$C(N, K)$ Landscapes: An Investigation of Epistasis and Crossover in Real-Valued GAs

Stephen Drake
Department of Informatics,
University of Sussex
Falmer, Brighton, BN1 9QH, UK.
E-mail: stevedr@sussex.ac.uk

Cognitive Science Reseach Paper No. 560

April, 2004

Abstract

A continuous implementation of Kauffman's NK landscapes is proposed in order that real-valued genetic algorithms' (GAs') handling of epistasis might be investigated in the same manner as is often favoured with respect to binary GAs. The performances of the three most widely used crossover operators are compared, and results subsequently presented contradict assumptions traditionally held about a GA's performance in terms of the crossover operators' disruptiveness and the nature of linkage between interacting genes.

1 Introduction

Tunable landscapes with different degress of non-linearity have long been a weapon in the armoury of those wishing to identify the characteristics which make a problem difficult for a GA to solve. The effects of epistasis on the usefulness of crossover in binary-coded GAs has been investigated in the past (for example, [3]), and perhaps most frequently used to carry out those investigations is Kauffman's NK model of fitness landscapes [5]. In this article, a way of implementing NK landscapes for real-coded GAs is presented and subsequently used to assess how the behaviours of three crossover operators are affected by the nature and the amount of epistasis to which they are subjected.

2 NK Fitness Landscapes

Kauffman's NK fitness landscapes [5] are stochastically generated fitness functions parameterized in such a way as to make them tunably rugged, thus offering the ability to assess the performance of evolutionary algorithms on problems with varying degrees of epistasis. Here N represents the length of a binary genotype while K is the number of linkages each gene has to other genes. To compute the fitness of a genotype, the fitness contribution of each locus is averaged as follows:

$$F(x) = \frac{1}{N}\sum_{i=1}^{N} F_i(x_i) \tag{1}$$

The fitness contribution, F_i, of a gene x_i is determined by using the binary allele of x_i together with values of the K interacting genes ($\{x_{i_1}, \ldots, x_{i_K}\}$ where $\{i_1, \ldots, i_K\} \subset \{1, \ldots, i-1, i+1, \ldots, N\}$) as an index into the table T_i of size 2^{K+1} of randomly generated numbers uniformly distributed over the interval [0.0, 1.0]. For a given gene i, the set of K linked genes may be randomly selected or consist of the immediately adjacent genes.

In the following section, a method of rendering continuous NK landscapes – henceforth referred to as $C(N, K)$ landscapes – is presented (the author believes) for the first time. Subsequent sections describe

experiments carried out which compare the performance on a range of $C(N,K)$ landscapes of three crossover operators (namely one-point, two-point and uniform) followed by a discussion of the results and how each operator differs in the nature of its exploration of the search space.

3 An NK Model of Fitness Landscapes for Real-Valued Strings

In the context of NK landscapes, the use of a lookup table is impracticable when generating fitness functions for real-valued, as opposed to binary, strings since the number of allelic combinations is infinite. The manner in which the fitness contribution of each locus is calculated has been modifed as follows: instead of being compared with the entries of a lookup table, the alleles of the current locus and its interactants are summed and the result multiplied by an arbitrary weight, ω_{x_i}, associated with that locus; this number is then used as the seed, t_{x_i}, for a pseudo-random number generator, Ψ. The first psuedo-random number generated using this seed is taken as the fitness contribution of the locus. (The numbers generated are uniformly distributed over the interval [0.0, 1.0].) More formally,

$$F_i(x_i) = \Psi_1(t_{x_i}) \tag{2}$$

where

$$t_{x_i} = \omega_{x_i} \left(x_i + \sum_{k=1}^{K} x_{i_k} \right) \tag{3}$$

(In this case, $0 \leq x_i \leq 100$ and $1 \leq \omega \leq 10,000$; however, these choices were arbitrary.) This method of combining the interactants' alleles to form what is likely to be a unique seed, and assigning the first number generated using that seed to a fitness contribution, retains the random nature of the assignment while ensuring that a particular allelic combination will always make the same contribution. $C(N,K)$ landscapes are therefore analagous to Kauffman's original NK landscapes.

