fragments (building blocks), is even more effective (58 generations versus 1732).

Table 2: Performance of 2X alone

prob	mean best	sd best	mean generations @ best
Onemax	0	0	35
Plateau	0	0	30
Plateau-d	0.0025	0.011	58
Trap	0.003	0.008	40
Trap-d	0.434	0.049	59

4.3 UX alone

Risk curves for UX are shown in Figure 3. Again the Onemax curve is trivial and so not shown and the Plateau-d and Trap-d curves are the same as their non-distributed counterparts since UX has no local bias. These curves reflect the same mechanism discussed for 2X, but show the increased disruptiveness of UX.

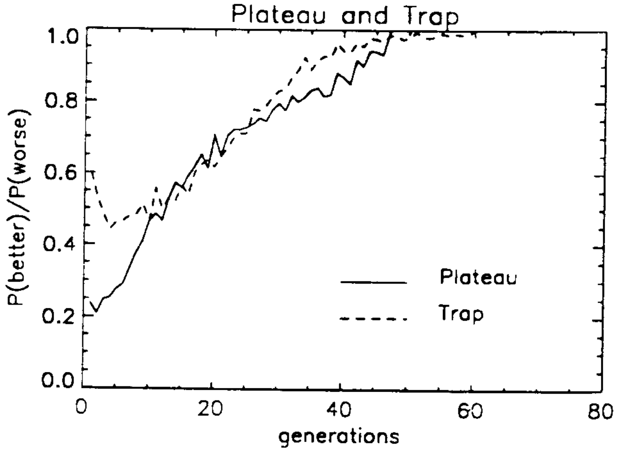

Figure 3. The *safety ratio* of using UX alone

The performance is shown in Table 3. This algorithm's performance is essentially the same as 2X on Onemax. On Plateau, its disruptiveness causes some trouble leading to slower search, but it's still able to find the global optimum. A longer epistatic segment would have caused more trouble. On Trap, UX is effectively trapped. It destroys all the optimal alleles and then climbs the false peaks as though this were a Onemax problem.

Table 3: Performance of UX alone

prob	mean best	sd best	mean generations @ best
Onemax	0	0	35
Plateau	0	0	37
Trap	0.491	0.013	41

5 Race Results

The fates of the crossover gene under the various race conditions are shown in Table 4. For each race, Table 4 presents the mean over 20 runs of the percent of the population with the crossover gene at its maximum point. This occurred at termination (convergence) for the runs without mutation, but with mutation the populations never converged and so ran to the specified computing limit (50,000 offspring). Once the best value ever found had essentially taken over the population, there was no longer any differential survival value to having the crossover gene, and its percent of the population tended to drift toward 50 under the action of mutation. Also shown are the mean best value and the average generation at which this value first appeared.

We can group the race outcomes into four classes.

Class 1: The crossover gene takes over and the population achieves a level of performance that mutation alone could not reach. Two cells in Table 4 fit this class: 2X (with and without mutation) on the Trap problem. This is the type of environment in which GA theory predicts success for crossover: there is significant but local epistasis. Not only is the performance better than mutation alone, this performance is achieved very rapidly. It is interesting to note that the performance of the population, once invaded by 2X, is worse than it was when 100% 2X was used from the beginning (see Table 2). When the subpopulation engaging in crossover is small, the gene pool variance can be lost too rapidly to sustain effective search. The effect of mutation may be to delay this convergence (56 vs. 62 generations to find its best) and to improve the performance slightly (.13 vs. .09). These differences are not statistically significant.

Class 2: The crossover gene takes over and the population performance is about the same as with mutation alone, but this performance level was reached much sooner. This class includes Onemax under all conditions, 2X (with or without mutation) on Plateau, 2X

Table 4: Race results (with mutation alone for comparison)

problem	MU		2X						UX					
			without MU			with MU			without MU			with MU		
	mean best	gen @ best	max % xo	mean best	gen @ best	max % xo	mean best	gen @ best	max % xo	mean best	gen @ best	max % xo	mean best	gen @ best
Onemax	0	298	100	.05	45	99	0	46	100	0	35	99	0	36
Plateau	.015	1732	100	.04	55	99	0	64	65	.49	26	99	.18	240
Plateau-d	*		40	.68	14	99	.16	482	*			*		
Trap	.33	450	100	.13	56	99	.09	62	5	.57	4	99	.33	173
Trap-d	*		6	.56	4	99	.34	191	*			*		

