Recombination Without Respect: Schema Combination and Disruption in Genetic Algorithm Crossover

Richard A. Watson **Jordan B. Pollack**

Dynamical and Evolutionary Machine Organization,
Volen Center for Complex Systems, MS018, Brandeis University,
Waltham, MA 02454, USA
richardw@cs.brandeis.edu

Abstract

One-point (or n-point) crossover has the property that schemata exhibited by both parents are 'respected'—transferred to the offspring without disruption. In addition, new schemata may, potentially, be created by combination of the genes on which the parents differ. Some argue that the preservation of similarity is the important aspect of crossover, and that the combination of differences (key to the building-block hypothesis) is unlikely to be valuable. In this paper, we discuss the operation of recombination on a hierarchical building-block problem. Uniform crossover, which preserves similarity, fails on this problem. Whereas, one-point crossover, that both preserves similarity and combines differences, succeeds. In fact, a somewhat perverse recombination operator, that combines differences but destroys schemata that are common to both parents, also succeeds. Thus, in this problem, combination of schemata from dissimilar parents is required, and preserving similarity is not required. The test problem represents an extreme case, but it serves to illustrate the different aspects of recombination that are available in regular operators such as one-point crossover.

1 INTRODUCTION

One feature common to one-point crossover [Holland 1975], *n*-point crossover, and uniform crossover [Syswerda 1989] is that all these operators preserve the similarity exhibited by the parents. That is, for any locus where the parents' genes have the same allele, the child will also take that allele. The operators are distinguished by how they handle the genes that *disagree* (Figure 1). Radcliffe [1991] calls the characteristic of preserving similarity "respect" and argues that respect is a necessary starting point for recombination operators. Syswerda [1989] argues that this feature of recombination is sufficient for successful crossover, and even suggests that the other characteristics of one or n-point crossover are undesirable. Chen [1999] amplifies this point of view that preserving similarity is more important than any other feature of recombination. He hypothesizes that the benefit of recombination comes from the fact that it exploits the property that "schemata common to above average solutions are above average". Chen & Smith [1996] suggest that "preservation of common schemata is the central source of power in recombination operators".

a	00011	101011
b	10101	100110
one-point	a0aa1	10bb1b
uniform	?0??1	10??1?

Figure 1: One-point, and uniform crossover are distinguished by how they handle alleles that disagree. In one-point, disagreements in gene values that occur to the left of the crossover point are resolved in favor of parent *a*, and those to the right are resolved in favor of parent *b*. In uniform crossover, loci with disagreements may take a gene from either parent at random.

If we are to believe that the important aspect of recombination is that it preserves the common parts of the parents then it makes little sense to combine parents that are too dissimilar. This concurs with the idea that parents selected from two different fitness peaks are likely to produce an offspring that lands in the valley between. This point of view is widely supported and typified by the proposal of niching and speciation methods such as [Deb & Goldberg 1989] where mating is restricted to individuals that are genotypically similar. It is also one of the motives behind spatially distributed GAs [Starkweather et al 1991] and multi-deme GAs [Goldberg et al 1996] that promote breeding within local populations whilst 'long-distance' breeding is less likely.

So, it seems we should expect recombination to work well when parents are similar, and an important feature of any crossover operator is that it preserves similarity. But, what about the building-block hypothesis [Holland 1975, Goldberg 1989]? Is it not also GA lore that recombination works when it is able to take the good parts (building-blocks) from two different parents and put them together? The idea behind recombination as it was originally conceived [Holland 1975] is to take sub-parts from individuals that supply *different* sub-solutions and combine them. (This explains why 'long-distance' mating, as we called it, is allowed at all in the distributed methods).

In this paper, we support the basic intuition behind the building-block hypothesis: the GA performs well when it is able to combine low-order schemata of above average fitness to find higher-order schemata of higher-fitness. More precisely, we should say that this kind of 'combination' *can* be valuable on some class of problems. To investigate this, we will separate the combination feature of crossover from the similarity preserving feature of crossover.

a) Similarity and combination: one-point crossover

Ordinary one-point crossover both preserves similarity and, potentially, combines differences[1]. A schema is heritable if the parents agree on the gene values at each loci of the schema, and/or if the schema does not span across a crossover point. New schemata may be created by combination.

b) Similarity without combination: respectful/uniform

Radcliffe [1991] supplies a crossover operator that explicitly preserves similarity but does not permit combination - 'Random Respectful Recombination', R^3. This operator assigns a randomly selected allele to any loci where the parents genes are not in agreement. R^3 is equivalent to Syswerda's uniform crossover when using binary encoding because selecting genes that disagree from either parent with equal probability is equivalent to random assignment of bits at these loci. In uniform crossover, or R^3, a schema is not heritable unless the parents agree on the gene values at each loci of the schema. New schemata may be created only by the 'macro-mutations' [Jones 1995] induced by the conflicting genes.