3.1 Continuous NKp Fitness Landscapes

The NK model assumes that every combination of interactants makes a positive contribution to fitness; in nature (and real-world problems?), it seems plausible that many combinations of alleles will make no contribution to fitness [1]. This can be reflected in the NK model by assigning such a combination 0 in the fitness table. Hence, an NKp landscape is one in which each allelic combination is assigned a fitness contribution equal to 0 with probability p; the case $p = 0$ corresponds to a normal NK landscape, while $p = 1$ results in all fitness table entries being set to 0, and therefore a completely flat landscape.

The method of contructing $C(N,K)$ landscapes described above in section 3 was modified to implement $C(N,K,p)$ landscapes quite simply as follows: recall that the fitness contribution, F_i, of a locus is a pseudo-randomly generated number between 0 and 1; this number is now passed to the probability function used to decide whether or not the allelic combination in question is to make a fitness contribution. Moreover, if $p \geq F_i$ then F_i is set to 0.

3.2 Experiments

Experiments were carried out for $N = 100$ and $K = 0$, 5, 20, 50 and 80. An oft-cited shortcoming of NK landscapes is that all genes are subject to the same degree of epistasis (since traditionally K is fixed), while real-world problems vary considerably in the amount of epistasis between parameter subsets [4]. Additional experiments were therefore carried out for variable K: when constructing a landscape each locus was assigned a random number of interactants between 0 and an upper value for K; namely, $K \leq 30$, 60, 90. Random and adjacent linkage types were used for all values of K; 50 runs of 500 generations were carried out for all values of K, each randomly generating a new landscape at its outset.

The crossover operators compared were one-point (1PX), two-point (2PX), and uniform (UX) crossovers.

3.3 The Genetic Algorithm

A geographically distributed GA was used, with the population spread across a two-dimensional toroidal grid of size 15×15, each cell of which contained a single individual. Mating was restricted to small groups of individuals which were generated as follows:

	1PX			2PX			UX		
K	Best	μ	σ	Best	μ	σ	Best	μ	σ
0	0.854	0.810	0.029	0.888	0.855	0.013	0.957	0.949	0.004
5R	0.724	0.671	0.019	0.741	0.692	0.015	0.820	0.788	0.013
5A	0.772	0.748	0.013	0.795	0.771	0.012	0.808	0.781	0.013
20R	0.656	0.626	0.013	0.671	0.636	0.013	0.707	0.676	0.012
20A	0.710	0.678	0.012	0.716	0.690	0.014	0.740	0.683	0.017
50R	0.629	0.614	0.007	0.643	0.622	0.008	0.645	0.631	0.007
50A	0.655	0.636	0.009	0.667	0.639	0.012	0.656	0.635	0.010
80R	0.631	0.616	0.008	0.649	0.621	0.008	0.655	0.630	0.008
80A	0.645	0.618	0.009	0.637	0.619	0.008	0.643	0.627	0.007

Table 1: Results for fixed K (random and adjacent linkage), averaged over 50 runs of 500 generations.

- Select a grid cell at random.
- Build a neighbourhood of six individuals around the current cell by, for each neighbour, generating x- and y-distances from the current cell dependent upon a binomial approximation to a Gaussian distribution where $n = 4$ and $p = 0.85$. Thus, for individuals in cells at consecutive distances away from the current cell, the probabilities of being selected are 0.52, 0.37, 0.1 and 0.01. (The direction of the distances – up, down, left or right – are chosen at random.) Rank the neighbourhood members according to their fitness.
- Select parents according to a linear selection function favouring the fitter individuals. (After mating, the fittest child replaced a member of the neighbourhood selected according to the inverse of this function.)

No mutation was applied in order that the effects of crossover could be observed in isolation.

4 Results

Table 1 contains the results for fixed-K runs. In terms of fitness increase (illustrated in figure 1, with random linkage seen on the left, and adjacent linkage on the right), the behaviour of the operators generally conformed to past empirical studies of NK landscapes: an increase in epistasis effected a decrease in performance. Each of the three achieved its highest fitness in the absence of any epistasis ($K = 0$); all performed worse the higher the value of K, for both fixed and variable K, although relative performance was better in the variable case.