* Experiment not run. The expected results are known to be the same as the table entry immediately above.

with mutation on Trap-d, and UX with mutation on Trap and Trap-d. The Onemax results are no surprise, crossover provides more effective recombination than mutation alone, but the problem class is easy and could have been solved even more rapidly by hillclimbing. The surprise on Plateau is not the success of 2X (this was expected for the same reasons as Trap), but the success of mutation alone. With epistasis limited to five bits, mutation can accumulate beneficial changes even when they don't convey any immediate survival benefit. Another unexpected result is that the highly disruptive crossovers (2X on Trap-d and UX on Trap and Trap-d) could do so well when coupled with mutation. They did better with a small invading population than they did when they were used at 100% from the beginning (see Tables 2 and 3). When we examine these searches we note that often the invading subpopulation actually dies out, but is resurrected again by the action of mutation on the crossover gene. When this resurrection happens with the population in a more favorable state because of convergence (the *safety ratio* curves in Figures 2 and 3 rise), then the crossover gene can take over.

Class 3: The crossover gene takes over, but the population performance is worse than it was with mutation alone. This class includes 2X with mutation on Plateau-d and UX with mutation on Plateau and Plateau-d. Unlike the previous class, here there is a destructive interaction between crossover and mutation. Crossover is so disruptive that the beneficial mutations cannot survive long enough to accumulate.

Class 4: The crossover gene fails to take over consistently and performance is worse than with mutation alone. This class includes 2X on Plateau-d and Trap-d and UX on Plateau, Plateau-d, Trap and Trap-d, all without mutation. These are all circumstances where crossover is highly disruptive and there's no mutation to help it out. On the Trap problem the demise of the crossover gene is quite dramatic with the best individual ever found often appearing in the initial population. Convergence is rapid after the loss of any mechanism to combat it. On the Plateau problem, however, sometimes crossover survives (40% of the time for 2X and 65% for UX). The population performance is better when recombination survives (mean best scores of .55 in the 8 runs where 2X took over versus .77 for the 12 were it died and .35 in the 13 runs where UX took over versus .76 when it died), but still not as good as when the crossover was used 100% from the start (see Tables 2 and 3). As noted with class 1 above, the small population does not provide sufficient defense against selection's pressure for convergence.

6 Conclusions

A subpopulation producing offspring by recombination using crossover was unable to invade an otherwise static population undergoing fitness proportional reproduction when crossover was excessively disruptive. This was observed for the highly disruptive UX, but also for the less disruptive 2X when the critical schemata had long defining lengths. Under these circumstances, crossover is not an evolutionarily viable strategy. However, with mutation simultaneously operating at a small rate, the invasion was always successful. This takeover resulted in the production of high performance offspring much more quickly than was the case when mutation alone was operating. Furthermore, a limited amount of deception in the problem effectively trapped the mutation-only population, but the low disruptiveness of 2X allowed its population to produce superior offspring. We observed that this takeover was not always immediate; it was sensitive to the relative riskiness of the crossover and mutation operators. As selection pressure led to more converged and more fit populations, mutation tended to become more risky while crossover became less risky. In some cases, we observed the early demise of the invading group only to see them retry later by having mutation turn the crossover gene back on. If conditions were sufficiently favorable, the new invaders were quickly successful. Under these circumstances, crossover is an evolutionarily viable strategy. However, the takeover by crossover was not always associated with superior performance. Highly disruptive crossover on mildly epistatic nondeceptive problems yielded population performance that was worse than it would have been if mutation alone had been at work.

The introduction of crossover into a population slowly evolving under selection and mutation can be a mixed blessing. It can increase the speed of evolution many fold as well as overcome some epistasis that mutation alone could not. It can also disrupt schemata that mutation could have exploited leading to poorer performance for the population.

