The Usefulness of Recombination

Wim Hordijk and Bernard Manderick

[1] Santa Fe Institute, 1399 Hyde Park, Santa Fe, NM 87501, email: wim@santafe.edu
[2] Computer Science Dept., Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel,
Belgium, email: bernard@arti.vub.ac.be

Abstract. In this paper, we examine the usefulness of recombination from two points of view. First, the problem of crossover disruption is investigated. This is done by comparing two Genetic Algorithms with different crossover operators (one-point and uniform) to each other on NK-landscapes with different values of K relative to N, and with different epistatic interactions (random and nearest neighbor). Second, the usefulness of recombination in relation to the location of local optima in the fitness landscape is investigated.
There appears to be a clear relation between the type of fitness landscape and the type of recombination that is most useful on this landscape. Furthermore, there also is a clear relation between the location of local optima in the fitness landscape and the usefulness of recombination.

1 Introduction

The concept of a fitness landscape has proved to be very useful in thinking about evolutionary processes. Given a certain problem, a fitness landscape is the distribution of fitness values over the space of possible solutions for this problem, where the solutions have a neighborhood relation determined by some metric. For example, the genetic algorithm usually uses bit strings to encode solutions and the Hamming distance between such strings defines a metric. The fitness values are determined by some fitness function, which takes as input a genotype, the (genetic) coding of an individual, and returns a real value denoting this individual's fitness. The higher the fitness, the better a solution for the given problem this individual is.

By assigning a fitness value to every possible individual (or genotype), a more or less "mountainous" landscape arises, where the highest peaks denote the best solutions. An evolving population is now envisioned as adapting on such a landscape, in search for the highest peaks.

In Nature, *recombination* is used for constructing new genotypes during reproduction. This means that different parts of two parent genotypes are recombined to form two offspring genotypes. This recombination is copied in evolutionary algorithms that are used for solving difficult problems. It is implemented in the form of different crossover operators.

The role of crossover is a two-edged knife. On the one hand, crossover is used for exploration of the fitness landscape, because it can make large jumps across such a landscape (in terms of the metric by which the landscape is defined). On the other hand, there is the danger of crossover disruption, or the breaking apart of good solutions that were already found. In most of the studies on crossover, the emphasis is on finding a way to minimize this crossover disruption. This paper investigates both aspects of crossover: the power of exploration and the danger of disruption.

First, the problem of crossover disruption is dealt with. Next, the usefulness of recombination in relation to the location of optima in the fitness landscape is examined. With the results of these two investigations, the validity of the next statement made by Kauffman is assessed (see [5]):

> "*recombination is useless on uncorrelated landscapes but useful under two conditions: (1) when the high peaks are near one another and hence carry mutual information about their joint locations in the fitness landscape and (2) when parts of the evolving individuals are quasi-independent of one another and hence can be interchanged with modest chances that the recombined individual has the advantage of both parents*".

Different fitness landscapes are constructed using the NK-model (see [5]). The parameters of this model can be used to tune the landscape from smooth (small fitness differentials between neighboring points) to very rugged (large fitness differentials between neighboring points). The more rugged the landscape is, the less information the fitness of one point gives about the expected fitness of another point.

In Section 2, the NK-model is explained in more detail. Section 3 deals with the problem of crossover disruption by comparing two Genetic Algorithms, with different crossover operators, on different fitness landscapes generated by the NK-model. Section 4 then examines the relation between the usefulness of recombination and the location of optima. Finally, in Section 5 the validity of Kauffman's statement about the usefulness of recombination is assessed.

2 The NK-model

The *NK-model* was introduced by Kauffman to have a problem-independent model for constructing landscapes that can be tuned from smooth to rugged (see [5]). This tunability led [6] to use these landscapes to investigate the performance of genetic algorithms. The main parameters of the model are N, the number of parts, or genes, in the system, and K, the number of genes that epistatically influence a particular gene. Another parameter is A, the number of alleles every gene has. It appears that almost all properties of the NK-model are independent of A, so in this paper A will be set at 2, i.e., the genotypes are bit strings (strings of 0's and 1's). The NK-model is also explained in terms of bit strings (i.e., the two allele case).