It is traditionally asserted that UX will perform well when genes are randomly linked but is too disruptive an operator to effectively manipulate adjacently linked genes (and vice versa for 1PX and 2PX); for example, in [6] the assertion is made that:

> It is essential to ensure tight linkage for GAs to work effectively. If two loci in a building block (BB) are not tighly linked, they are easily disrupted by crossover operators.

Similarly, Hordijk and Manderick claim that 'with uniform crossover... there is a large chance that a good configuration of neighbouring epistatically interacting bits will be disrupted'. Results gathered from their experiments supported their claim, illustrating that 'for high random epistasis, uniform crossover becomes too disruptive' and also that, for adjacent linkage, 'it appears that one-point crossover works better than uniform crossover... As expected, uniform crossover is too disruptive'. The results presented here directly contradict such claims: UX was unaffected by the type of linkage used – for all values of K there was no appreciable difference in performance between random and adjacent runs. (Such a difference *was* observed, however, for 1PX and 2PX: as predicted by past studies, higher fitnesses were achieved with adjacent linkage than were achieved with random linkage.) Contrary to what Hordijk and Manderick report, uniform crossover is *not* too disruptive 'for high random epistasis': as can be seen from figure 1, for high epistasis ($K = 80$) – with either random or adjacent linkage – UX performs significantly better than the other two (less disruptive) operators. Only for $K = 50$ (adjacent) did UX perform worse than

	1PX			2PX			UX		
K	Best	μ	σ	Best	μ	σ	Best	μ	σ
$\leq$ 30R	0.686	0.651	0.014	0.706	0.665	0.016	0.806	0.753	0.018
$\leq$ 30A	0.750	0.706	0.016	0.753	0.718	0.014	0.772	0.750	0.014
$\leq$ 60R	0.665	0.633	0.012	0.679	0.641	0.014	0.751	0.708	0.015
$\leq$ 60A	0.704	0.675	0.015	0.717	0.686	0.013	0.742	0.711	0.014
$\leq$ 90R	0.667	0.625	0.012	0.669	0.634	0.011	0.735	0.688	0.017
$\leq$ 90A	0.689	0.660	0.013	0.704	0.668	0.014	0.737	0.690	0.016

Table 2: Results for variable K (random and adjacent linkage), averaged over 50 runs of 500 generations.

1PX and 2PX. For runs incorporating variable K, the reproduced their patterns of performance relative between linkage types; however, UX performed significantly better than 1PX and 2PX in all cases.

5 Operators' Ergodicity

An attempt was made to assess the operators' ergodicity – their respective abilities to reach all parts of the search space with equal probability – through visualizing three attributes of crossover operators:

Distributional bias The number of genes that are exchanged under a specific crossover operator.

Positional bias The frequency with which genes are exchanged at particular loci.

Step size The Euclidean distance across the search space between parents and children. Specifically, in this case, the step size is taken to be the Euclidean distance between the child accepted into the population (i.e. the fittest) and the parent nearest to it genotypically.

5.1 Distributional Bias

Figure 2 illustrates typical distributional bias. During the first generation of a run, for UX, just over 45% of an accepted child's genes are different from the parent genotypically nearest to it (this is to be expected since $P_0 = 0.5$); for both 1PX and 2PX only 25% of genes differ. Unsurprisingly, the overall pattern of these three operators' distributional biases echo that of their convergence: the average proportion of genes whose alleles are altered during a crossover gradually decreases over the course of a run as particular alleles come to predominate in the population. For UX, this proportion takes very much longer to reach 0 than it does for 2PX; in turn, the time taken is longer for 2PX than it is for 1PX. For 1PX and 2PX the decrease is significantly faster when adjacent linkage is used; all decreases slow as K increases.

