Selective Crossover

as an

Adaptive Strategy for Genetic Algorithms

Kanta Premji Vekaria



A dissertation submitted in partial fulfilment of the requirements for the degree of **Doctor of Philosophy**

of the

University of London

Department of Computer Science University College London

March 2000

ProQuest Number: 10011159

All rights reserved

INFORMATION TO ALL USERS The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10011159

Published by ProQuest LLC(2016). Copyright of the Dissertation is held by the Author.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code. Microform Edition © ProQuest LLC.

> ProQuest LLC 789 East Eisenhower Parkway P.O. Box 1346 Ann Arbor, MI 48106-1346

Abstract

Since the proposal of the first genetic algorithm (GA) many recombination operators have been proposed. Some are problem specific and require a great deal of knowledge about the problem being solved, resulting in good but highly specialised operators. Other recombination operators have been proposed for more general use. One advantage for such operators is the little knowledge required about the problem being solved; however, the synergy of these operators, the problem being solved and other GA parameters does not always yield optimum performance from the GA. More recently, adaptive recombination operators have been proposed to bridge the gap between general and specialised recombination operators.

This thesis presents a novel adaptive recombination operator, namely "Selective Crossover", for use with a genetic algorithm. Selective crossover was designed with three properties that make it a viable strategy to use when little or no knowledge is available about the problem being optimised.

The first property is the identification of allele changes made to the candidate solution during recombination. The second property is the use of correlations between parental and offspring fitnesses to discover beneficial alleles. The third property is the preservation of alleles at each locus, during recombination, according to their previous contributions to beneficial changes in fitness.

This thesis makes six contributions. The first is the design and implementation of selective crossover. The second is a measurement and comparison of the performance of selective crossover and two traditional recombination operators on a number of different problems. The third is an empirical analysis of the adaptive properties in selective crossover. The fourth is an identification and analysis of four key biases inherent in selective crossover and a demonstration of the existence of these biases in two other similar operators. The fifth is an analysis and comparison of schema propagation in selective crossover and two traditional recombination operators. The final contribution is a construction of a schema survival probability for selective crossover.

To my brother Lalji, and sisters Bhanu and Damyanti with gratitude and love.

Time passes, There is no way We can hold it back — Why, then, do thoughts linger on, Long after everything else is gone?

— Ryokan, 1758 - 1831

Acknowledgements

I would like to thank my supervisor Chris Clack for his constant support, encouragement, guidance, constructive criticisms, and for reading multiple drafts of this thesis.

This work was partially funded by EPRSC and the UCL Computer Science Department who never refused me any conference funds. Endless thanks to BBC News Online for allowing me to work on the project at unusual part-time hours, which in turn allowed me to finance my PhD in the first two years. Special thanks to Matthew Karas who made that possible. To Richard Chandler for the many debates about the theory of selective crossover and for answering my endless statistical questions. To Bill Langdon for thoroughly reading and constructively criticising the draft of this thesis. To David Goldberg, William Spears, Lee Altenberg and Richard Watson for their helpful comments and suggestions as well as showing a great deal of enthusiasm and interest in my work. Additional thanks to William Spears for the use of his GA code (GAC) and Mitchell Potter for use of his code on NK landscapes. My gratitude to Kwok Wong for his creative skills in constructing some figures in this thesis.

My sincere thanks and love to my parents without whom I would not have come this far. Special thanks to my sister-in-law, Vanita Vekaria, for being extremely understanding. My nephew Vijay and niece Pratiksha for their daily hugs worth waking up for. To my fellow 'sufferers' at UCL Computer Science, past and present, for their kind advice and being extremely reassuring through the traumatic moments; Julia Schnabel, Mazliza Othman, Jungwon Kim, Nadia Kausar, Laura Dekker, and Jose Borges. Special thanks to Jorge Ortega-Arjona and Manlio Valdivieso-Casique for constantly cheering me up – *Muchas gracias*! Also thanks to Angela Gupta and Hitesh Mistry (the fellow survivors of WHS) and Hiral Patel for their constant support and great friendship of many years and many years to come. And last but not least, sincere thanks to Darren White, my dearest friend, for his constant moral encouragement, for standing by me even in my worst moods and more importantly for not letting me quit when the going got tough.