c) Combination without similarity: disrespectful

An operator that supplies the converse is not to be found in the literature. What kind of crossover does not preserve similarity? Purely for the purposes of illustrating our point, we introduce a new recombination operator - "disrespectful crossover". Figure 2 shows that disrespectful crossover exhibits the quite perverse feature of assigning a new random value to any loci where the parents agree. The remaining genes are transmitted as per one-point crossover. Disrespectful combination is the complement of uniform crossover in that it *respects differences* rather than similarities. New schemata may be created by combination or by the macro-mutations induced by the agreeing genes.[2]

a	00011 | 101011
b	10101 | 100110
one-point	a0aa1 10bb1b
uniform	?0??1 10??1?
disrespectful	a?aa? ??bb?b

Figure 2: Disrespectful crossover is contrasted with one-point, and uniform crossover. In disrespectful crossover gene values that occur to the left of the crossover point are resolved in favor of parent *a*, and those to the right are resolved in favor of parent *b*, except where the gene values of the parents agree. *Genes at loci where the parents agree are replaced with random alleles.*

With the aid of these three crossover operators we can compare the operation of an operator that only preserves similarity, with one that only allows combination, and with regular one-point that supplies both. Our purpose is to understand what the different components of a successful recombination operator might be, and more specifically to ascertain whether the combination of distinct building-blocks can play any part in the operation of the GA.

For these purposes we will use a hierarchical building-block problem, hierarchical-if-and-only-if (H-IFF), from previous work [Watson et al 1998, Watson & Pollack 1999b]. This problem is designed to investigate the class of problems for which genetic algorithms are well suited. The work in this paper continues to delineate the properties of this type of problem, and to explore the essential characteristics of a GA that will solve it.

We will see that the GA, using deterministic crowding [Mahfoud 1995] as a diversity maintenance technique, and one-point crossover is able to solve H-IFF. Since uniform crossover does not succeed, we conclude that similarity preserving is not sufficient. Conversely, disrespectful recombination does succeed. This indicates that the critical aspect of recombination in this problem is combining different schemata. The reader may find it surprising that this operator succeeds - it means that the massive disruption caused by this operator apparently does not matter. We will discuss the particular properties of the problem we are using that make this possible, and note that disruptive recombination fails completely when applied to a variant of the problem that should be easy.[3] In this variant of H-IFF the value of competing building-blocks is depressed to zero, and those blocks that remain

[1] Two-point crossover, or in general, n-point crossover for low n, exhibits the same properties.

[2] The point of the operator is to supply combination without preserving similarity - but, we shall investigate later whether the macro-mutation aspect of the operator may be valuable.

[3] Recall that the operator was for the purposes of illustrating a point - we would never (almost never) propose disrespectful crossover as a serious option for an applied GA.

are separable.[4] The other side of the coin is that uniform crossover works very well on this easy variant. One-point crossover, which offers both similarity preservation and combination, succeeds always. It is interesting to note that nearly all building-block problems in the GA literature are separable [Watson et al 1998]. Thus we suggest that the reason the community has been unable to pin-down an understanding of the combinative aspects of the GA may be simply because the class of problems discussed has been inappropriate. An alternative explanation is that the combinative aspects of recombination are only effective in a particular class of problem - the extent of this class is yet to be determined.

The remaining sections of this paper are organized as follows: The next section describes the hierarchical building-block function that we will use in these investigations. Section 3 describes the genetic algorithm details and our experimental set-up. Section 4 presents results and discusses the conditions where each operator is successful. Section 5 concludes.

2 HIERARCHICAL-IF-AND-ONLY-IF

Previous work [Watson et al 1998, Watson & Pollack 1999b] introduced a test problem which is specifically designed to investigate the class of problems for which GAs are well suited. Like the second version of the Royal Roads problem defined in [Forrest & Mitchell 1993], it is a hierarchical building-block problem. But unlike the Royal Roads, the building-blocks in H-IFF are not separable. In H-IFF the building-blocks are strongly and non-linearly dependent on one another, i.e. the optimal schemata for one block is strongly dependent on the setting of bits in other blocks.