5.2 Positional Bias

Figure 3 illustrates typical positional bias at various generational intervals for each operator. Positional bias is only pronounced for 1PX: during the first generation, the two ends of a genotype have the highest probabilities – both 0.5 – of being altered; these probabilities decrease along the genotype until the likelihood of the centremost gene being altered is negligable. This hypsiloid graph softens and eventually flattens over the generations as the population converges. The lack of any positional bias in UX was mooted as the operator's main advantage [8]: all genes are equally likely to be altered, with a probability of just under 0.5 at the start of a run, and which decreases with the level of convergence. It has been shown theoretically that UX and 2PX are very different in terms of distributional and positional bias [7]; however, empirical evidence presented here suggests the opposite – that, in fact, 2PX exhibits the same (lack of) positional bias as UX. Moreover, all genes are equally likely to be altered, but with a significantly lower probability than UX (as befits it minimally disruptive nature): during the first generation, all genes have a probability of roughly 0.25 of being altered. The rate of decrease in positional bias is roughly the same for all three operators for low and medium epistasis; UX exhibits no difference in rate of decrease between random and adjacent linkage; both 1PX and 2PX exhibit faster rates of decrease for adjacent linkage than for random.

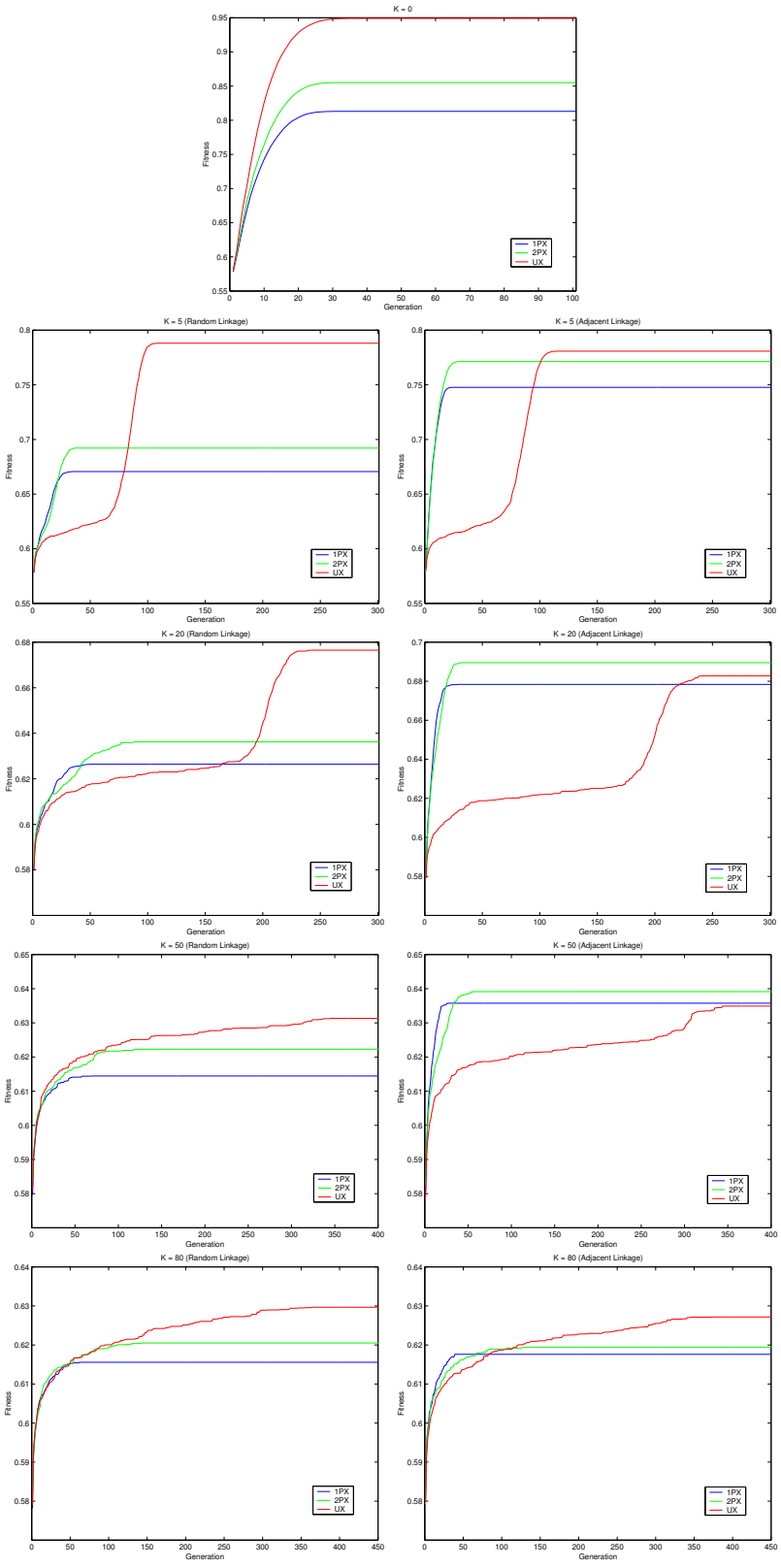

Figure 1: Fitness increase for fixed K (random linkage, left; adjacent linkage, right) averaged over 50 runs.)