Contents

1	I	ntrod	uction	.6
	1.1	Moti	vation1	.7
	1.2	Thesi	is Objective1	.9
	1.3	Cont	ributions2	20
	1.4	A Ro	ad Map for this Thesis2	21
2	B	lackgi	round Work2	4
	2.1	What	t is a Genetic Algorithm?2	24
	2.2	Term	ninology2	25
	2.3	Com	ponents of the Genetic Algorithm2	26
	2	.3.1	Population2	26
	2	.3.2	Evaluation2	28
	2	.3.3	Selection2	29
	2	.3.4	Recombination	30
	2	.3.5	Mutation3	\$0
	2.4	Choo	osing Operators and Parameters	31
	2.5	Schei	ma Theorem and Building Block Hypothesis3	3
	2.6	Epist	asis and Deception3	6
	2.7	Sumr	mary3	57
3	R	Related	d Work3	8
	3.1	Intro	duction to Recombination	9

	3.2 Stat	tic Recombination Operators	
	3.2.1	One-point Crossover	
	3.2.2	N-point Crossover	
	3.2.3	Uniform Crossover	
	3.2.4	Bit-Based Simulated Crossover	
	3.3 Ada	aptive Recombination Operators	
	3.3.1	Schaffer and Morishima (1987)	
	3.3.2	Louis and Rawlins (1991)	
	3.3.3	White and Oppacher (1994)	
	3.3.4	Eshelman and Schaffer (1995)	44
	3.3.5	Spears (1995)	
	3.4 Bias	ses in Static Recombination Operators	45
	3.4.1	Positional Bias	45
	3.4.2	Distributional Bias	45
	3.4.3	Exploration and Exploitation	
	3.5 The	eoretical Static Recombination Analysis	47
	3.5.1	Framework	47
	3.5.2	Schema Survival Probability for Uniform Crossover	47
	3.6 Enc	coding and Linkage in Genetic Algorithms	49
	3.6.1	Holland (1992)	50
	3.6.2	Goldberg, Korb and Deb (1989)	50
	3.6.3	Kargupta (1996)	51
	3.6.4	Harik (1997)	51
	3.6.5	Smith (1998)	51
	3.7 Clas	ssification of Parameter Adaptation in Genetic Algorithms	
	3.8 Sun	nmary	54
4	Select	tive Crossover	
	4.1 Inst	piration	
	4.2. Ter	minology	58
	12 V.	v Properties	5 0
	т. л. г.су	y I 10/01/105	

	4.4	Impl	ementation	59
	4.5	Matl	hematical Representation	62
	4.6	Revi	iew of Properties	65
	4.7	Exa	mples of Recombination	66
	4.8	Revi	iew of Strategies Used	70
	4.9	Sum	mary and Conclusions	71
5	Р	erfoi	rmance Evaluation of Selective Crossover	73
	5.1	Exp	erimental Method	73
	5.2	GA	Parameters	76
	5.3	Test	Suite of Benchmark Problems	76
	5.	.3.1	One Max Problem	77
	5.	.3.2	Royal Road Function	77
	5.	.3.3	Random L-MaxSAT Problems	79
	5.	.3.4	NK Landscape Problems	81
	5.	.3.5	Deceptive Trap Functions	83
	5.4	Exp	erimental Results	.85
	5.	.4.1	One Max Problem	85
	5.	.4.2	Royal Road Function	87
	5.	.4.3	Random L-MaxSAT Problems	88
	5.	.4.4	NK Landscape Problems	.90
	5.	.4.5	Deceptive Trap Functions	.91
	5.5	Ana	lysis and Discussion of Results	.93
	5.6	Sum	nmary and Conclusions	.94
6	F	'eatu	res of Adaptation in Selective Crossover	96
	6.1	Clas	sification of Selective Crossover	.96
	6.	.1.1	Deterministic Features	.97
	6	.1.2	Adaptive Features	98
	6	.1.3	Self-Adaptive Features	98