In H-IFF the interdependency of blocks is implemented via two sets of competing schemata. That is, although blocks at one level in the hierarchy are confined to non-overlapping partitions as in other building-block problems, each partition has two optimal settings for the bits therein. These competing schemata are maximally distinct – specifically, all-ones and all-zeros. Which of these two schemata should be used depends on which has been used in a neighboring block – if the neighboring block is based on ones then so should the block in question, if zeros then zeros. This compatibility of blocks is rewarded by additional fitness contributions from the next level up in a hierarchical structure. Each correct pair of blocks creates a new single block for the next level in the hierarchy. The desirable setting for this meta-block is determined by *its* neighboring meta-block, and so on, up the hierarchical levels.

The fitness of a string using H-IFF can be defined using the recursive function given below. This function interprets a string as a binary tree and recursively decomposes the string into left and right halves. Each resultant sub-string constitutes a building-block and confers a fitness contribution equal to its size if all the bits in the block have the same allele value - either all ones or all zeros. The fitness of the whole string is the sum of these fitness contributions for all blocks at all levels.

$$f(B)=\begin{cases} 1, & \text{if } |B|=1, \text{ else} \\ |B| + f(B_L) + f(B_R), & \text{if } (\forall i\{b_i=0\} \text{ or } \forall i\{b_i=1\}), \\ f(B_L) + f(B_R), & \text{otherwise.} \end{cases} \quad \text{Eq. 1}$$

where B is a block of bits, $\{b_1,b_2,...b_n\}$, $|B|$ is the size of the block=n, b_i is the ith element of B, and B_L and B_R are the left and right halves of B (i.e. $B_L=\{b_1,...b_{n/2}\}$, $B_R=\{b_{n/2+1},...b_n\}$). The length of the string evaluated, n, must equal 2^p where p is an integer (the number of hierarchical levels).

Some features of this apparently simple function should be highlighted. Since both ones and zeros are rewarded each partition contains two equal-fitness competing schemata. Each block-solution, either ones or zeros, represents a schema that contains one of the two global optima at all-ones or all-zeros. Local optima in H-IFF occur when incompatible building-blocks are brought together. For example, consider "11110000"; viewed as two blocks from the previous level (i.e. "1111" and "0000") both blocks are good - but when these incompatible blocks are put together they create a sub-optimal string that is maximally distant from the next best strings i.e. "11111111" and "00000000". (See solid curve in Figure 3)

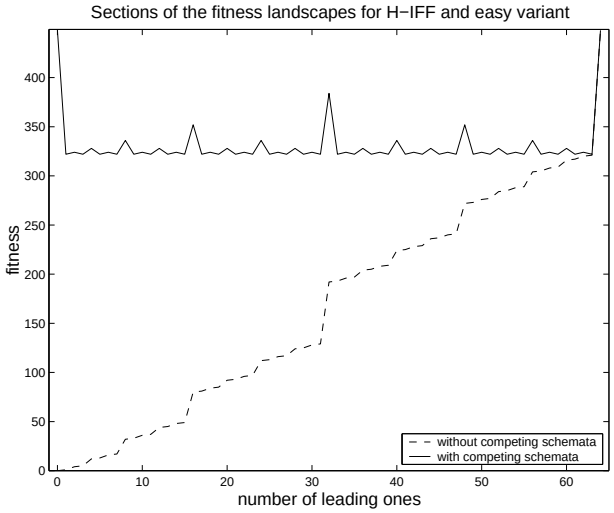

Figure 3: Sections through the H-IFF landscape. The solid curve shows the regular H-IFF problem (i.e. with competing building-blocks), as per Equation 1. The dashed curve shows H-IFF without competing building-blocks (Equation 2). In both cases the curves show a particular section through a 64-bit landscape starting from all zeros on the left and ending with all ones. Specifically, they show the fitness of the strings "000...0", "100...0", "110...0", ..."111...1". This reveals the local optima in Eq. 1 and the 'easy' nature of Eq.2.

[4] This variant of H-IFF is equivalent to a hierarchically consistent form of the second Royal Roads function defined in [Forrest & Mitchell 1993].

However, although local optima and global optima are distant in Hamming space, they may be close in recombination space [Jones 1995], for example, consider a population containing both "11110000" and "00001111" individuals. Thus H-IFF exemplifies a class of problems for which recombinative algorithms are well-suited. Our previous work [Watson et al, 1998] showed that H-IFF is easy for a GA to solve given that diversity in the population is maintained and genetic linkage is tight:

- Diversity maintenance methods are addressed in [Watson & Pollack 1999b], alternative methods are analyzed in [Watson & Pollack 2000], and in this paper we will use the diversity maintenance technique we have found best so far, which is *deterministic crowding* [Mahfoud 1995].