Suppose every bit b_i ($i = 1, \ldots, N$) in the bit string $\underline{b}$ is assigned a fitness w_i of its own. The fitness w_i of a bit b_i, however, does not only depend on the value

(0 or 1) of this specific bit, but also on the value of K other bits b_j in the same bit string ($0 \leq K \leq N-1$). These dependencies are called *epistatic interactions.*

So, the fitness contribution of one bit depends on the value of $K+1$ bits (itself and K others), giving rise to a total of 2^{K+1} possibilities. Since, in general, it is not known what the effects of these epistatic interactions are, they are *modelled* by assigning to each of the 2^{K+1} possibilities at random a fitness value drawn from the Uniform distribution between 0.0 and 1.0. Therefore, the fitness contribution w_i of bit b_i is specified by a list of random decimals between 0.0 and 1.0, with 2^{K+1} entries. This procedure is repeated for every bit b_i, $i = 1, \ldots, N$ in the bit string $\underline{b}$.

Having assigned the fitness contributions for every bit in the string, the fitness of the entire bit string, or genotype, is now defined as the average of the contributions of all the bits:

$$W = \frac{1}{N} \sum_{i=1}^{N} w_i$$

One further aspect of the NK-model characterizes how the K epistatic interactions for each bit are chosen. Generally, this is done in one of two ways.

The first way is by choosing them at random from among the other $N-1$ bits. This is called *random interactions.* It is important to note that no reciprocity in epistatic influence is assumed. This means that if the fitness of bit b_i depends on bit b_j, it is not necessary that the reverse also holds. So, the epistatic interactions for a bit are determined independent of the other bits.

The second way is by choosing the K neighboring bits as epistatic interactions. The $K/2$ bits on each side of a bit will influence the fitness of this bit. This is called *nearest neighbor interactions.* To make this possible, *periodic boundary conditions* are taken into account. This means that the bit string is considered as being circular, so the first and the last bit are each others neighbors. Note that for K=0 and $K = N-1$, there is no difference between the two types of interactions. In the first case, the fitness of each bit depends only on its own value, and in the second case, the fitness of each bit depends on the value of all the bits in the string.

The fitness landscapes that result from this NK-model are called *NK-landscapes.* A low value of K gives rise to a rather smooth landscape (small fitness differentials between neighboring points), while increasing K, relative to N, results in an ever more rugged landscape (large fitness differentials). For K=0, the landscape is very smooth and contains only one peak, because for every bit either the value 0 or the value 1 has the highest fitness contribution, so there is one single bit string having the highest fitness contribution at all bit positions, and every other bit string can sequentially be changed to this optimal bit string. For K=$N-1$, the landscape is completely random, because changing the value of one bit changes the fitness contribution of all bits. In this case there is no correlation at all between the fitness of neighboring points.

3 Crossover disruption

The notion of a *schema* is central to understand how a Genetic Algorithm works. A schema is a set of individuals in the search space, and the GA is thought to work by directing the search towards schemata containing highly fit regions of the search space. For example, the string `1**01*00*` is a schema, where a `*` means *don't care*, either value (`0` or `1`) is allowed. In schemata, `0` and `1` are called *defined bits*, the *order* is the number of defined bits, and the *defining length* is the distance between the first and the last defined bit. So, the schema in the example is of order 5 and has defining length 7.

According to the building block hypothesis [3, 2], a Genetic Algorithm works well when short, low-order, highly fit schemata (so-called building blocks) are recombined to form even more highly fit higher-order schemata. So, a GA works well when crossover is able to recombine building blocks to longer, higher-order schemata with a high fitness. On the other hand, it follows from the Schema Theorem [3, 2] that long, high-order schemata are more sensitive to crossover disruption than short, low-order ones. So, opposed to the usefulness of crossover in constructing longer, highly fit schemata, there is the danger of disrupting them again.

To investigate this construction-disruption duality, two types of crossover, one-point and uniform, are compared on NK-landscapes with different types of epistatic interactions, random and nearest neighbor. Uniform crossover is believed to be maximally disruptive, while one-point crossover is more conservative. But this depends highly on the type of epistatic interactions within a genotype.