	6.1.4	A New Taxonomy for Type of Change	
	6.2 Ana	lysis of the Adaptive and Self-Adaptive Features	102
	6.2.1	Experiments	102
	6.2.2	Results and Analysis	103
	6.3 Ana	lysis of the Dominance Values in the Population	105
	6.3.1	Experiments	105
	6.3.2	Results and Analysis	106
	6.4 Sun	nmary and Conclusions	
7	Biase	s in Adaptive Recombination Operators	
•	71 5		112
	7.1 Exis	sting Allele-Based Adaptive Recombination Operators	113
	7.1.1	Masked Crossover	113
	/.1.2	Adaptive Uniform Crossover	110
	7.2 Ana	alysis of Biases in Allele-Based Adaptive Recombination Operators	119
	7.2.1	Directional Bias	119
	7.2.2	Credit Bias	119
	7.2.3	Initialisation Bias	
	7.2.4	Hitchhiker Bias	122
	7.2.5	Interdependencies Between Biases	
	7.3 Mo	re Hitchhiking and Selective Crossover	125
	7.4 Ana	alysis of the Effect of Biases on Selective Crossover	127
	7.4.1	Selective Crossover with Less Credit Bias	
	7.4.2	Selective Crossover without the Initialisation Bias	
	7.4.3	Results	132
	7.5 Ana	alysis	134
	7.6 Sur	nmary and Conclusions	
8	8 Schei	ma Propagation in Selective Crossover	138
-	0 1 41	. O	120
	8.1 Alt	ernative Koyai Koad Encodings	139
	8.2 Ana	alysis of Schema Propagation	142
	8.2.1	Experiments	142

	8.	2.2	Results on Performance
	8.	2.3	Results on Schema Propagation
	8.	2.4	Analysis
	8.3	Sche	ema Survival Probability for Selective Crossover151
	8.4	Sum	mary and Conclusions
9	C	oncl	usions
	9.1	Disc	cussion of Results160
	9.2	Con	tributions of this Thesis
	9.3	Lim	itations of this Thesis164
	9.4	Futu	ıre Work165
B	iblio	grap	ohy168
A	ppe	ndix	A: Publications178

.

List of Figures

Figure 2.1: Generational cycle of a conventional genetic algorithm
Figure 3.1: One-point crossover
Figure 3.2: Two-point crossover
Figure 3.3: Uniform crossover
Figure 3.4: An example of hyperplane survival - Parent1 is a member of the third-order
hyperplane $H_3 = "**111"$ and Parent2 is an arbitrary string. Recombination is
performed to produce two offspring
Figure 4.1: A Generational Cycle of a Genetic Algorithm using Selective Crossover59
Figure 4.2: An individual in selective crossover consists of an additional dominance
vector60
Figure 4.3: Recombination with Selective Crossover
Figure 4.4: Updating Dominance Values
Figure 5.1: Simple Royal Road Function
Figure 5.2: An example of epistatic interactions for an NK landscape where N=4 and
K=282
Figure 5.3: A Deceptive Trap function where a bit string of all 1s is the global
optimum and a bit string of all 0s is the local optimum
Figure 5.4: Results for the One Max problem - performance comparison of selective,
two-point and uniform crossover
Figure 5.5: Results for the Royal Road function - performance comparison of selective,
two-point and uniform crossover
Figure 5.6: Results for the Random L-MaxSAT problems - performance comparison of
selective, two-point and uniform crossover
Figure 5.7: Results for the NK Landscape problems - performance comparison of

selective, two-point and uniform crossover
Figure 5.8: Results for the Deceptive Trap problems - performance comparison of
selective, two-point and uniform crossover
Figure 6.1: Taxonomy of parameter control by Eiben et al. (1999)
Figure 6.2: New taxonomy for type of parameter control
Figure 6.3: Classification of other adaptive recombination operators using this new
taxonomy
Figure 6.4: GA Cycle of selective crossover that does not change the dominance
values
Figure 6.5: The evolution of the distribution of dominance values in the population, for
the One Max problem107
Figure 6.6: The evolution of the distribution of dominance values in the population, for
the Royal Road function108
Figure 6.7: The evolution of the distribution of dominance values in the population, for
the NK Landscape N=32 K=8109
Figure 7.1: Recombination using masked crossover
Figure 7.2: Masked crossover - creation of new binary masks if both children are
'good'. (? Denotes randomly generated)114
Figure 7.3: Masked crossover - creation of new binary masks if both children are
'bad'. Assume fitness of Parent2 > Parent1. (? Denotes randomly generated) 114
Figure 7.4: Recombination using adaptive uniform crossover (AUX). The automata
belonging to the offspring are updated here given that the fitness of Child1 >
Parent1, Parent2 and fitness of Child2 < Parent1, Parent2117
Figure 7.5: Population of 4 individuals, which demonstrates that the dominance values
are not uniformly distributed at each locus due to the small population size. As the
population size increases the distribution of the dominance values at each locus is
more likely to be uniform
Figure 7.6: Example of hitchhiker bias in selective crossover when applied to the One
Max problem123
Figure 7.7: Interdependencies amongst biases in allele-based adaptive recombination
operators
Figure 7.8: Reducing the magnitude of the credit bias by sharing the fitness increase
amongst alleles that were exchanged. Child1 produced a fitness increase of 0.1.