- This paper does not address problems of poor genetic linkage - all of the following experiments use building-blocks where the relevant genes are adjacent on the genome (as defined in Eq.1). It is known that one of the conditions for successful crossover is the tight-linkage of genes [Altenburg 1995]; our experiments here concern different aspects of crossover, as discussed above. However, it is worth noting that although the performance of one-point crossover is poor when applied to a variant of H-IFF that has random linkage, the performance of uniform crossover is approximately the same as this poor level of performance regardless of linkage, [Watson et al 1998]. Algorithms to address poor linkage are addressed in [Watson & Pollack, 1999a].

Variations and more general forms of HIFF that allow different alphabets, different numbers of sub-blocks per block (instead of pairs), unequal fitness contributions of competing blocks, and the construction of other hierarchically-consistent building-block functions, are defined in [Watson & Pollack 1999b]. For the purposes of this paper, the canonical form given above and one variation will be useful. Specifically, a variation that assigns unequal fitness contributions will enable us to decrease the strength of competition between competing schemata.[5] (See dashed curve in Figure 3)

The fitness of a string, $f(B)$, in H-IFF without competing building blocks is given by:

$$f(B)=\begin{cases} 1, & \text{if } |B|=1 \text{ else,} \\ |B| + f(B_L) + f(B_R), & \text{if } (\forall i\{b_i=1\}), \\ f(B_L) + f(B_R), & \text{otherwise.} \end{cases} \quad \text{Eq. 2}$$

where B, |B|, B_L and B_R are as per Equation 1.

[5] Two schemata *compete* to the extent that they are both desirable (have high fitness contributions), and that the allele values they specify (at shared loci) disagree. In H-IFF competing blocks share all loci and disagree at all of them. *Biased* H-IFF [Watson & Pollack 1999b] controls competition by controlling the relative desirability of these blocks. Figure 3 shows biased H-IFF where the value of blocks based on zeros is depressed to zero.

3 EXPERIMENTAL METHODS

For the main comparison of crossover operators we will use the same underlying GA throughout. H-IFF requires that the GA does not converge - if this is allowed, the GA will become trapped in local optima just as a hill-climber would [Watson et al 1998]. To prevent this, earlier work used a resource-based diversity maintenance technique that required knowledge of the block structure. Here we are able to report that an off-the-shelf technique, that has no knowledge of the problem structure, also works very well on H-IFF. This is deterministic crowding, DC, [Mahfoud 1995].

3.1 DETERMINISTIC CROWDING (DC)

Deterministic crowding operates on the premise of restricted *competition* rather than restricted *mating*. Any individual may be recombined with any other, but an offspring is likely to replace an individual that is genotypically similar. In fact, competition is restricted to parents and their own offspring. Restricted competition assists in preventing convergence because sub-populations that are occupying different niches need not out perform one-another to propagate. However, it does not prevent them from mating and producing an offspring that might be superior. DC applies naturally to a *steady state* algorithm (as opposed to a *generational* algorithm that produces an entire population of offspring before any replacements are made). Two individuals are selected at random to be parents. These produce two offspring. Each offspring competes with one of the parents. The competitive pairs of offspring and parents are chosen so as to minimize the difference between the offspring and the parents (Figure 4).

- Initialize population to random strings.
- Repeat until satisfied:
  - Pick two parents, *p1* & *p2*, at random from the population.
  - Produce a pair of offspring, *c1* & *c2*.
  - Pair-up each offspring with one parent according to the pairing rule below.
  - For each parent/offspring pair, if the offspring is fitter than the parent then replace the parent with the offspring.

Pairing rule: if $H(p1,c1)+H(p2,c2) < H(p1,c2)+H(p2,c1)$ then pair *p1* with *c1*, and *p2* with *c2*, else pair *p1* with *c2*, and *p2* with *c1*, where H gives the genotypic Hamming distance between two individuals.

Figure 4: A simple form of a GA using deterministic crowding as used in our experiments.

3.2 A CROSSOVER RATE FOR DC

Usually, crossover in GAs is not applied at every reproduction but applied with some probability or 'crossover rate'. Without mutation, a reproduction

without recombination cannot create new schemata - however, in the regular GA, it can duplicate whole individuals. In deterministic crowding this is not so. A 'reproduction' without recombination has absolutely no effect on the population - individuals cannot be duplicated since offspring, at most, replace only one parent. In order to re-introduce a crossover rate we produce offspring by recombination with some probability, *c* (*c*=0.9 in the experiments that follow), and produce two offspring that are both copies of one parent with probability 1-*c*. This gives us some control over the restriction of competition that deterministic crowding provides.