With one-point crossover, a random crossover point between the first and last bit is chosen, and the parts of the two parents after this crossover point are swapped, thus creating two children.

With uniform crossover, for each bit position on the two children it is decided randomly which parent contributes its bit value to which child.

One-point crossover is more disruptive when the epistatic interactions in a bit string are randomly distributed than when they are the nearest neighbors: with random interactions almost every possible crossover point will affect the epistatic relations of almost all bits in a bit string, while with nearest neighbor interactions only the epistatic relations of the bits in the vicinity of the crossover point are affected.

With uniform crossover, however, there is a large chance that a good configuration of neighboring epistatically interacting bits will be disrupted. But when the epistatic interactions are random, uniform crossover can recombine good values for these interacting bits, while one-point crossover is unable to do this.

Next, the experimental setup for investigating the problem of crossover disruption is described, after which the results of the experiments are presented.

3.1 Experimental setup

Two Genetic Algorithms, with the following parameter values, are applied to different NK-landscapes: a population size of 50, deterministic tournament with tournament size s=3, a crossover rate $p_c = 0.75$, and a mutation rate $p_m = 0.005$

In deterministic tournament selection, s individuals are picked at random from the old population, and the fittest of them is put in the mating pool. This is repeated until the mating pool is as large as the old population. The tournament size s can be used to tune the selection pressure.

One GA uses one-point crossover (GA-ONEP), while the other GA uses uniform crossover (GA-UNIF). Both GA's are allowed to do a total of 10,000 function evaluations. Every 50 function evaluations, i.e. every generation, the maximum fitness in the population is recorded. NK-landscapes with the following values for N and K are taken: N=100, K=0, 2, 5, 25, 50, and 99. Both random and nearest neighbor interactions are considered. All results are averaged over 100 runs, each run on a different landscape, but with the same values for N and K.

3.2 Results

Figures 1 to 3 show the results of applying the two GA's to the different NK-landscapes. The abbreviation RND stands for random interactions, while NNI stands for nearest neighbor interactions. So, GA-ONEP (RND) means the Genetic Algorithm with one-point crossover applied to an NK-landscape with random epistatic interactions. Note that for K=0 and K=99 no distinction is made between random interactions and nearest neighbor interactions, because they are exactly the same for these values of K (see Section 2).

It appears that there are two phases in the search: the first phase consists of finding a good region in the fitness landscape, mainly by global search (crossover), and the second phase consists of trying to find the highest peaks within this region, mainly by local search (mutation). (See [4, 5] for a more detailed discussion about these phases in an adaptive search). Initially, the graphs are increasing rapidly (the first phase), but then they become gradually less steep (the second phase) until they are completely smooth, indicating that the highest peaks in a relatively good region are found. Only the completely random landscape (K=99, Figure 3) does not fit into this picture, because global search appears to be useless on this landscape (see again [4, 5]). Therefore, the case of K=99 is left out of the rest of the analysis.

Since the second phase in the search is dominated by local search (mutation), only the performance in the first phase of the search is evaluated here to examine the usefulness of recombination, which is a global search strategy. Differences in performance in the second phase are mainly a reflection of differences in performance in the first phase. Furthermore, the results are viewed in two ways: taking one type of GA and comparing random with nearest neighbor interactions, and taking one type of epistatic interactions and comparing GA-ONEP with GA-UNIF.

Table 1 presents the results from the first viewpoint. It shows for both types of GA's on which type of landscape (that is, with random interactions (RND) or with nearest neighbor interactions (NNI)) they are better able to find a good region in the landscape in the first phase of the search. "Better able" means either finding such a region faster, or finding a better region (that is, containing higher peaks), or both. An X means that there is no (significant) difference between the two types of epistatic interactions. The significance of the differences is checked with a statistical procedure called the *t-test* (see [1]).