Since two alleles were exchanged their dominance values get increased by 0.05 Figure 8.1: Encoding A -the original encoding of a Royal Road function. In this Figure 8.2: Encoding B - A new encoding of a Royal Road function. In this encoding Figure 8.3: Encoding C - A new encoding of a Royal Road function. In this encoding Figure 8.4: Encoding D – A new encoding of a Royal Road function. In this encoding Figure 8.5: Example of schema propagation in a single recombination event. Parent1 contains schemata S1, S3 and S7. Parent2 contains schemata S1, S4, S8. Schemata S1 and S3 were propagated from Parent1 and schemata S1, S4 and S5 were propagated from Parent2......143 Figure 8.6: Schema propagation in two-point crossover with different Royal Road encodings. DL# is the defining length of level 0 schemata. The results show the Figure 8.7: Schema propagation in uniform crossover with different Royal Road encodings. DL# is the defining length of level 0 schemata. The results show the Figure 8.8: Schema propagation in selective crossover with different Royal Road encodings. DL# is the defining length of level 0 schemata. The results show the number of level 0 schemata propagated per individual at each generation......147 Figure 8.9: Encoding A – the original Royal Road encoding where the defining length of level-0 schemata δ (level-0) =7. A comparison of schema propagation between Figure 8.10: Encoding B - a Royal Road encoding where the defining length of level-0 schemata $\delta(\text{level-0}) = 14$. A comparison of schema propagation between two-

Figure 8.11: Encoding C – a Royal Road encoding where the defining length of level-0 schemata δ (level-0) =28. A comparison of schema propagation between two-

point, uniform and selective crossover.	149
Figure 8.12: Encoding $D - a$ Royal Road encoding where the defining length of level-	0
schemata $\delta(\text{level-0}) = 56$. A comparison of schema propagation between two)-
point, uniform and selective crossover.	150

List of Tables

Table 2.1: An example collection of operators and parameters associated with a
genetic algorithm
Table 5.1: A table of fitness contributions for the NK landscape given in Figure 5.282
Table 5.2: Results for the One Max problem. Mean number of evaluations taken to
solve the One Max problem. The standard deviation is shown in brackets
Table 5.3: Results for the Royal Road function. Mean number of evaluations taken to
solve the Royal Road function. The standard deviation is shown in brackets
Table 5.4: Results for the Random L-MaxSAT problems. Mean number of evaluations
to find the best solution for low, medium and high epistasis. The standard
deviation is shown in brackets
Table 5.5: Results for the NK Landscape problems. Mean number of evaluations to
find the best solution for $K=4$, 20 and 31. The standard deviation is shown in
brackets90
Table 5.6: Results for the Deceptive Trap problems. Mean number of evaluations
completed for order 10, 15 and 20 partially deceptive trap functions. The standard
deviation is shown in brackets. The solution quality is given in square brackets91
Table 5.7: A summary of relative performance of selective, two-point and uniform
crossover. A 1* represents best performance93
Table 6.1: Results of applying selective crossover without its adaptive feature to the
One Max, Royal Road and NK landscapes. Previous results with adaptive feature
are also recalled in this table104
Table 7.1: Updating rules for adaptive uniform crossover (AUX). +P _{superiorReward} is the
probability with which an automaton will be rewarded – moved to the next state

 $P_{inferiorPenalty}$ is the probability with which an automaton will be penalised – moved

to the previous state
Table 7.2: Relative strength of biases present in masked crossover, adaptive uniform
crossover and selective crossover
Table 7.3: Mean fitness of best solution found when analysing the biases in selective
crossover without mutation in the GA. Results are from 50 independent runs for
the Royal Road and NK Landscape problems. The standard deviation is shown in
brackets132
Table 7.4: Mean solution found and mean evaluations taken when analysing the biases
in selective crossover with mutation in the GA. These are results from 50
independent runs for the Royal Road function. On all runs the global solution was
found. The standard deviation is shown in parentheses
Table 7.5: Mean solution found and mean evaluations taken when analysing the biases
in selective crossover with mutation in the GA. Results from 50 independent runs
for NK landscapes N=32 K= $\{4, 20, 31\}$. The standard deviation is shown in
parentheses133
Table 8.1: Mean number of evaluations taken to find the solution for different Royal
Road encodings. The standard deviation is shown in parentheses

Chapter 1

Introduction

The fundamental problem of optimisation is to arrive at the best possible decision in any given set of circumstances. There are many situations where the 'best' is unattainable for one reason or another; often we may not be sure what is meant by the 'best'. The first step therefore in an optimisation problem is to choose some quantity, typically a function of several variables, to be maximised or minimised, subject possibly to one or more constraints. The commonest types of constraint are equalities and inequalities, which must be satisfied by the variables of the problem. The next step is to choose a method to solve the optimisation problem; such methods are usually called optimisation techniques, or optimisation algorithms. One such algorithm is the genetic algorithm.