In these experiments, UX seemed generally to perform worse than did 2X and we would be remiss if we left this conclusion to be drawn. Because of its highly disruptive nature, UX does fair poorly under the selection scheme used in these experiments wherein the offspring replace the parents and the only way to preserve a parental structure into the next generation is to employ a reproduction without modification. UX can perform impressively with a more conservative selection scheme that allows parental structures to live for many generations, only being replaced when offspring (their own or others') appear who are superior[5,13]. Under these circumstances UX can cope successfully with problems that defy 2X because of non-local epistasis.

When Fogel and Atmar ran races similar to ours, they observed mutation winning and crossover losing. We speculate there are several reasons: 1) In their experiments, crossover was always linked with an inversion operator that was not true inversion as described by Holland[10]. Their inversion operator swapped alleles between genes instead of just reordering the genes. 2) Their representation coded the real parameter values and did not use a binary representation. Thus crossover was limited to recombining the alleles present and could do no within-parameter search.

We believe Fogel and Atmar are correct to point out the power of selection and mutation for search. This power is often underestimated in the GA community. We believe it is not correct to infer from their results that crossover provides no added value. We believe our experimental results begin to shed more light on the circumstances under which crossover will deliver the search power first described by Holland. However, these results also suggest circumstances in which crossover has a detrimental impact on search. It must be used with care.

References

[1] R. A. Caruana, J. D. Schaffer and L. J. Eshelman. Using Multiple Representations to Improve Inductive Bias: Gray and Binary Coding for Genetic Algorithms. *Proceedings of the 6th International Conference on Machine Learning*, 375-378. Morgan Kaufmann, San Mateo, CA, 1989.

[2] Y. Davidor. Epistasis Variance — Suitability of a Representation to Genetic Algorithms. In G. J. E. Rawlins (editor), *Foundations of Genetic Algorithms and Classifier Systems*, Morgan Kaufmann, San Mateo, CA, 1991.

[3] K. A. De Jong. Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. Dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, 1975.

[4] L. J. Eshelman , R. A. Caruana and J. D. Schaffer. Biases in the Crossover Landscape. *Proceedings of the Third International Conference on Genetic Algorithms*, 10-19. Morgan Kaufmann, San Mateo, CA, 1989.

[5] L. J. Eshelman. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In G. J. E. Rawlins (editor), *Foundations of Genetic Algorithms and Classifier Systems*, Morgan Kaufmann, San Mateo, CA, 1991.

[6] D. B. Fogel and J. W. Atmar. Comparing Genetic Operators with Gaussian Mutations in Simulated Evolutionary Processes Using Linear Systems. *Biological Cybernetics, 63* (1990), 111-114.

[7] D. E. Goldberg. Simple Genetic Algorithms and the Minimal, Deceptive Problem. In L. Davis (editor), *Genetic Algorithms and Simulated Annealing*, 74-88. Morgan Kaufmann, San Mateo, CA, 1987.

[8] J. J. Grefenstette. Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man & Cybernetics, SMC-16*, 1, (January-February 1986), 122-128.

[9] J. J. Grefenstette and J. E. Baker. How Genetic Algorithms Work: A Critical Look at Implicit Parallelism. *Proceedings of the Third International Conference on Genetic Algorithms*, 20-27. Morgan Kaufmann, San Mateo, CA, 1989.

[10] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[11] G. E. Liepins and M. D. Vose. Representational Issues in Genetic Optimization. *Journal of Experimental and Theoretical AI*, May 1991.

[12] J. D. Schaffer, R. A. Caruana, L. J. Eshelman and R. Das. A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, 51-60. Morgan Kaufmann, San Mateo, CA, 1989.

[13] J. D. Schaffer, L. J. Eshelman and D. Offutt. Spurious Correlations and Premature Convergence in Genetic Algorithms. In G. J. E. Rawlins (editor), *Foundations of Genetic Algorithms and Classifier Systems*, Morgan Kaufmann, San Mateo, CA, 1991 forthcoming.

[14] N. N. Schraudolph and R. K. Belew. Dynamic Parameter Encoding for Genetic Algorithms. Technical Report CNLS, MS-B258, Los Alamos National Laboratory, Los Alamos, NM.

[15] C. G. Shaefer. The ARGOT Strategy: Adaptive Representation Genetic Optimizer Technique. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 50-58. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.