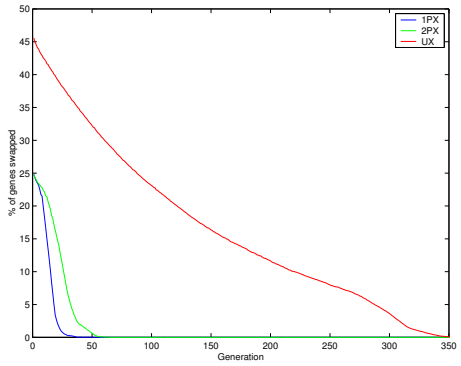

Figure 2: Comparison of operators' distributional bias averaged over 50 runs.

K	**1PX**	**2PX**	**UX**
0	19	26	52
5R	9	12	52
5A	12	16	52
20R	8	18	52
20A	8	11	54
50R	10	23	52
50A	6	12	51
80R	10	23	51
80A	10	21	53

Table 3: Number of contributing individuals to final solutions, averaged over 50 runs

Interpretation of positional bias can be misleading: as stated above, UX and 2PX exhibit the same positional bias (that is, that all genes are equally likely to be exchanged); however, this disregards the probabilities of certain genes being exchanged *together* – probabilities which, quite obviously, will differ significantly between the two operators.

5.3 Step Size

Typical examples of average step-size and their standard deviation are illustrated in figure 4. As with the two biases, the graphs of step sizes taken over the course of a run reflect the crossover operators' rates of convergence; however, it can be seen that, during the early generations, while UX takes significantly larger steps on average than either 1PX or 2PX takes, the size of its steps varies much less than those two operators. This suggests that UX effects were wider and more uniform coverage of the fitness landscape (and therefore a more efficient global search) than do 1PX and 2PX. For 1PX and 2PX, the decrease of step size (like the biases) slows as K increases, and adjacent linkage causes decreases which are faster than random linkage for equivalent values of K.

6 Discussion

The one- and two-point operators are clearly sensitive to the nature, in addition to the amount, of epistasis present in a problem, as illustrated by the contrast in behaviour between adjactent and random linkage. Patterns observed in positional bias, distributional bias and step-size are all characterized by that of convergence; therefore, it is perhaps the rate of convergence which provides an explanation for the operators' aforementioned epistatic sensitivity.

The rate of convergence for both 1PX and 2PX is slower when performance is lower, namely when linkage between interacting genes is random. Convergence – premature or not – occurs because (segments of) genes are repeatedly propogated throughout the population; the more the same genes are propogated