First pioneered by John Holland in the 1960s (Holland 1992), genetic algorithms have been demonstrated to be a successful optimisation algorithm. They were inspired by, and mimic, some of the processes observed in natural evolution. Based on the Darwinian principle of 'survival of the fittest', genetic algorithms manipulate a population of candidate solutions using selection, recombination and mutation processes. These processes allow good solutions to survive in preference of weaker ones.

The genetic algorithms that have been proposed for optimisation problems in recent years have tended to become more and more elaborate (Goldberg 1989a; Michalewicz 1994, Bäck 1996; Mitchell 1996); the result being a large parameter space for a genetic algorithm. Thus, choosing the appropriate parameters for a genetic algorithm has become a more difficult task and is in itself an optimisation problem.

One such parameter is recombination, which is considered essential for the success

of genetic algorithms and is thought to be responsible for the generation and propagation of solutions (Holland, 1992; Schaffer and Eshelman, 1991; Spears, 1993). Since the proposal of the first genetic algorithm many recombination operators have been developed. Some are problem specific and require a great deal of knowledge about the problem being solved, resulting in good but highly specialised operators. For example, on ordering problems a number of recombination operators have been proposed that respect the ordering of alleles (Goldberg and Lingle, 1985; Starkweather et al., 1991; Falkenauer, 1994). Other recombination operators have been proposed for more general use (De Jong, 1975; Syswerda, 1989; Spears and De Jong, 1991b). One advantage for general operators is the little knowledge required about the problem being solved; however, the synergy of these operators and the problem being solved and other parameters does not always yield best performance from the genetic algorithm. More recently, adaptive recombination operators have been proposed to bridge the gap between general and specialised recombination operators (Schaffer and Morishima, 1987; Louis and Rawlins, 1991; White and Oppacher, 1994; Eshelman and Schaffer, 1995; Spears, 1995). The aim of these adaptive methods is to adapt dynamically to problem characteristics in the hope of creating a more robust optimisation strategy. We know from "No Free Lunch" theorems (Wolpert, and Macready 1997) that an algorithm that is suited for all problems cannot exist, since for fixed parameter/operator sets there will be problems for which they are optimal and other problems for which their performance is poor. However, is it possible to devise an adaptive recombination operator that is a suitable strategy to use for a wide range of problems in which little is known about the problem space being searched?

1.1 Motivation

This thesis presents a novel adaptive recombination operator, namely "Selective Crossover", for use with a genetic algorithm. Its inspiration comes from natural evolution, specifically Dawkins' model of evolution and dominance characteristics in nature. Dawkins' (1989) model of evolution is based on the gene. In his theory the gene is considered to be the fundamental unit of natural selection. Since natural selection acts on the individual and the individual consists of unique chromosomes, these chromosomes actually have a life span of one generation. However, the chromosome consists of many

genes and a subset of these genes last for many more generations because it is the genes that are passed onto the offspring not the entire chromosome. Thus natural selection favours the gene.

Dominance in nature is associated with genetic material within diploid chromosomes (two sets of chromosomes) where the alleles contained in one set can be regarded as a direct alternative to the alleles in the other set. When building the organism the alleles in one set compete with those in the other set. Alleles that are dominant are expressed in the phenotype of an organism and those that are less likely to be expressed are recessive. The relationship between a dominant and recessive gene is complex: some genes that have been known to be dominant have become more recessive in successive generations and vice versa. Merrell (1984) suggests that these shifts in dominance are in response to changes in the environment. Thus, those genes that increased an individual's fitness have become more dominant by evolving over generations; however, a precise model to show this is not available.