	K=2	K=5	K=25	K=50
GA-ONEP	NNI	NNI	NNI/RND	NNI
GA-UNIF	X	RND	RND	X

Table 1. Comparison of random interactions (RND) with nearest neighbor interactions (NNI) for GA-ONEP and GA-UNIF in the first phase of the search. An entry RND means that the GA works better on a landscape with random interactions than on a landscape with nearest neighbor interactions. An X means that there is no difference.

The table shows that one-point crossover (GA-ONEP) works better on a landscape with nearest neighbor interactions than on a landscape with random interactions. So, one-point crossover is better able to combine configurations of nearby interacting bits (without disrupting them too much again), than configurations of random interactions. The entry NNI/RND for K=25 reflects the fact that the graph of GA-ONEP is initially increasing faster for landscapes with nearest neighbor interactions, but is overtaken by random interactions, for which it eventually finds a better region (that is, containing higher peaks), as can be seen in Figure 2.

The table shows furthermore that uniform crossover (GA-UNIF) works better with random interactions on NK-landscapes with intermediate epistasis (K=5 and K=25). Apparently, for very low and very high epistasis, uniform crossover is just as disruptive, no matter whether the epistatic interactions are randomly distributed or nearby.

Table 2 presents the results from the second viewpoint. It shows for both types of epistatic interactions which type of GA (GA-ONEP or GA-UNIF) is better able to find a good region in the landscape in the first phase of the search ("better able" in the same sense as in Table 1). Again, an X means no (significant) difference.

The table shows that for smooth and rugged landscapes (K=0, 2 and 5) uniform crossover (GA-UNIF) works better than one-point crossover (GA-ONEP) when the epistatic interactions are random. So, for low, random epistasis, uniform crossover is better able to combine building blocks, without disrupting them too much again, than one-point crossover. One-point crossover (GA-ONEP), however, works better than uniform crossover (GA-UNIF) for very rugged land-

	K=0	K=2	K=5	K=25	K=50
RND	GA-UNIF	GA-UNIF	GA-UNIF	GA-ONEP	GA-ONEP
NNI	GA-UNIF	X	GA-ONEP	GA-ONEP	GA-ONEP

Table 2. Comparison of GA-ONEP with GA-UNIF for random interactions (RND) and nearest neighbor interactions (NNI) in the first phase of the search. An entry GA-ONEP point means that one-point crossover works better on that particular landscape than uniform crossover. An X means that there is no difference.

scapes (K=25 and 50) when the epistatic interactions are random. So, for high random epistasis, uniform crossover becomes too disruptive.

For nearest neighbor interactions, it appears that one-point crossover (GA-ONEP) works better than uniform crossover (GA-UNIF) for K=5, 25 and 50 . As expected, uniform crossover is too disruptive, compared with one-point crossover, in these cases.

So, these results show that there is a clear relation between on the one hand the type of recombination that is used (one-point or uniform crossover) and the type, and also the amount, of epistatic interactions (random or nearest neighbor) on the fitness landscape, and on the other hand the usefulness of recombination.

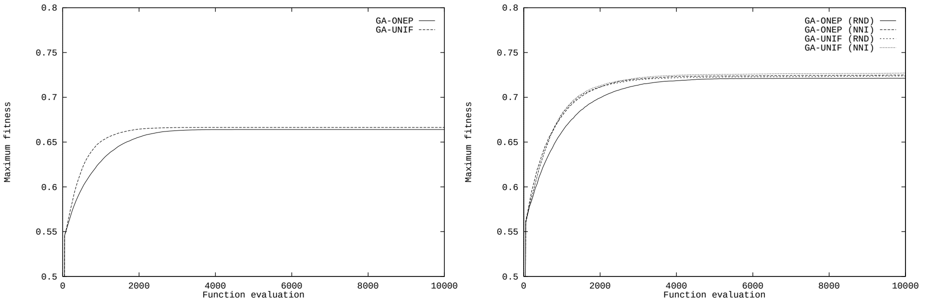

Fig. 1. Left: The maximum fitness of the two GA's on NK-landscapes with N=100 and K=0. Right: The maximum fitness of the two GA's on NK-landscapes with N=100 and K=2.

