Biases Introduced by Adaptive Recombination Operators

Kanta Vekaria and **Chris Clack**
Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK
{K.Vekaria, C.Clack}@cs.ucl.ac.uk

Abstract

The broad goal of adaptive techniques is to acquire knowledge dynamically about the search space and to use this knowledge to bias the evolutionary process. The effectiveness of any adaptive technique is therefore determined by the biases being used. In this paper we identify four key biases introduced by adaptive recombination operators and analyse the relationship between these biases. We use these four biases (*directional, credit, initialisation* and *hitchhiker*) to characterise three adaptive recombination operators. We show that the biases introduced by adaptive recombination are not always beneficial to GA performance and we explore methods for minimising the detrimental effects.

1. INTRODUCTION

Adaptation in genetic algorithms (GAs) has become a key strategy to enhance GA capabilities. The broad goal of adaptive techniques is to acquire knowledge dynamically about the search space and to adjust the GA appropriately. One or many parameters can be adapted. For example, adapting crossover or mutation probabilities (Davis 1989), adapting representations of individuals (Goldberg, Korb & Deb 1989) and adapting recombination operators (this is further discussed in the next section). The biases that are used in an adaptive technique to generate new individuals determine their effectiveness; bias is a mechanism used to push search towards particular regions in the search space. The general bias of a GA is implemented by selection according to fitness. Some GAs incorporate biases that are specific to a particular problem. In this paper we look at biases on alleles used to direct search by focusing on adaptive recombination operators. We show that the biases introduced by adaptive techniques are not always beneficial to GA performance but can hinder the performance of a GA. This paper identifies four key biases introduced by adaptive recombination: Directional Bias, Credit Bias, Initialisation Bias and Hitchhiker Bias. We describe the four biases drawing on three adaptive recombination operators that use local fitness information to create future offspring. We analyse how these biases interact with each other. We then show, with empirical results, the effects of the initialisation, credit and hitchhiker biases on a GA using selective crossover (Vekaria & Clack 1998). This is done by eliminating the initialisation bias and reducing the credit bias.

The paper is organised as follows: the next section provides a brief overview of adaptive recombination operators and goes on to describe in more detail three operators that use fitness information to guide the GA. Section 3 identifies and describes four biases imposed by these adaptive operators. This is followed by an analysis of the interaction amongst these four biases. Section 4 discusses potential methods to eliminate the initialisation bias and to reduce the credit bias; it goes on to show empirically how this elimination or reduction of biases affects GA performance. Section 5 and 6 critically analyse the results and conclude respectively.

2. ADAPTIVE RECOMBINATION

Recombination, also known as crossover, is a commonly used operator in a GA. Traditionally, recombination has been considered as the primary operator of a GA and thought to be responsible for the generation and propagation of solutions. More recently, there have been many studies on the role played by traditional crossover operators, compared with mutation, in a GA (Schaffer & Eshelman 1991, Spears 1993 and Wu, Lindsay & Riolo 1997). Crossover operators have also been classified on their usefulness in terms of generating and propagating solutions (Eshelman & Schaffer 1995). There are now many different ways of implementing recombination (Spears 1997). Some forms of recombination are more suitable for certain problems than others and some proposed for general problems incorporate adaptive methods and thus are classed as adaptive recombination operators. Examples of adaptive recombination follow in chronological order:

Schaffer and Morishima (1987) proposed Punctuated Crossover that evolved the positions at which crossover was allowed to occur. They accomplished this by appending a crossover bitmap to the end of the encoded solution. The extra bitmap underwent the same crossover as the corresponding alleles thereby evolving the crossover positions. Inferior children were discarded along with their crossover bitmaps.

Louis and Rawlins (1991) proposed Masked Crossover, which uses binary masks to direct crossover. Relative fitness information is translated into the binary mask, thus guiding crossover towards local fitness increases.

White and Oppacher (1994) proposed Adaptive Uniform Crossover where each bit string in the population is augmented at each bit position with an automaton. Each automaton state maps to a crossover probability for that bit string location. Their operator also uses fitness information to identify groups of bits to be kept together when crossover occurs.

Spears (1995) proposed a 1-bit adaptation that allowed the GA to choose between uniform and two-point crossover while solving the problem. A single bit is appended to each individual; this bit determines which operator should be used for crossover. If both parents sample a 1 then two-point crossover is used; if both sample a 0, uniform crossover is used; otherwise either is used with 50% probability.

Vekaria and Clack (1998) proposed Selective Crossover that biases alleles that are known to have increased an individual's fitness. It achieves this by attaching a real valued vector to an individual to accumulate fitness information from recombination in previous generations. It uses this information to preserve known fit alleles.