Note that this duplication of individuals does 'respect' the schemata provided by a parent and allows a method for schemata to be propagated without disruption even when using disrespectful recombination. But it should be clear that this is a different issue from the nature of the recombination operator. It is quite natural for a GA to duplicate good individuals and the schemata they contain. But, as Syswerda indicates, in regard to recombination operators, we are not interested in "string gains" but rather the construction of *new* schemata [Syswerda 1989] which this duplication does not allow.

3.3 DISRESPECTFUL HCT

After the main comparison experiments we will discuss whether the successful operation of disrespectful recombination is perhaps simply because of its disruptive properties. That is, the more similar the parents are the more random bits are injected into the offspring. To investigate whether this kind of *convergence controlled variation* [Eshelman et al 1996] may be responsible for the success of the operator we contrast it with a different operator. This operator borrows from the "headless chicken test" of Jones [1995]. Jones' test is designed to verify whether the genes supplied by crossover are equivalent to 'macro mutations' and, accordingly, whether the exploration enabled by crossover is akin to making random jumps. The difference with our operator is that the random bits are placed exactly where the parents agree rather than between arbitrary crossover points. The corresponding 'disrespectful headless chicken test' (disrespectful HCT) uses the operator in Figure 5.

a	00011|101011
b	10101|100110
disrespectful	a?aa? ??bb?b
disrespectful HCT	a?aa? ??aa?a

Figure 5: The 'recombination' operator used in the 'disrespectful headless chicken test'. *Disrespectful HCT* ignores the crossover point and resolves all alleles that disagree in favor of parent *a*. Parent *b* supplies no genes but instead is used to 'focus' random mutations. Specifically, all loci where the parents agree are assigned random allele values.

4 RESULTS AND DISCUSSION

In the experiments that follow we use a GA with the following parameters: population size 1000, mutation rate 0, crossover rate 0.9 (see Section 3.2), maximum evaluations 10^6. The selection and replacement method of the deterministic crowding algorithm is detailed in Figure 4. To measure performance we recorded simply how many of 30 runs found a global optimum in a 64-bit problem, using each crossover operator. The most evaluations required by any run that succeeded was less than 604,000 – indicating that our evaluation limit was sufficient.

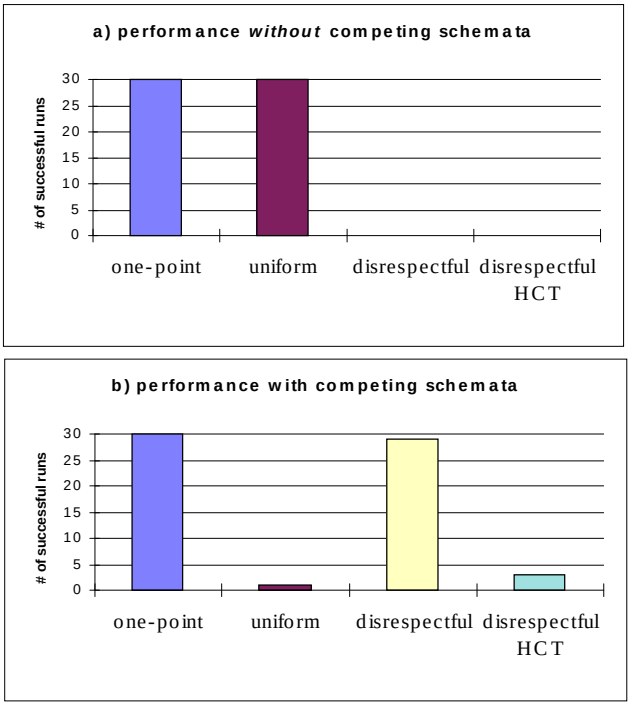

Figure 6: Performance of GA using deterministic crowding and various crossover operators. a) Performance on the 'easy' problem where building-blocks are separable. b) Performance on the standard H-IFF where blocks compete strongly. A control experiment using the 'disrespectful HCT' operator (Figure 5), will be discussed in Section 4.2).

4.1 DISCUSSION

The results show that uniform crossover, which is respectful of the parents' similarities is successful on the variant of H-IFF that has no competing schemata. They also show that disrespectful crossover, that randomizes the parents' similarities and instead combines their differences, is successful on the standard H-IFF that has strongly competing building-blocks. One-point crossover, that both respects similarities and combines differences is successful at both of these extremes.