Selective crossover, our new adaptive recombination operator, uses both the analogy of dominance, where alleles in one chromosome compete with those on the other chromosome, and the analogy of evolution of dominance. The purpose is to see if recombination at each allele in a haploid (single chromosome) genetic algorithm can be evolved such that alleles in one parent compete with those on the other parent chosen for crossover. Here the alleles are competing to be retained in a fitter individual and the use of correlations between parental and offspring fitnesses would allow the means of discovering beneficial alleles. This in turn allows recombination in the genetic algorithm to adapt to the problem space being searched.

Adaptive recombination operators that have been proposed by others (Schaffer and Morishima, 1987; Louis and Rawlins, 1991; White and Oppacher, 1994; Eshelman and Schaffer, 1995; Spears, 1995) provide an evaluation of their algorithm or strategy in terms of performance comparisons with other techniques. This is usually done using a test suite of problems or a real world application. Little effort is given to analysing "how?" and "what?" provides this increased performance in the new strategy. This extra information would also allow us to understand the limitations of the new strategy and provide us with justifications for its use on other problems. For these reasons we have conducted four different evaluations of selective crossover in terms of performance, adaptive behaviour, biases and schema propagation.

1.2 Thesis Objective

This thesis examines the following hypothesis:

Hypothesis:

When little or no knowledge is available about the problem being optimised by a genetic algorithm, a viable strategy is to use an adaptive recombination operator with the following three properties:

- 1. **Detection** It detects alleles that were changed during recombination to identify modifications made to the candidate solution.
- 2. Correlation It uses correlations between parental and offspring fitnesses as a means of discovering beneficial alleles.
- 3. **Preservation -** It preferentially preserves alleles at each locus, during recombination, according to their previous contributions to beneficial changes in fitness.

There are two main aims of this thesis. First, to design and implement a new adaptive recombination operator, *"selective crossover"* with the above three properties. Second, to undertake an extensive evaluation of selective crossover using four different criteria. As we shall see, both aims have been achieved.

Selective crossover was designed with three key properties. Firstly, selective crossover detects alleles that were changed during crossover to identify actual modifications made to the candidate solution during recombination. Secondly, this acquired knowledge is then combined with parental and offspring fitness correlations to discover potentially beneficial alleles. Finally, alleles are preferentially preserved, during recombination, according to their previous beneficial fitness contributions.

Selective crossover was first evaluated in terms of performance where performance is measured as the number of evaluations taken to solve a problem. The performance of selective crossover was compared with two traditional recombination operators, two-point and uniform crossover, on a set of five different and well-studied benchmark problems. Given our performance measure selective crossover was demonstrated to be better or comparable to two-point and uniform crossover on most problems.

The second evaluation in terms of adaptive features within selective crossover allowed us to observe empirically the dynamics of selective crossover on different problems. This confirmed the internal adaptive behaviour of selective crossover and also demonstrated that selective crossover adapts to each problem in a different manner.

The third evaluation was completed by a critical analysis of the biases that selective crossover imposes on search. This identified some limitations and possible enhancements to selective crossover.

The final evaluation, in terms of schema propagation, was undertaken using different encodings of a problem and tracking schema, in the population, before and after recombination. The alternative encodings positioned genes at different locations on the chromosome and were used to analyse the affect they have on the performance of selective crossover, two-point crossover and uniform crossover. The schema propagation of all three recombination operators was compared, allowing three conclusions to be drawn. Firstly, the performance of selective crossover is consistent regardless of the encoding used. Secondly, the survival rate of a schema in selective crossover is not affected by the encoding. Thirdly, selective crossover provides a better balance between exploration and exploitation than the two other traditional recombination operators.

1.3 Contributions

This thesis makes six main contributions.

- 1. The design and implementation of *"selective crossover"*, a new adaptive recombination operator that incorporates correlations between parents and offspring as a means of discovering and preserving beneficial alleles at each locus during recombination to produce fitter offspring.
- 2. A measurement and comparison of the performance of selective crossover and two traditional recombination operators on a number of different problems.

- 3. An empirical analysis that demonstrates adaptive behaviour in selective crossover.
- 4. An identification of four key biases inherent in selective crossover, a demonstration of the existence of these biases in two other similar operators and an empirical analysis to study the effects of these biases on selective crossover.
- 5. An analysis and comparison of schema propagation in selective crossover and two traditional recombination operators.
- 6. A construction of a schema survival probability for selective crossover; this demonstrates that schema survival in selective crossover is problem dependent.

This thesis also makes a secondary contribution; a description of a new taxonomy to classify selective crossover and other adaptive strategies in evolutionary computation, which overcomes the limitations in the existing taxonomy provided by Eiben *et al.* (1999)(described in Chapter 3).