Goldberg, Korb and Deb (1989), Harik (1997) and Smith (1998) have proposed other adaptive techniques that learn linkage. For a lengthy survey on recombination the reader is referred to Spears (1997). Masked Crossover, Adaptive Uniform Crossover and Selective Crossover are further discussed in Section 2.1, 2.2 and 2.3 respectively. All three adaptive recombination operators use fitness information to bias search in a GA.

2.1 MASKED CROSSOVER

Masked Crossover uses relative fitness information to guide the crossover operator. The relative fitness of the children, with respect to their parents' indicates the desirability of moving in that direction along the search space. To do this each chromosome has an associated bit vector, (a binary mask). These binary masks are the primary element that dictates crossover. On initialisation the binary masks are generated randomly. Consider two parents Parent1 and Parent2 and their associative binary mask vector M1 and M2. A recombination event will create two children Child1 and Child2, which are initially direct copies of Parent1 and Parent2 respectively. The binary masks are then compared linearly across the chromosome and the solution vectors are updated as shown in Figure 1 and as defined below:

```
for i=0 to ChromosomeLength
 if M1i == 1 and M2i == 0
   copy i bit from Parent1 to Child1 and Child2
 else if M1i == 0 and M2i == 1
   copy i bit from Parent2 to Child1 and Child2
 else
   copy i bit from Parent1 to Child1
   copy i bit from Parent2 to Child2
```

The inheritance of the mask is not done in the same way: instead they use rules for mask propagation. Figure 2 gives an example of how binary masks get inherited by the children. Children are categorised into three types; Good (fitter than best parent), Average (fitness within range of parents) and Bad (less or equally fit than worst parent). With two children produced by each crossover and three types of children, six rules were devised for the corresponding six ways of pairing the children: both good, both bad, both average, average/bad, average/good, or good/bad. The rules may modify both the masks of the children and the masks of the parents. For example, if both children are good then the masks of the children are given by ORing the masks of the parents and the parents' masks are left unchanged (see Figure 2). By contrast, if both children are bad then the childrens' masks are set to zero in those positions where a bit has been inherited from the other parent and the parents' masks are also changed (see Figure 3). Louis does not use generational replacement; instead he uses a steady state population. This means that parents will still remain in the population after crossover and if their masks produced inferior children then it is likely to do so again; therefore the binary masks of both parents also undergo a change to reflect detrimental effects.

Parent1 / Parent2

1	0	1	1	0	1	Binary Masks	0	0	1	0	1	0
0	1	1	1	1	1	Alleles	0	0	0	0	0	0

Create children

Child1 / Child2

0	1	1	1	0	1		0	0	0	1	0	1

Figure 1: Masked Crossover

Child1 / Child2

1	?	1	1	1	1	CM1 CM2	1	?	1	1	1	1
0	1	1	1	0	1	Alleles	0	0	0	1	0	1

Figure 2: Creation of new binary masks if both children are good. (? denotes randomly generated).

Child1 / Child2

1	0	?	1	0	1	CM1 CM2	0	0	1	0	1	0
0	1	1	1	0	1	Alleles	0	0	0	1	0	1

Figure 3: Creation of new masks if both children are bad. Assume Parent2 > Parent1 in fitness. (? denotes randomly generated).

When activating these rules, masked crossover does not take into account population homogeneity (when both parents have matching alleles at loci *i*). The binary mask is updated on the basis of which parent contributed to create a good or bad child and not what differing bits were introduced to the child. For example, in Figure 1 an exchange occurring at locus 1 does not change the bit value at that locus; hence if there was a fitness increase or decrease it would not be the result of the exchange at locus i. Since masked crossover uses a binary mask there is only a binary relationship between an allele that is considered fit and one that is not. This means that an allele that contributes little to the fitness increase is considered equally to an allele that contributes more to the fitness increase. The changes in the binary mask do not reflect the magnitude of the fitness increase and thus allow very little competition amongst the alleles in the population. Masked crossover penalises alleles when there is a fitness decrease but due to interactions amongst genes it is difficult to determine exactly which allele was the cause of the fitness decrease; hence, alleles that are not directly related to the fitness decrease get penalised too.

2.2 ADAPTIVE UNIFORM CROSSOVER

Adaptive uniform crossover (AUX) uses discretised linear automatons with N+1 states where N is an even integer. The set of states $S = \{s_0, s_1, \ldots, s_N\}$. Associated with each state is a probability of choosing an action. For each individual of length l there are l automatons associated with it, see Figure 4. The * in each automaton indicates the current state of the automaton. Each state maps to the probability of crossover (bit exchange) at that location.

When bits are exchanged their associated automatons are also passed onto the child. States are changed as a result from the feedback of the environment and the rules that have been invoked. Examples of rules that can be invoked are given in Table 1.