We should understand why it is that the operators succeed and fail where they do. Firstly, uniform crossover: In H-IFF without competing building-blocks

any block found at lower levels in the hierarchy will be useful in finding blocks at higher levels. Thus the problem has no local optima (see dashed curve in Figure 3) and provides a gradient towards the global optimum at all points in the landscape. Uniform crossover is certainly capable of solving a problem where the mutation landscape is this easy.[6] However, in H-IFF with competing building-blocks, there are many local optima (2^{32} in fact, [Watson et al 1999b]). The distance between local optima and the next-best optima increases exponentially as hierarchical levels are ascended - the second best optima in the whole problem are 32 bits different from either global optimum. It is therefore highly unlikely that the macro-mutations afforded by uniform crossover will perform a jump of this size. However, any given run may be fortunate enough to avoid this worst-case scenario and one run was able to find the optima after all.

Secondly, disrespectful crossover: We know that crossover between two individuals such as "00001111" and "11110000" can produce the global optimum at "11111111" and "00000000". This is the recombination of low-order schemata to find higher-order schemata that the building-block hypothesis describes. The disrespectful feature of our operator is not problematic in these cases since the parents have no alleles in agreement. Given that the population is appropriately diverse, which evidently it is, such recombination is able to escape from the local optima that H-IFF creates. Such recombination events exploit the 'crossover landscape' rather than the mutation landscape. But, disrespectful crossover has severe disadvantages too. It is not able to successfully cross "00111111" with "00001111", for example. This will create "??11????" where the "?"s are randomly assigned. So most crossovers will be disastrous – creating massive disruption and useless offspring – just the sort of case imagined by the "two peaks make a valley" perception of crossover mentioned in our introduction. But this disruptive crossover still succeeds. Evidently, the selection of good offspring when they do occur is strong enough to overcome the disruptive effects of the operator.

This is a lesson we can take away: in some cases, what matters about a crossover operator is not the likelihood of successful offspring *on average* but rather its creative potential. The average offspring using uniform crossover has a much higher fitness than the average offspring of our disruptive operator. But disrespectful crossover still has the occasional offspring that exploits the combination of building-blocks. Chen [1999] concurs that in some circumstances creative potential is more important than 'average expectation', though he makes this point to defend the relatively high disruption of uniform crossover compared to one-point crossover.

But why does disrespectful crossover not succeed on the easy problem? As noted, in order for a schemata to be propagated from a parent to an offspring under this operator the two parents must *disagree* at all loci. This is very unlikely in the easy H-IFF where there are no competing building-blocks. So, recombination with this operator, in this problem, is even more disruptive than in H-IFF with competing blocks. (In a problem where each block has more than one desirable solution, the condition that parents disagree is not so improbable.) Moreover, the macro-mutations applied by disrespectful crossover are focused on exactly those parts of the individuals where they are agreed. Since Chen's hypothesis that schemata common to above average individuals are above average [Chen 1999] holds for this problem, this disruption is not at all useful. In fact, on this problem, there is no useful variation in the algorithm at all. Those individuals subject to recombination have good schemata destroyed (unless the second parent happens to have complementary zeros, which is highly unlikely), and those individuals that are not subject to recombination (see Section 3.2) are duplicated unchanged.

Finally, one-point crossover: This has the advantages of both preserving similarity and combination. And one-point crossover succeeds on both extremes of the problem. By separating out the different aspects of recombination in the other operators we now know that combination is an essential part of the operation of one-point crossover on H-IFF. We also know, by the same reasoning, that preserving similarity is an essential part of the operation of one-point crossover on the version of H-IFF without competing building-blocks.

4.2 DISRESPECTFUL HCT RESULTS

Earlier we mentioned that, in addition to combination, disrespectful crossover also offers a different kind of 'macro-mutation' [Jones 1995] from that offered by uniform crossover. Specifically, whereas uniform crossover may discover schemata by randomizing genes that disagree between parents, disrespectful crossover may discover new schemata by randomizing genes that agree between parents (or by combination). Our control experiment is designed to ascertain whether these macro-mutations are the source of success for disrespectful crossover. The disrespectful HCT operator, described in Section 3.3, separates the macro-mutations from the combination. Figure 6 includes the results of this operator. We see that it is not successful on the easy problem, and not successful on regular H-IFF except in a few runs. Thus it is the combinative feature of disrespectful crossover that is responsible for success in the main experiments.

4.3 MACRO-MUTATION AND DIVERSITY

However, there is an alternative view to the macro-mutations supplied by disrespectful crossover. Specifically, the more similar parents are on average, the more mutation is applied - this makes the operator a form of "convergence controlled variation" [Eshelman et al 1996]. Moreover, this mutation is directed specifically at those loci where convergence is strongest. We might call it "convergence sensitive macro-mutation". We

[6] It is also capable of solving harder problems where the macro mutations are useful in escaping local optima.

know from the previous section that this feature alone is insufficient to solve H-IFF (combination is also required) but it does have potential as a diversity maintenance technique.