4 Recombination and the location of optima

In the previous section, we have investigated the role of crossover disruption. Here, we look at an alternative view on crossover which stresses its exploratory

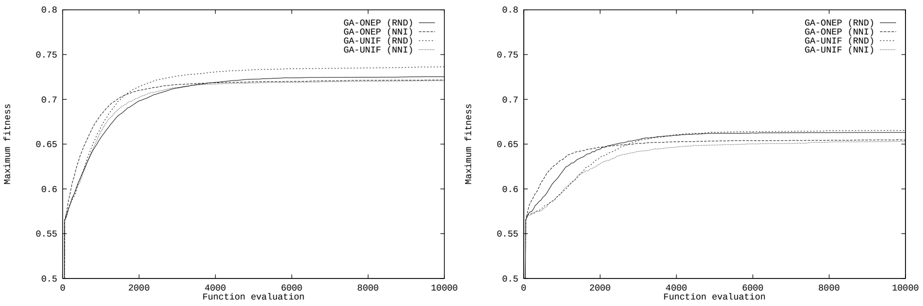

Fig. 2. Left: The maximum fitness of the two GA's on NK-landscapes with N=100 and K=5. Right: The maximum fitness of the two GA's on NK-landscapes with N=100 and K=25.

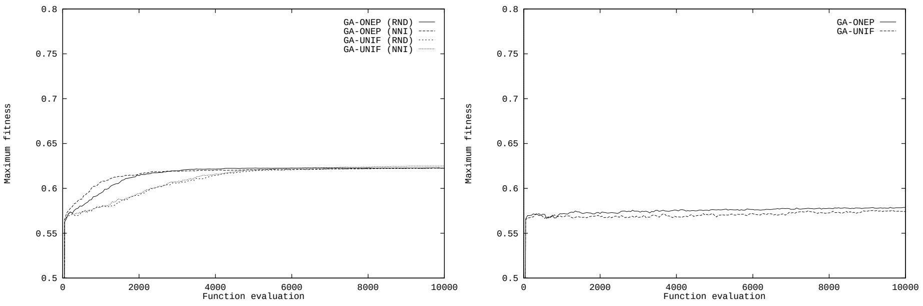

Fig. 3. Left: The maximum fitness of the two GA's on NK-landscapes with N=100 and K=50. Right: The maximum fitness of the two GA's on NK-landscapes with N=100 and K=99.

power. This view is exemplified by S. Kauffman. According to him, the first condition that has to be met, for recombination to be useful, is that the high peaks in the landscape are near one another and hence carry mutual information about their locations in the fitness landscape.

To examine to what extent this condition holds, three search strategies are compared. The first strategy is iterated hill climbing where we start with a population of bit strings and to each bit string we apply local hillclimbing until we get stuck in a local optimum. Then we randomly generate a new population and repeat local hillclimbing. This procedure is repeated until a predefined number of function evaluations is reached and during this procedure we keep track of the best fitness in each generation.

The second and third strategy combine iterated hill climbing with one-point and uniform crossover, respectively. The basic difference with the first strategy has to do with the generation of a new starting population each time we get stuck in local optima. Now, we use these local optima as parents to generate offspring using crossover. This offspring constitutes our new starting population. In the first strategy, the newly generated starting population is completely independent of the previous population.

If peaks in the fitness landscape carry mutual information then the second and third strategy should be able to exploit that information while the first one cannot do so. We have considered two crossover operators to eliminate the possible effect of crossover disruption.

The experiments described below are done on a fixed fitness landscape, where the relative positions of local optima are determined. Both random interactions and nearest neighbor interactions are considered.

4.1 Experimental setup

To investigate the relation between the usefulness of recombination and the location of optima, a fixed NK-landscape is generated for N=100 and K=2, both for random interactions and for nearest neighbor interactions. Three versions of an iterated hillclimbing (IHC) strategy are then applied to each of these two landscapes. Two of these version also incorporate crossover: one using one-point crossover (IHC-ONEP), and one using uniform crossover (IHC-UNIF). A population size of 10 is taken for all three strategies. The three strategies are all allowed to do 50,000 function evaluations. During the run, the maximum fitness in the population is recorded every 50 function evaluations.