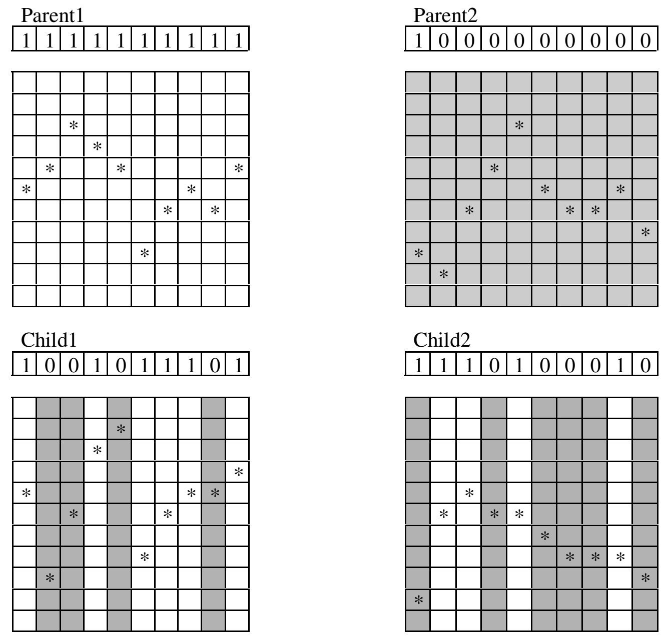

Figure 4: Adaptive Uniform Crossover. Fitness of Child1 > Parent1, Parent2. Fitness of Child2 < Parent1, Parent2

Table 1: Rules for Adaptive Uniform Crossover

Offspring fitness	Reward bits from Parent1 with prob.	Reward bits from Parent2 with prob.
>Parent1	$+P_{superiorReward}$	0
>Parent2	0	$+P_{superiorReward}$
<Parent1	$-P_{inferiorPenalty}$	0
<Parent2	0	$-P_{inferiorPenalty}$
=Parent1	$+P_{sameReward}$ or $-P_{samePenalty}$	0
=Parent2	0	$+P_{sameReward}$ or $-P_{samePenalty}$

A reward implies that the automaton moves from state i to i+1. A penalty implies that the automaton moves from state i to i-1. If the automaton was in state N prior to a reward then no change is applied to the automaton. This similarly applies when the automaton is in state 0 prior to a penalty. The initial states of the automatons were set randomly. In Figure 4, the fitness of Child1 is greater than both its parents'; therefore, according to the rules the states of all automatons get updated by moving to i+1, if the current state is i. The fitness of Child2 is less than both parents; therefore the states of all automatons get updated by moving to i–1, if the current state is i.

AUX uses a finite automaton; therefore the automaton cannot move to another state if the final state is reached and if there is a fitness increase. The change in states of the automaton do not reflect the magnitude of the fitness increase, hence the automaton changes state if there is a small or large increase in fitness thereby causing stagnation. AUX does not take into account population homogeneity. The algorithm only rewards or penalises automatons on the basis of which parent contributed to which child. This has a disadvantage; if both parents have identical bits at locus 1 and 6 it does not make any difference which parent contributed those bits, since the child was going to inherit them anyway.

2.3 SELECTIVE CROSSOVER

In selective crossover each individual has associated with it a real-valued vector, and thus each allele has an associated dominance value. Recombination uses two parents to create two children. During recombination two parents are selected and their fitness is recorded. The dominance value of each allele in both parents is compared linearly across the chromosome. The allele that has a higher dominance value contributes to Child1 along with the dominance value. If both dominance values are equal then crossover does not occur at that position. Figure 5 gives an example of selective crossover: the shaded alleles have a higher dominance value than its competing allele. To keep diversity in the population Child2 inherits the non-dominant alleles.

After crossover the two new children are evaluated. If a single child's fitness is greater than the fitness of either parent, the dominance values (of those alleles that were exchanged during crossover) are increased proportionately to the fitness increase. This is done to reflect the alleles' contribution to the fitness increase. Figure 6 gives an example of the mechanism. It follows on from the selective crossover example given in Figure 5. In Figure 6, only Child1 has an increase in fitness of 0.1 (compared with the fittest parent) hence its dominance values get updated. In Figure 5 the bit values of Parent1 and Parent2 at loci 1 and 2 did not get exchanged during crossover and the bit values at loci 4 are the same in both parents; this also applies to loci 6. Thus, after selective crossover, the alleles that caused a change in the chromosome are only those held at loci 3 and 5. Since the change of those alleles at loci 3 and 5 resulted in an increase in fitness, only their dominance values get increased in Child1 (shaded in Figure 6).