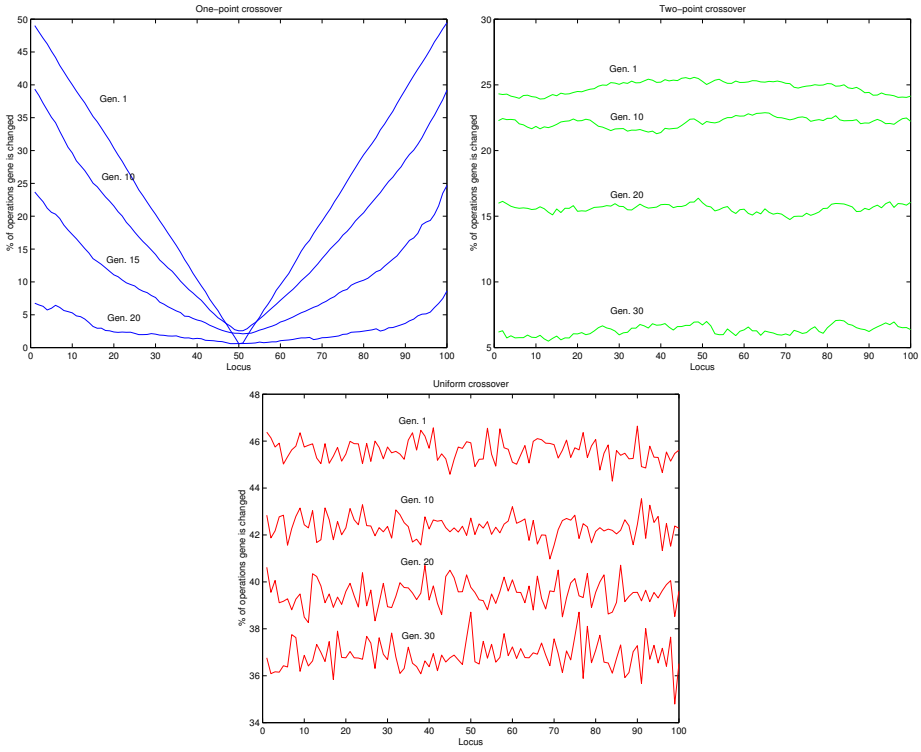

Figure 3: Operators' positional bias at various generational intervals, averaged over 50 runs.

the faster the population will converge. Intuitively, then, convergence will be slower when genes are *not* repeatedly propogated, and so, by implication, when the selection mechanism is unable to focus on distinctly fit individuals. Thus two obvious instances in which convergence will be slowed can be identified: when fitness variance among population members is low (for example, when plateaux are to be found on the fitness landscape – hence, in the context of NKp landscapes, when p is high); and when there exists a wide fitness disparity between two or more instances of a particular collection of genes (that is, when genes previously associated with high fitness suddenly become associated with low fitness – the occurrence of 'lethals', or unfit children resulting from fit parents). There is some evidence to suggest that it is the latter situation which is arising: first, the neighbourhood fitness variances of adjacent-linkage and random-linkage runs remained identical during the early generations – contrary to the former instance. It seems reasonable to assume that in the event of certain genes being simultaneously associated with high and low fitness, a greater number of destructive crossover operations would occur: indeed, higher levels of destruction (in terms of amount and duration) are observed in all random-linkage runs for low- to mid-K (as illustrated in figure 5); higher levels of destruction are also observed for high-K but are not so pronounced. (As K increases, the manner in which fitness on NK landscapes is calculated will gradually reduce the distinction between random and adjacent linkage: a very large number of interactants, even if assigned randomly, will inevitably be adjacent.)

The accepted explanation for such operators as 1PX and 2PX performing worse with random linkage is that schemata are more likely to be disrupted than they are when adjacent linkage is used. While this is probably true, it is perhaps more accurate (with regard to real-valued GAs, at least), rather than pronounce those operators too disruptive in certain situations, to accuse them of being insufficiently explorative in most situations – an attribute which might explain their behaviour: their failure with random linkage, their comparative success with adjacent linkage, and their generally worse performance compared with UX in either case. (Table 3 simply illustrates each operator's exploratiVeness by recording the average number of population members which contributed genes to the best solution at the end of a run.) When $K = 0$, no destructive operations occurred and, as expected, fitness increase was fastest and highest for all operators; that is, when all genes are independent (and when building block hypothesis [2] achieves the height of its feasiblility). The pattern of fitness increase evident when $K = 0$ is also exhibited by 1PX and 2PX for higher values of K, but only during adjacent-linkage runs; fitness increase graphs

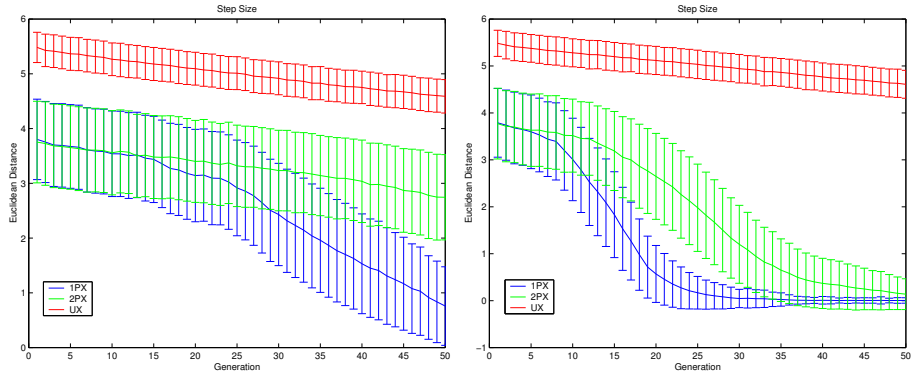

Figure 4: Average sizes of step taken through the search space by the compared operators in one application (with standard deviation), averaged over 50 runs (adjacent linkage, left; random linkage, right).