1.4 A Road Map for this Thesis

After this introduction Chapter 2 presents a traditional genetic algorithm. It highlights both the vast parameter space associated with genetic algorithms and the issues surrounding the choice of parameters that have led to the development of adaptive strategies. Additionally, Chapter 2 presents the original theoretical explanation of genetic algorithms, due to Holland (1992) and discusses the concepts of epistasis and deception.

Chapter 3 focuses on the recombination operator and provides a survey of work on static recombination operators, adaptive recombination operators and strategies proposed to learn linkage. It also presents theoretical work on schema survival in static recombination operators. The increase in adaptive strategies has led to many classifications of adaptation that are also presented here.

Chapter 4 provides a detailed description of selective crossover and emphasises its three key properties. The next four chapters describe experiments.

Chapter 5 provides an empirical evaluation in terms of performance. Selective crossover is compared with two static recombination operators, two-point and uniform crossover (described in Chapter 3), on a set of five different and well-studied benchmark problems. This evaluation shows that the performance of selective crossover is either superior or comparable to two-point and uniform crossover.

Chapter 6 identifies the limitations of the taxonomy to classify adaptive strategies provided by Eiben *et al.* (described in Chapter 3). A new taxonomy is presented which accommodates strategies that use one or more methods of change. This new taxonomy is also used to classify and identify the methods of change in selective crossover. An empirical analysis demonstrates that without the three key properties (see hypothesis), the performance of selective crossover is worse than two-point and uniform crossover. This chapter also provides an empirical analysis of the adaptive behaviour in selective crossover. The experiments to track the population dynamics conclude by demonstrating that selective crossover adapts recombination and also suggests that selective crossover adapts to the problems being optimised, in contrast to static behaviour.

Chapter 7 presents a critical analysis of selective crossover and two other similar adaptive recombination operators. Four key biases were identified to be inherent in selective crossover; directional, credit, hitchhiker and initialisation bias. Experiments show that some of these biases are detrimental to the performance of selective crossover; however they can be reduced and thus further enhance selective crossover.

Chapter 8 describes a series of experiments to investigate schema propagation in selective crossover. Two conclusions result from these experiments: firstly, selective crossover is insensitive to the encoding of gene positions in the chromosome, unlike other recombination operators. Selective crossover shows consistent behaviour even when the encoding is altered so that related genes are located at the extremes of the chromosome. Secondly, when the schema propagation of selective crossover is observed and compared with two-point and uniform crossover, selective crossover appears to provide more exploration in early generations and more exploitation in later generations in comparison to the other two operators.

Chapter 8 also constructs a schema survival probability for selective crossover, which demonstrates that schema survival is dependent on the current dominance values in the population. Since the dominance values are a function of the evaluation function, this suggests that schema survival in selective crossover is problem dependent.

22

Chapter 9 concludes the main body of the thesis with a discussion of the adaptive recombination operator developed and results produced by the evaluations undertaken. Conclusions on the use of selective crossover are made and avenues of future work are also discussed.

Chapter 2

Background Work

The aim of this chapter is to set the scene for this thesis by providing an introduction to the conventional genetic algorithm. We first provide (in Section 2.1) an introductory paragraph to set genetic algorithms in context with other evolutionary algorithms. This is followed by an introduction to the terminology drawn from natural genetics that is used in genetic algorithms. Section 2.3 then provides a description of each component in a conventional genetic algorithm and Section 2.4 highlights the many operators and parameters associated to the genetic algorithm and how difficult it is to choose appropriate settings that give best performance. A brief explanation of the schema theorem and building block hypothesis is provided in Section 2.5. Finally, Section 2.6 introduces two concepts, epistasis and deception, which are used in GA research to characterise problems in terms of difficulty and are used in this thesis to choose problems that serve as a good test-bed to evaluate selective crossover.

2.1 What is a Genetic Algorithm?

Genetic algorithms (GAs) are a family of adaptive techniques, devised by John Holland (Holland, 1992), that may be used to solve search and optimisation problems. GAs were inspired by and mimic some of the processes observed in natural evolution (i.e. survival of the fittest) and are thus characterised as evolutionary algorithms. There are other evolutionary algorithms such as Evolution Strategies (Schwefel, 1981), Evolutionary Programming (Fogel *et al.*, 1966) and more recently GAs have been extended to form

Genetic Programming (Koza 1992). However, GAs have four elements that together differentiate them from these other evolutionary algorithms:

- 1. A population of chromosomes.
- 2. Selection according to fitness.
- 3. Recombination to produce new offspring.
- 4. Random mutation of new offspring.