Parent1 – fitness = 0.36

0.4	0.3	0.01	0.9	0.1	0.2	Dominance values
1	0	0	1	0	0	Alleles

Parent2 – fitness = 0.30

0.01	0.2	0.4	0.2	0.9	0.3
0	1	1	1	1	0

Child1 – fitness = 0.46

0.4	0.3	0.4	0.9	0.9	0.3
1	0	1	1	1	0

Child2 – fitness = 0.20

0.01	0.2	0.01	0.2	0.1	0.2
0	1	0	1	0	0

Figure 5: Recombination with Selective Crossover

Child1 – fitness = 0.46

0.4	0.3	0.5	0.9	1.0	0.3	New dominance values
1	0	1	1	1	0	

Child2 – fitness = 0.20

0.01	0.2	0.01	0.2	0.1	0.2
0	1	0	1	0	0

Figure 6: Updating Dominance Values

3. BIASES

Search procedures of a GA make use of biases to help direct the search. A bias is a mechanism used to push search towards particular regions in the search space. The general bias of a GA is implemented by selection according to fitness. In this paper we look at biases on alleles used in adaptive recombination to direct GA search. These biases affect the performance of a GA; in some cases it can hinder the search capabilities.

Eshelman, Caruana and Schaffer (1989) described two recombination biases apparent in traditional non-adaptive crossover operators: Positional Bias and Distributional Bias.

Positional bias exists when the creation of a new individual is dependent upon the location of the alleles in the chromosome. Booker (1992) showed that, of the *n*-point recombination operators, one-point crossover has the highest positional bias. Booker showed that for $n < L/2$ (where n is the number of crossover points and L is the length of the chromosome) the positional bias tends to decrease as n increases for *n*-point recombination. Uniform crossover or Uniform Parameterized crossover (Spears 1998) has no positional bias.

Distributional bias exists if the amount of material being exchanged is concentrated toward a mean value. If the distribution of the alleles being exchanged is uniform (ranging from 0-(L-1)), there is no bias. The more the distribution differs from the uniform distribution the higher the distributional bias. Booker found that the distributional bias of *n*-point recombination tends to increase as n increases, as the distribution becomes less and less uniform. Just as population homogeneity affects the explorative power of a recombination operator, it also affects the distributional bias. Spears (1998) extended the work by Eshelman *et al* to include population homogeneity. He concluded, and confirmed the results of Booker and Eshelman *et al*, that one-point and two-point crossover do not have distributional bias, whereas uniform (parameterized) crossover has high distributional bias. The bias increases as P_0 decreases from 0.5 to 0.0.

Eshelman, Caruana and Schaffer (1989) showed that crossover operators that have high distributional bias (uniform crossover) outperformed those that had high positional bias (one-point crossover). However, their study was limited to a small set of problems.

As new adaptive recombination techniques are developed new biases are introduced. We extend the study by Eshelman, Caruana and Schaffer (1989) and identify four key biases introduced by adaptive recombination: directional bias, credit bias, initialisation bias, and hitchhiker bias. We analyse the relationship amongst these biases and characterise masked crossover, adaptive uniform crossover and selective crossover in terms of the four new biases by analysing the specific methods used (for selecting bits to be exchanged) by these adaptive operators.

3.1 DIRECTIONAL BIAS

Directional bias exists if alleles are favoured (or not favoured) for their credibility. This bias determines the direction the GA is likely to converge to. Directional bias benefits GA search as it pushes the GA towards fitter regions in the search space.

All three operators have directional bias as they favour those alleles exchanged during crossover that show a fitness increase relative to their parents. In the case for masked crossover and AUX they also penalise alleles that show a fitness decrease. This form of bias means that the operators follow contours on the search space where fitness increases; therefore, the GA can potentially be misled and is unable to solve deceptive problems. Vekaria & Clack (1998) show that selective crossover like two-point and uniform crossover was unable to solve trap functions that are deceptive (order 20 and above).

3.2 CREDIT BIAS

The credit bias is the degree at which an allele gets favoured with respect to its credibility. Credit bias is used as a means for accumulating fitness information on each generation and using the knowledge, during recombination, to exploit alleles. The amount of credit bias decides the amount of allele exploitation. Too much exploitation will limit the exploration and too little exploitation can slow convergence. A high credit bias means more exploitation and hence reduced exploration. A recombination operator has high credit bias if alleles are given maximum credit regardless of their fitness contribution.

Masked crossover has high credit bias. It uses a binary mask. This means that there is only a binary relationship between an allele that is considered good and one that does not contribute to the fitness and is considered bad

(Section 2.1). This allows very little competition between the alleles.

Masked crossover and AUX change the binary mask and automaton state respectively when there is a fitness increase, but the changes incorporated do not reflect the magnitude of the fitness increase. For example, the same changes occur to the binary mask and automaton when there is a fitness increase of 5 and 10. AUX uses a finite automaton with probabilities; this means that once the automaton reaches the final state it cannot move to another state if there is a fitness increase; and hence causes stagnation (reduced directional bias). AUX also has high credit bias.