In the above experiments we used deterministic crowding to maintain diversity – previous work showed that the regular GA (with one-point or uniform crossover) cannot solve H-IFF without a diversity maintenance method. But, here we tried a GA using disrespectful crossover on a steady state GA without deterministic crowding or any other form of diversity maintenance. We needed to adjust a few parameters (tournament size 4 (mating best two and replacing worst), cross-over rate 0.95), but to our own surprise, it worked. All 30 runs were successful.

This operator has the desirable property that any schemata that become too common in the population are penalized. But the effect is extreme; a schema can only possibly survive if there is some other schema in the population that disagrees with the first at all loci. Thus the operator does not work when applied to the easy problem since there is only one type of schemata that is desirable (no runs were successful). So, although it was successful at maintaining diversity in the standard H-IFF, it seems unlikely that disrespectful crossover could be used as a serious diversity maintenance technique. In the meantime, it makes an interesting side-effect.

4.4 CAVEATS

We should emphasize that although the results in Figure 5 seem 'clear cut', different choices in the parameters of the experimental set-up produce different results. For example, uniform crossover succeeds on H-IFF with competing building-blocks if the values of competing blocks are imbalanced – H-IFF bias, for biases less than about 0.8 [Watson & Pollack 1999b], are solvable. In this case uniform crossover does not need to perform the combination that one-point (and disrespectful) crossover affords since the distances between local optima and next-best optima are reduced. On the other hand, disrespectful crossover (with or without deterministic crowding) can succeed on the easy version of the problem if we reintroduce a little additional mutation (which was excluded from the above experiments for clarity). Thus, all algorithms can solve the easy problem (when using additional mutation), but only algorithms with combination (i.e. one-point and disrespectful) can solve the standard H-IFF.

Also, thus far, we have only talked about whether an algorithm succeeded or not, and not the time to completion. Table 1 shows the average number of evaluations required to find the global optima in those algorithms that succeeded reliably. We see that disrespectful crossover on H-IFF takes about 13 times longer to succeed than uniform on the easy problem, and still more than 9 times longer than one-point crossover compared on equal problems.

Most importantly, the relevance of these results (as for any) depends on the nature of the problem at hand. Here we have only examined two variants of one problem. Whether combination or macro-mutations are required in a given problem is to be judged case by case. However, it is notable that the above experiments indicate that the utility of uniform crossover is best on the variant that is separable, and that separability is a limitation that is common to problems in the GA literature [Whitley 1995]. In any case, it only requires one example to disprove the notion that preserving common schemata is the *only* 'source of power' in recombination.

	'easy' H-IFF	H-IFF
one-point	26k	40k
uniform	28k	-
disrespectful	-	366k
disrespectful HCT	-	-
Disrespectful w/o crowding	-	127k

Table 1: Average number of evaluations to find optimum on runs which were successful. The three operators and the disrespectful HCT are shown (with deterministic crowding), and disrespectful crossover is also shown without deterministic crowding. Averages are shown for those algorithms that were reliably successful.

5 CONCLUSIONS

The status of the building-block hypothesis is controversial – some believe that the combination of low-order building-blocks to find higher-order building-blocks does not play a significant role in the operation of the GA. This view is typified by 'recombination' operators that do not permit combination, such as uniform crossover. Indeed, there is considerable evidence that many of the problems used in the GA literature, designed to test the operation of the GA, do not require combination (for example, [Mitchell et al 1992, Syswerda 1989, Jones 1995]). It is also widely believed that disruption caused by mating individuals that are too dissimilar is best avoided. Thus, it may not be wise to assume that combination is required, and it may be wise to expect crossover operators that preserve the similarity of parents to be preferable. Chen is undoubtedly correct that the heuristic of 'common schemata' being 'good schemata' can be a valuable method for focusing variation on inferior parts of the genotype. And, Syswerda is correct that we might prefer an operator like uniform crossover if we do not know that genetic linkage is tight. However, we should not allow such wisdom to become dogma. The value of an operator, or any aspect of an algorithm, is conditioned on the problems to which it is applied.

In this paper we have separated the similarity preserving property of recombination from the combination property. For this purpose (and for this purpose only), we introduced a new crossover operator that destroys

schemata common to both parents. Using this and uniform crossover we showed that preserving schemata common to the parents is not sufficient to solve a particular building-block problem. Whereas, combination is sufficient, despite the considerable disruption that this operator causes. However, each operator has its own niche. In a problem that should be easy – uniform succeeds but disrespectful fails.