Furthermore, the locations of the local optima are determined for both landscapes by applying random ascent hillclimbing with memory (see above) to 10,000 randomly chosen starting points. Every found local optimum is then recorded, together with its fitness.

4.2 Results

Figures 4 show the results of applying the three search strategies, IHC, IHC-ONEP and IHC-UNIF, to the fixed (K=2) fitness landscape for random and nearest neighbor interactions, respectively. It is clear that on the landscape with random interactions both crossover operators are useful. If crossover is applied to the population of local optima, the maximum fitness in the population stays relatively high, indicating that the locations of two optima give information about the locations of other optima. Also, the graphs appear to be gradually increasing. There is not much difference between IHC-ONEP and IHC-UNIF, and both outperform IHC.

For nearest neighbor interactions, however, the distinction is less clear. The IHC-ONEP and IHC-UNIF strategies appear to be just a little better than the IHC strategy during the search, but not much. Crossover contributes just a little in finding good regions in the landscape. Furthermore, the graphs are certainly not increasing, but instead appear to decrease a little after a while. Again, there is not much difference between IHC-ONEP and IHC-UNIF.

Figures 5 show the locations of local optima, relative to the best one found, for the landscapes with random interactions and nearest neighbor interactions, respectively. The fitness of the found optima is plotted against the (normalized) Hamming distance from the fittest local optimum that was found. For the landscape with random interactions 9,970 different local optima were found, while for the landscape with nearest neighbor interaction 10,000 different local optima were found.

There is a clear similarity between the two plots. The optima with a relatively higher fitness tend to be closer to the best optimum than optima with a relatively lower fitness. This shows a feature of the landscapes that Kauffman called a *massif central*: there is one place in the landscape where all the good optima are situated, surrounded by the less good optima (see [5]). This feature is the cause that crossover can help in finding a good region in the landscape: recombining the information of two optima gives a high chance of finding still better optima.

Besides the similarity, there is also one striking difference between the plots, though. For random interactions, the good optima are *much closer* to the best optimum (and thus to each other) than for nearest neighbor interactions. The number of optima with a (normalized) Hamming distance of 0.10 or less from the best optimum is 48 for random interactions, while it is only 2 for nearest neighbor interactions. This explains the difference in crossover performance (relative to no crossover) between the two landscapes. For nearest neighbor interactions, the good optima are just a little too far from the best optimum (and probably also from each other), to give enough information about the location of the highest peaks.

Kauffman already did this landscape analysis himself, and the plots shown here are very similar to his plots, which also show the similarity between random and nearest neighbor interactions (see [5]). But because he used different scales for both plots, the striking difference between the two is much harder to detect. At least, Kauffman does not say anything about it.

So, these results show that there also is a clear relation between the location of local optima in the fitness landscape, and the usefulness of recombination.

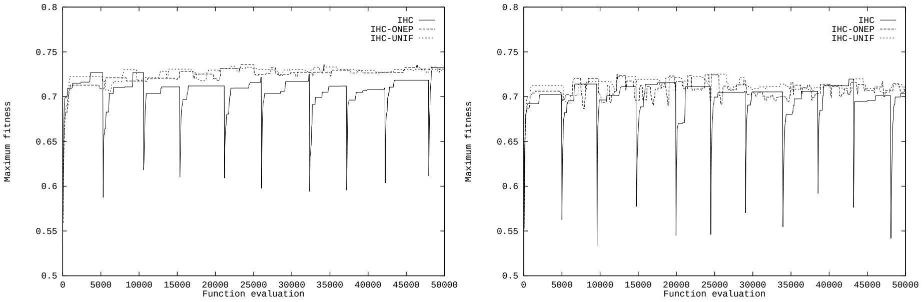

Fig. 4. Left: Comparison of 1-point (IHC-ONEP) and uniform (IHC-UNIF) crossover and no crossover (IHC) on an NK-landscape with N=100 and K=2, random interactions (RND). Right: Comparison of 1-point (IHC-ONEP) and uniform (IHC-UNIF) crossover and no crossover (IHC) on an NK-landscape with N=100 and K=2, nearest neighbor interactions (NNI).