In the case for selective crossover, different variations in the fitness increase are captured as the fitness increase determines the increase of the dominance values (see Figure 6). Selective crossover uses real values; there is no restriction on how big these values can get, other than computational restrictions. These continuous values allow more competition amongst the alleles and therefore provide more explorative power in early generations. Selective crossover has less credit bias than masked crossover and AUX.

3.3 INITIALISATION BIAS

A recombination operator has an initialisation bias if alleles are favoured, during initialisation, without knowing their credibility. This means that alleles are exploited without any knowledge if their contribution to fitness. This form of exploitation occurs prior to any exploration and therefore restricts the GAs explorative power. Given a *"large enough"* population this bias will not exist because the distribution is evenly spread and therefore averaged out. As ideal population sizes are unknown for different problems, this bias can potentially affect GA performance.

In selective crossover the dominance vector is randomly assigned with real values and with a constraint that they must be in the range [0,1] inclusive. This implies that on the first few generations the GA has a potential of being misled. For example, consider the One Max problem where the aim is to have all 1's in the chromosome. Now during initialisation 50% of the population is highly likely to contain 0's, as the population is randomly generated. Since the dominance values are randomly assigned it is likely that a proportion of the population contains 0's that are more favourable than the 1's. The distribution of dominance values can be skewed towards a desirable start or an undesirable start: the distribution is desirable when the population contains a large percentage of 1's with dominance values > 0.5; the distribution is undesirable when the population contains a large percentage of 0's with dominance values > 0.5. Therefore on the first recombination process of the GA, this uncertainty of whether a 0 gets favoured more than the 1 during initialisation gives us less confidence in GA behaviour. In the case for adaptive operators we hope for guarantees on behaviour.

Masked crossover and AUX have initialisation bias: in masked crossover, the binary mask vector is randomly assigned with bit values and a 1 favours a allele whilst a 0 does not: similarly, the states of the AUX automaton are set randomly.

3.4 HITCHHIKER BIAS

Hitchhiker bias exists if alleles get favoured when they are not the cause of a fitness increase.

Selective crossover increases the dominance values of only those alleles that were exchanged during recombination and resulted in a fitness increase in the child. The increase of the dominance values is determined by the fitness increase relative to the parents. In the case of the One Max problem where schemas containing 1's are fitter than those containing 0's, if a '0' is introduced in a child as well as three 1's the fitness will increase and so will the dominance values. The dominance values, of the three 1's and the '0' get increased by the same amount (the fitness increase). Hence in following generations the '0' will be regarded as dominant and get passed down to future generations. For the One Max problem such an event is not desirable. For example in Figure 7, four alleles were exchanged to create Child1 and Child2. Given the one-max problem, if Child1 has a fitness increase of 3; the '0' that was also exchanged at loci 2 will get favoured. The dominance values of all four alleles (shaded in Figure 7) will increase by 3. This means that the '0' at loci 2 is hitchhiking; it did not contribute to the fitness increase. *For this reason, we view the hitchhiker bias as being detrimental to the evolutionary process.*

Masked crossover and AUX has more hitchhiker bias than selective crossover because neither check for population homogeneity, hence favouring alleles which are not the result of a fitness increase relative to the parents, see Figures 2 and 4. Also both operators penalises alleles that were exchanged and caused a fitness decrease but this has a potential of making error as those alleles that are not the cause of the fitness decrease get penalised too.

Child1 – fitness = 4

1.2	0.4	0.8	3.0	1.0	0.3
1	0	1	1	1	0

Child2 – fitness = 2

0.11	0.3	0.01	1.8	0.1	0.2
0	1	0	1	0	0

Figure 7: Hitchhiker bias

Due to the continuous nature of the dominance values in selective crossover there is increased competition amongst the alleles thereby reducing the effect of the hitchhiker bias. For example, Figure 7 shows only a single recombination event whereas other individuals may contain a '1' at loci 2 with a higher dominance value; thus, selective crossover has a better chance of discarding hitchhikers in the population.

3.5 ANALYSIS AND SUMMARY

The directional, credit and hitchhiker biases are not independent biases and each one is a direct result of another. Figure 8 shows the relationship amongst the four identical biases. Directional bias is a high level bias and can be considered as the general direction required by the GA. Credit bias falls a level below the directional bias and is a direct result of it. Having decided the direction, the credit bias represents *"how"* to follow the direction by means of assigning credit to alleles. Credit bias can be assigned in many ways as seen by the adaptive recombination examples given in this paper. The credit bias in selective crossover is determined by the fitness; the dominance values get updated by the fitness increase (another variation of the credit bias is to increase the dominance values by sharing the fitness increase amongst the changed alleles). The hitchhiker bias is a direct result of the credit bias. Without a credit bias on the alleles there is no hitchhiker bias on alleles. From Figure 8 we can see how directional, credit and the hitchhiker bias can be grouped together. The initialisation bias does not fall within this group as it is introduced independently into the adaptive technique. The initialisation bias is also a cause of the hitchhiker bias because alleles are being assigned a credit when they may not be fit alleles; therefore the initialisation bias can be harmful to GA search.