We also showed that the macro-mutations afforded by both uniform and disrespectful crossover have their limitations (whether the mutations are 'inside' (disrespectful HCT) or 'outside' (uniform) the similarities of the parents). Neither is sufficient to solve both variants of the problem.

One-point crossover, however, does succeed on both problems, and from our explorations we have a clearer picture of the different characteristics that are at work in this operator.

Acknowledgments

Thanks to Stephen Chen for the motivating discussion of this work, to Sevan Ficici for suggesting that there might be a way to design an operator that does not preserve similarity, and to Jon Rowe and Christopher Ronnewinkel for helpful discussion.

References

Altenberg, L, 1995 "The Schema Theorem and Price's Theorem", in *Foundations of Genetic Algorithms 3*, eds. Whitley & Vose, pp 23-49, Morgan Kaufmann, San Francisco.

Chen, S, & Smith, S, 1996, "Commonality and Genetic Algorithms", tech. report CMU-RI-TR-96-27, Robotics Institute, Carnegie Mellon University, (December 1996).

Chen, S, 1999, "Is the Common Good? A New Perspective Developed in Genetic Algorithms", PhD diss., Robotics Institute, Carnegie Mellon University.

Deb, K & Goldberg, DE, 1989, "An investigation of Niche and Species Formation in genetic Function Optimization", *ICGA3*, San Mateo, CA: Morgan Kaufmann.

Eshelman, LJ, Mathias, KE, & Schaffer, JD, 1996, "Convergence Controlled Variation." In *Foundations of Genetic Algorithms 4*, Belew and Vose, eds. Morgan Kaufmann.

Forrest, S & Mitchell, M, 1993, "Relative Building-block fitness and the Building-block Hypothesis", in Whitley, D, ed. *FOGA 2*, Morgan Kaufmann, San Mateo, CA.

Goldberg, DE, 1989, "Genetic Algorithms in Search, Optimisation and Machine Learning", Reading Massachusetts, Addison-Wesley.

Goldberg, DE., Cantupaz, E., Kargupta, H. & Horn, J., 1996, "Towards A Theory Of Deme Sizing For Multiple-Run genetic Algorithms", *Foundation of Genetic Algorithms (FOGA'96)*. Illinois Genetic Algorithm Laboratory Technical Report Number 96002.

Holland, JH, 1975, "Adaptation in Natural and Artificial Systems", Ann Arbor, MI: Michigan University Press.

Jones, T, 1995, Evolutionary Algorithms, Fitness Landscapes and Search, PhD dissertation, 95-05-048, University of New Mexico, Albuquerque.

Mahfoud, S, 1995, "Niching Methods for Genetic Algorithms", PhD diss., Dept. General Engineering, University of Illinois. Also, IlliGAL Report No. 95001.

Mitchell, M, Forrest, S, & Holland, JH, 1992, "The royal road for genetic algorithms: Fitness landscapes and GA performance", *Procs. of first ECAL*, MA. MIT Press.

Radcliffe, NJ. 1991, "Forma Analysis and Random respectful Recombination." *In Proc. Fourth International Conference on Genetic Algorithms.*

Starkweather, T. Whitley, D. & Mathias, K., 1991, Optimization using Distributed Genetic Algorithms. in *PPSN1*, pp. 176-185.

Syswerda, G, 1989, "Uniform Crossover in Genetic Algorithms." *In Proc. Third International Conference on Genetic Algorithms.*

Watson, RA, Hornby, GS & Pollack, JB, 1998, "Modeling Building-Block Interdependency", *Parallel Problem Solving from Nature, proceedings of Fifth International Conference /PPSN V*, Springer 1998, pp.97-106 .

Watson, RA, & Pollack, JB, 1999a, "Incremental Commitment in Genetic Algorithms", *Proceedings of 1999 Genetic and Evolutionary Computation Conference (GECCO 99).* Banzhaf, Daida, Eiben, Garzon, Honavar, Jakiela, Smith, eds., Morgan Kauffmann, pp.710-717.

Watson, RA, & Pollack, JB, 1999b, "Hierarchically-Consistent Test Problems for Genetic Algorithms", *Proceedings of 1999 Congress on Evolutionary Computation (CEC 99).* Angeline, Michalewicz, Schoenauer, Yao, Zalzala, eds. IEEE Press, pp.1406-1413.

Watson, RA, & Pollack, JB, 2000, "Analysis of Recombinative Algorithms on a Hierarchical Building-Block Problem", *Foundations of Genetic Algorithms /FOGA 2000*, submitted.

Whitley, D, Mathias, K, Rana, S & Dzubera, J, 1995 "Building Better Test Functions", *ICGA-6*, editor Eshelman, pp239-246, Morgan Kauffmann, San Francisco.