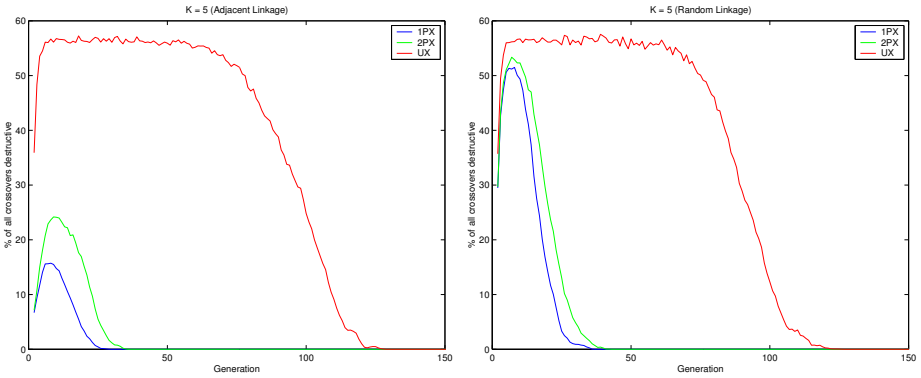

Figure 5: An example of the effect of linkage type on the quantity of destructive crossovers, averaged over 50 runs (adjacent linkage, left; random linkage, right)

for equivalent random-linkage runs are shallower, at times vaguely sinusoidal in shape. It is feasible, then, that when linkage is adjacent, rather than exploring the landscape *per se*, 1PX and 2PX are simply recombining relatively fit building blocks – fortuitously assembled segments of genes – which were present since the population was generated; moreover, randomly generated fit sequences of interactants are spliced together without those sequences themselves being explored for different, potentially even more promising, alleles and combinations thereof. (This behaviour would be most evident when the number of K-schemata – by which is meant a set of K interacting genes – disrupted is minimized: when K is smaller than, or equal to, the amount of a genotype altered during crossover (which, recalling results of measuring distributional bias in section 5.1, is around 25% when the population is at its most diverse). Conversely, when linkage is random, they are indeed disruptive but only of those (comparatively fit) K-schemata which were to have been found in the population from its generation (rather than constructed by recombination); thus, they are effectively forced into exploration but, owing to their minimallly disruptive nature, are unable to explore widely or efficiently enough to discover promising areas of the landscape before premature convergence takes hold.

The pattern of fitness increase observed for UX suggests that during the early stages of a run, while new combinations are continually created, random search is in operation (though only in the context of NK landscapes); that is, crossover is maximally explorative until convergence eventually begins to force the operator to piece together the fittest building blocks it has hitherto been constructing (as opposed to merely finding *à la* 1PX and 2PX). At this point a further significant fitness increase is observed.

Convergence was measured as the average difference between a population member and the best solution found so far; therefore, if a graph reaches 0, the population has converged on the fittest solution found. Figure 6 illustrates the average rate and level of convergence of 50 runs and its pattern is typical of those observed in all experiments regardless of K; a comparison is made between runs using adjacent

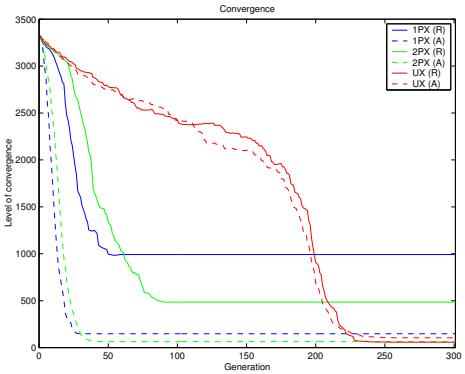

Figure 6: Example of the effect of linkage type on convergence, averaged over 50 runs.

and random linkage for an equivalent value of K. It can be seen that during a random-linkage run the population will *not* converge to the fittest solution found under 1PX and 2PX; that the population is drawn away from the fittest solution by 2PX and to an even greater extent by 1PX suggests that the positional bias – or rather, the relative postional bias, as proposed in section 5.2 – is the cause of this behaviour.