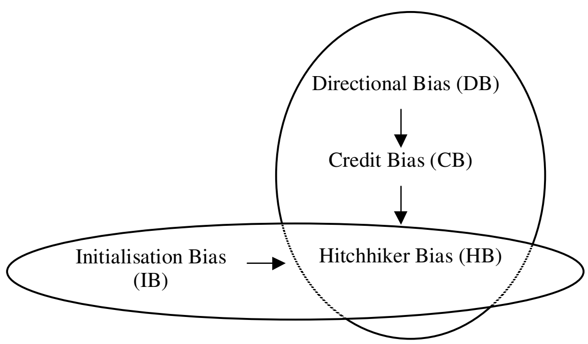

Figure 8: Relationships between biases

Table 2 provides a summary of the biases imposed by the respective operators. Uniform crossover is also shown in Table 2 to show that it does not possess these biases.

Table 2: Strength of biases present in recombination operators.

	IB	DB	CB	HB
Uniform	None	None	None	None
Selective	High	High	Low	Medium
Masked	High	Medium	High	High
Adaptive uniform	High	Medium	High	High

4. ELIMINATING OR REDUCING BIASES

In this section we show, using selective crossover as an example, how the initialisation bias and the credit bias affect GA performance. We focus only on the initialisation and credit bias because the directional bias is a key characteristic of selective crossover and cannot be eliminated. The hitchhiker bias may potentially get reduced, as explained in Section 4.2, by reducing the credit bias. To see how the initialisation and credit bias affects GA performance it is eliminated or reduced respectively. The results are compared against the original algorithm. Experiments were carried out with no mutation to analyse the recombination operator on its own merits. The problems used were Royal Road (R2) (Forrest & Mitchell (1993)) and NK Landscapes (Kauffman (1989)).

4.1 ELIMINATING THE INITIALISATION BIAS

In the previous section we identified that the initialisation bias causes a hitchhiker bias; and thus may be harmful for GA search. To determine whether the initialisation bias is harmful and (if so) to eliminate this form of bias, a new method where dominance values are initialised at zero was compared with the original selective crossover.

In this new method of initialisation, where dominance values all start with zero, the first recombination process cannot be carried out as described in Figure 5 as all dominance values are equal. We therefore use uniform crossover, where $P_0 = 0.5$, and then increase the dominance values according to the fitness increase presented by the children. The steps to avoid this form of bias during initialisation are:

1. Initialise all dominance values as 0.0.
2. First, and only first, process of recombination is done with uniform crossover where P_0 is 0.5.
3. Increase dominance values as shown in Figure 6.
4. All successive recombination processes are done as shown in Figure 5.
5. All other processes of a GA remain the same.

As a reminder, during a recombination event if two parents have an equal dominance value at the same loci the bits do not get exchanged. The results are shown in Table 3 and discussed in Section 4.3.

4.2 REDUCING THE CREDIT BIAS[1]

To determine whether the credit bias is harmful we reduce the magnitude of the bias by increasing the dominance values by only a fraction of the fitness increase. For example, if five alleles were exchanged during crossover and the fitness increase was 3 then each dominance value belonging to these alleles will get increased by 0.6 and *not* 3. The fitness increase is shared between the changed alleles, as it is difficult to tell which allele actually contributed to the fitness increase (Figure 9). By using this method of reducing the credit bias we may be able to reduce the hitchhiker bias. In this method if a single allele gets exchanged it will get complete recognition for its fitness contribution. On the other hand if two alleles were exchanged it is difficult to know which allele is hitchhiking; therefore, by reducing the credit bias we increase competition amongst the alleles.

The fitness increase of Child 1 with respect to its parents is 0.1 (see Figure 5). As two alleles were changed we do not know which allele influenced the fitness, therefore we share the fitness increase amongst all the alleles that were changed. In this example, the dominance values get increased by 0.05. By doing this, if a single allele was

[1] This experiment was carried out independent of eliminating the initialisation bias. This means the initialisation bias still existed in this experiment.

exchanged and fitness increased then that allele will get full recognition; its dominance value will get increased by the complete fitness increase. In this method we are only decreasing the magnitude of the credit bias we give to the alleles (the amount the dominance values get increased by). By reducing the credit bias we may still be able to aid convergence by reducing the hitchhiker bias. The processes for this method are:

1. Initialise all dominance values randomly.
2. Apply selective crossover as shown in Figure 5.
3. Increase dominance values as shown in Figure 9
4. All successive recombination processes are done as shown in Figure 5.
5. All other processes of a GA remain the same.