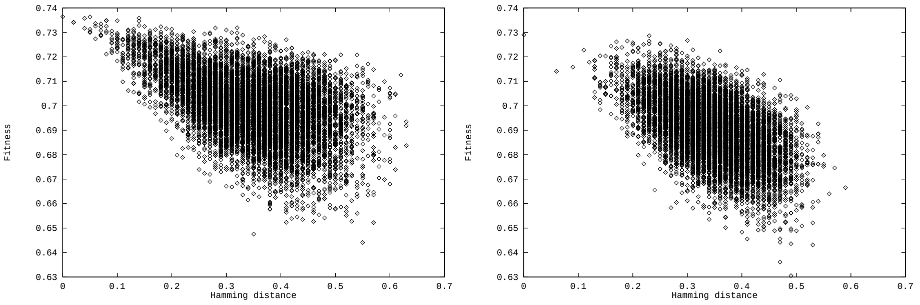

Fig. 5. Left: The correlation between the fitness of local optima and their (normalized) Hamming distance from the fittest local optimum found on an NK-landscape with N=100 and K=2, random interactions. Right: The correlation between the fitness of local optima and their (normalized) Hamming distance from the fittest local optimum found on an NK-landscape with N=100 and K=2, nearest neighbor interactions.

5 Conclusions

There appears to be a clear relation between on the one hand the type of recombination that is used and the type and amount of epistatic interactions on the fitness landscape, and on the other hand the usefulness of recombination.

In the first phase of a search, when a good region in the landscape is searched for by global search (i.e. crossover), one-point crossover works better when the epistatic interactions are nearby than when they are randomly distributed. In the latter case, one-point crossover is too disruptive. On NK-landscapes with intermediate values of K (K=5, 25) uniform crossover works better when the epistatic interactions are randomly distributed. For lower or higher values of K, there is no difference between random and nearest neighbor interactions for uniform crossover.

On smooth and rugged landscapes (K=0, 2, 5) with random interactions, uniform crossover is faster than one-point crossover in finding good regions in the landscape. On very rugged landscapes (K=25, 50) with random interactions, however, one-point crossover is faster than uniform crossover. When the epistatic interactions are the nearest neighbors, one-point crossover is the better type in the first phase of the search on rugged to very rugged landscapes (K=5, 25, 50). As expected, uniform crossover is too disruptive in this case.

Furthermore, there is also a clear relation between the location of local optima in the fitness landscape, and the usefulness of recombination. Recombination is most useful when relatively high optima tend to be near each other. Recombining the information about the location of two optima gives a fair chance of finding even better optima. When the highest optima are not really close enough to each other, recombination becomes less useful.

With these conclusions, the validity of the next statement made by Kauffman can be assessed (see [5]:

> *"recombination is useless on uncorrelated landscapes but useful under two conditions: (1) when the high peaks are near one another and hence carry mutual information about their joint locations in the fitness landscape and (2) when parts of the evolving individuals are quasi-independent of one another and hence can be interchanged with modest chances that the recombined individual has the advantage of both parents"*.

The first condition is validated, considering the conclusion above. The second condition, however, is not validated. As the above results show, the usefulness of recombination depends on the type of recombination that is used and the type and amount of epistatic interactions on the landscape. It is certainly not always necessary that the "parts of the evolving system are quasi-independent of one another". So, the results presented here do not fully support Kauffman's statement about the usefulness of recombination, but imply a more refined one.

References

1. L. J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics.* Duxbury Press, 1987.
2. D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.
3. J. H. Holland. *Adaptation in Natural and Artificial Systems.* MIT Press, 2nd edition, 1992.
4. W. Hordijk. Population Flow on Fitness Landscapes. Master's thesis, Erasmus University Rotterdam, 1994. Available via anonymous ftp at ftp.cs.few.eur.nl in pub/doc/masterstheses.
5. S. A. Kauffman. *Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, 1993.
6. B. Manderick, M. de Weger, and P. Spiessens. The Genetic Algorithm and the Structure of the Fitness Landscape. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150. Morgan Kaufmann, 1991.