For the solution to which the population converges to share only a few, if any, genes with the fittest solution found, highly fit K-schemata must be broken by recombination into constituent parts which are considerably less fit than neighbouring alternatives to those constituent parts or the K-schemata which they form. Recall that the observed positional bias for 1PX is towards a genotype's extremities; it was suggested above that when linkage is adjacent, 1PX (and 2PX) merely recombine fit K-schemata already present in the population rather than constructing new ones: when linkage is random it appears that 1PX will attempt to behave in the same way. Hence, fit K-schemata of which interactants are positioned at either end of a genotype will be disrupted and eventually lost as 1PX draws the population towards less, but comparatively, fit individuals whose fit K-schemata are predominantly or entirely positioned around the centre of the genotype and are therefore less likely to be disrupted. Since 2PX has a relative positional bias similar to that of 1PX it will likewise attempt to direct the population, but the effect will be diluted by the absence of any observed positional bias: overall, all genes have the same probability of being altered so even if a K-schemata is disrupted there is a greater chance that it will be reconstructed in the future. UX imposes no bias whatsoever, is therefore immune to the nature of linkage and will not direct the population in any direction other than of the fittest individual.

7 Conclusion

$C(N,K)$ landscapes, a continuous-variable implementation of NK landscapes, has been presented; an oft-quoted shortcoming of the original implementation of NK landscapes – that, unlike real-world problems, the amount of epistasis between parameter subsets is fixed [4] – was rectified by carrying out experiments which initialized landscapes with variable K. The subsequent investigation of epistasis in real-valued genetic algorithms produced results which reflect those published with regard to epistasis in binary GAs; moreover, a GA will perform less well in the face of increased epistasis. However, it has also served to augment results presented in previous chapters which suggest that, contrary to traditional opinion, highly disruptive – or rather, explorative – crossover operators contribute to a more successful search of continuous fitness landscapes. The uniform crossover operator produced the best results in all but a few instances; it performed particularly well when subjected to very high epistasis, and with variable-K landscapes (which may be more analagous to real-world problems than are fixed-K landscapes). The highly disruptive nature of UX which, in the realm of binary-coded GAs, is customarily frowned upon (and, indeed, restricted by setting a high value for p) seems to have proved entirely beneficial to the performance of a real-coded GA: the lack of any positional bias in UX renders it maximally explorative, immune to the nature of linkage between interacting genes, and causes population diversity to be maintained for longer.

References

[1] L. Barnett. Ruggedness and neutrality – the *nkp* family of fitness landscapes. In C. Adami, R. K. Belew, H. Kitano, and C. E. Taylor, editors, *Proceedings of the Sixth International Conference on Artificial Life*. MIT Press, 1998.

[2] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[3] W. Hordijk and B. Manderick. The usefulness of recombination. In F. Moran, A. Moreno, J.J. Merelo, and P. Chacon, editors, *Advances in Artifical Life: Proceedings of the Third European Conference on Artificial Life*. Springer-Verlag, 1995.

[4] K. A. De Jong, M. A. Potter, and W. M. Spears. Using problem generators to explore the effects of epistasis. In T. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*. Morgan Kaufmann, 1997.

[5] S. A. Kauffman. Adaptation on rugged fitness landscapes. *Lectures in the Sciences of Complexity*, 1, 1989.

[6] M. Munetomo and D. E. Goldberg. Designing a genetic algorithm using linkage identification by nonlinearity check. Technical Report 98014, Illinois Genetic Algorithms Laboratory, 1998.

[7] W. M. Spears. Crossover or mutation? In D. Whitley, editor, *Foundations of Genetic Algorithms 2*. Morgan Kaufmann, 1992.

[8] G. Syswerda. Uniform crossover in genetic algorithms. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann, 1989.