Child1 – fitness = 0.46

0.4	0.3	0.45	0.9	0.95	0.3
1	0	1	1	1	0

New dominance values

Child2 – fitness = 0.20

0.01	0.2	0.01	0.2	0.1	0.2
0	1	0	1	0	0

Figure 9: Reducing the magnitude of the credit bias

4.3 RESULTS

Experiments were carried out with no mutation to analyse the recombination operator on its own merits. Problems used were Royal Road (R2) (Forrest & Mitchell 1993) and NK Landscapes (Kauffman 1989). The choices of these two problems are due to the tight building blocks inherent in Royal Roads and the epistasis and difficulty associated with NK landscapes problems. Table 3 shows the results of eliminating and reducing the initialisation and credit biases respectively. It also shows the results of uniform crossover, which does not contain any of the above named biases.

Table 3: The effect of the initialisation and credit biases on GA search. The standard deviation is shown in parentheses.

	Royal Road R2	NK Landscapes	
		K=31	K=8
Uniform Crossover	44.00 (10.07)	0.7396 (0.0081)	0.7572 (0.0178)
Original selective crossover	14.93 (7.65)	0.7214 (0.0141)	0.7142 (0.0241)
No initialisation bias	19.20 (8.42)	0.7218 (0.0160)	0.7230 (0.0241)
Reduced credit bias	12.35 (6.4)	0.7060 (0.0127)	0.7074 (0.0229)

The GA was allowed to run until the population had completely converged. The length of the chromosome used for the Royal Road function was 64 and the population size was 128. The length of the chromosome used for the NK landscapes was 32 and population size was 200. In Table 3, "Original selective crossover" refers to the algorithm described in Section 2.3. The numbers in Table 3 show the average fitness of the best solutions found; these are averages taken from 50 independent runs. The maximum achievable fitness for the Royal Road and the NK landscape are 192 and 1.0 respectively.

5. ANALYSIS

From the results we can see that the initialisation bias does have a harmful affect on the GA. The GA was able to find a better solution than the original selective crossover for the entirely uncorrelated landscape (k=31) and also when k=8 (0.7218 and 0.7230 compared with 0.7214 and 0.7142). The GA was also able to find better solutions for the Royal Road function. These results show that by initialising the dominance values with anything other than zero introduces a harmful hitchhiker bias, which can lead to premature convergence.

Reducing the credit bias had an adverse affect on the GA. Reducing the magnitude of the dominance increase resulted in the GA converging to poor solutions compared to the original algorithms in both the NK Landscapes and the Royal Road function. This shows that although a directional bias can aid the GA, the GA is very sensitive to the credit bias imposed. This raises the question: *"How much credit bias to impose to ensure a balance between exploration and exploitation?"* Our intuition is that the balance is specific to the problem the adaptive GA is being applied to.

From the results we can see that without mutation uniform crossover performed better; it found better solutions in both problems. In a previous study when mutation was used (Vekaria, 1998), selective crossover outperformed uniform crossover by finding the complete solution (for the Royal Road problem) and doing so in the least number of evaluations. This result may be due to the fact that mutation in selective crossover is normal point mutation and does not increase the hitchhiker bias. If mutation increased the fitness of the individual then only that allele's dominance value gets increased; therefore exhibiting no increase in the hitchhiker bias.

These results show how sensitive a GA is to adaptive recombination due to the biases imposed and the degree at which they are imposed. The initialisation bias in adaptive recombination is not beneficial for GA performance. The GA is very sensitive to the amount of credit bias imposed.

6. SUMMARY AND CONCLUSION

We have studied the biases on alleles introduced by adaptive recombination operators that use fitness information to direct the GA. We have identified four key different biases in adaptive recombination: *directional bias, credit bias, initialisation bias* and *hitchhiker bias.* Directional bias exists if alleles are either favoured or not favoured for their credibility. Credit bias is the degree at which an allele gets favoured with respect to its credibility. Initialisation bias exists if alleles get favoured during initialisation without knowing their credibility. Hitchhiker bias exists if alleles get favoured when they do not contribute to the fitness increase.

We have analysed the relationship between these biases and have shown that they are not independent: credit bias is a direct result of the directional bias, and both credit bias and initialisation bias contribute to hitchhiker bias. We have used these four biases to characterise three adaptive recombination operators (masked, adaptive uniform, and selective crossover) and to compare the degree to which they exhibit these biases; selective crossover has higher directional bias yet lower credit bias and lower hitchhiker bias than the other two operators. By comparison, uniform crossover (a non-adaptive operator) exhibits none of these biases.

We have explored the effects of the initialisation bias and credit bias on GA search by eliminating or reducing the biases in selective crossover and applying it to the Royal Road and NK Landscape problems. The results indicate that eliminating the initialisation bias (in the absence of mutation) does improve genetic search. Reducing the credit bias by the method proposed reduces the deleterious effects of the hitchhiker bias yet has an overall adverse effect on the performance of the GA; this illustrates the sensitivity of selective crossover towards the credit bias.

We conclude that, whilst some bias is essential for the operation of an adaptive mechanism, these biases are not always beneficial; furthermore, introducing biases to aid a GA can also unintentionally introduce other biases, to which the GA may be very sensitive. Thus, design of adaptive recombination techniques is a complex task, which is not yet fully understood and requires further investigation.

7. FUTURE WORK

Future extensions of this study intend firstly, to investigate the effects of applying different degrees of credit bias on problems that contain different characteristics. Secondly, look at the effects of these biases when mutation is re-introduced and to understand how mutation assists selective crossover (this study showed that mutation benefits selective crossover). Thirdly, to analyse biases in other local search GAs.

Acknowledgements

This research was supported by the Engineering and Physical Sciences Research Council. We thank Peter Bentley, Robin Hirsch and the anonymous reviewers for their valuable comments.

References

Booker, L.B. (1992) Recombination Distribution for Genetic Algorithms. *Proceedings of Foundations of Genetic Algorithms 2*, 29-44. Morgan Kauffman.

Davis, L. (1989) Adapting Operator Probabilities in Genetic Algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, 61-69. Morgan Kauffman.

Eshelman, L.J., Caruana, R. A., & Schaffer J. D. (1989) Biases in the Crossover Landscape. *Proceedings of the Third International Conference on Genetic Algorithms*, 10-19. Morgan Kauffman.

Eshelman, L.J. & Schaffer J.D. (1995) Productive Recombination and Propagating and Preserving Schemata. *Foundations of Genetic Algorithms 3,* 299-313. San Francisco, CA: Morgan Kauffman.

Forrest, S. & Mitchell, M. (1993) Relative Building-Block Fitness and the Building Block Hypothesis. *Foundations of Genetic Algorithms 2,* 109-126. San Francisco, CA: Morgan Kauffman.

Goldberg, D.E., Korb, B. & Deb, K. (1989) Messy Genetic Algorithms: Motivation, Analysis, and First Results. In *Complex Systems*, Vol. 3. 493-530.

Harik, G.R. (1997) *Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms.* Doctoral dissertation, University of Michigan, Ann Harbor.

Kauffman S.A. (1989) Adaptation on Rugged Fitness Landscapes. *Lectures in the Sciences of Complexity*, Vol1, 527-618. Addison Wesley.

Louis, S.J. & Rawlins, G.J.E. (1991) Designer Genetic Algorithms: Genetic Algorithms in Structure Design. *Proceedings of the Fourth International Conference on Genetic Algorithms*, 53-60. Morgan Kauffman.

Schaffer, J.D. & Eshelman L.J. (1991) On Crossover as an Evolutionary Viable Strategy. *Proceedings of the Fourth International Conference on Genetic Algorithms*, 61-68. Morgan Kauffman.

Schaffer, J.D. & Morishima, A. (1987) An Adaptive Crossover Distribution Mechanism for Genetic Algorithms. *Proceedings of the Second International Conference on Genetic Algorithms*, 36-40. Lawrence Erlbaum Associates.

Smith, J.E. (1998) *Self Adaptation in Evolutionary Algorithm.* Doctoral dissertation, University of the West of England, Bristol, UK.

Spears, W.M. (1993) Crossover or Mutation. *Proceedings of Foundations of Genetic Algorithms 2*, 221-237. Morgan Kauffman.

Spears, W.M. (1995) Adapting Crossover in Evolutionary Algorithms. *In Proceedings of the Fourth Annual Evolutionary Programming Conference*, 367-384.

Spears, W.M. (1997) Recombination Parameters. *The Handbook of Evolutionary Computation*, Oxford University Press.

Spears, W.M. (1998) *The Role of Mutation and Recombination in Evolutionary Algorithms.* Doctoral dissertation, George Mason University, Virginia.

Vekaria, K. (1998) Increasing the Reliability of Genetic Algorithms Using Adaptive Recombination. Internal Note IN/98/2, Dept. of Computer Science, University College London, UK.

Vekaria, K. & Clack, C. (1998) Selective Crossover in Genetic Algorithms: An Empirical Study. *Proceedings of the 5th Conference on Parallel Problem Solving from Nature*, 438-447. Springer-Verlag.

White, T. & Oppacher, F. (1994) Adaptive Crossover Using Automata. *Proceedings of the Parallel Problem Solving from Nature Conference,* 229-238. NY:Springer Verlag.

Wu, S., Lindsay, R.K. & Riolo, R.L. (1997) Empirical Observations on the Role of Crossover and Mutation. *Proceedings of the Seventh International Conference on Genetic Algorithms,* 362-369. Morgan Kauffman.