Many of the terms used in GAs are drawn from biology hence we first introduce the terminology in the next section.

2.2 Terminology

All living organisms consist of cells, and each cell contains the same set of one or more *chromosomes*. A chromosome is comprised of numerous *genes* that encode for a particular *trait*, such as eye colour. The set of possible values taken up by each trait are called *alleles* (e.g. blue, green, brown). The position of a gene is defined by its *locus* and is identified separately from the gene's function.

Many organisms have multiple chromosomes in each cell. Those organisms with a pair of chromosomes are called *diploid* and organisms with a single set of chromosomes are called *haploid*. In nature, most sexually reproducing organisms are diploid.

Sexual reproduction in diploid and haploid organisms occurs in different ways. During diploid sexual reproduction, *recombination* (or crossover) takes place. In each parent a *gamete* (new single chromosome) is formed by combining genes between each pair of chromosomes. The gametes from the two parents combine to create a complete set of diploid chromosomes. In haploid sexual reproduction, genes are exchanged between two parents' single-strand chromosomes to form the new haploid chromosome. *Mutation* takes place due to recombining errors and is usually an allele change in offspring. The *fitness* of an organism is defined as the probability that the organism will survive to reproduce.

Most GA applications use *haploid* (single-chromosome) individuals. The term *chromosome* refers to a candidate solution to a problem. Quite often these chromosomes are also called individuals. The *genes* are the single units or short blocks of adjacent units in the chromosome. Genes are located at certain places on the chromosome, which are

called *loci*. Each gene encodes to a particular element, *trait*, of the candidate solution. The values taken up by a gene are *alleles* and are defined by the alphabet used to make up the candidate solution. Given a bit string representation the alphabet is the set $\{0,1\}$ and thus an *allele* is either 0 or 1. The set of parameters presented by a particular chromosome is referred to as a *genotype*. The genotype contains the information to construct the solution (as in genetic terms to construct the organism), which is referred to as the *phenotype*.

2.3 Components of the Genetic Algorithm

In this section we describe a conventional genetic algorithm (GA), which is just one of the many ways of implementing a GA. For an extensive review of current strategies and alternative implementations of the GA the reader is referred to Goldberg (1989a), Michalewicz (1994) and Mitchell (1996).

The conventional GA is comprised of five components: population, evaluation, selection, recombination and mutation. The use and issues surrounding each component are highlighted in the successive sub-sections. Each cycle of a GA is called a *generation* and is represented as shown in Figure 2.1 and Algorithm 2.1. In each generation the GA manipulates a population of chromosomes or individuals using evaluation, selection, recombination and mutation processes. The GA continues to run through many generations until either a solution is found or once a fixed number of generations have elapsed; this termination criterion is pre-determined by the GA practitioner.

2.3.1 Population

The GA maintains a population of chromosomes, candidate solutions, with associated fitness values. Therefore, before a GA can be run a suitable encoding or representation of the problem must be established. An encoding consists of a string of parameters related to the problem. For example, consider a parameter optimisation problem where a set of variables need to be optimised; to minimise or maximise some function $F(x_1, x_2,...,x_n)$. Each parameter x can be represented as a 10-bit binary number (suitably scaled). The chromosome would therefore comprise of 10n binary digits (genes) and its length is therefore 10n. Given a binary encoding the alleles will take up values of either 0 or 1.



Figure 2.1: Generational cycle of a conventional genetic algorithm.

```
Procedure Genetic_Algorithm;
T=0; /* starting generation */
Initialise_population(P);
Evaluation(P);
WHILE NOT finished DO {
T=T+1; /* Next generation */
Selection(P);
Recombination(P);
Mutation(P);
Evaluation(P);
}
```

Algorithm 2.1: Pseudo code for a conventional genetic algorithm.

Encodings for a GA are not limited to binary encodings. For many applications it is most natural to use an alphabet of many characters such as a q-ary alphabet (q > 2) or real numbers to form a chromosome. Moreover, there have been many extensions to the binary encoding such as Gray coding (Caruana and Schaffer 1988). The use of a binary encoding in preference to other encodings has been widely debated. Earlier work by Holland (1992) and Goldberg (1989a) suggest the use of a binary encoding as it offers the maximum use of schemata (see Section 2.5). More recently Reeves (1993) has also suggested that there are more theoretical advantages in using a binary encoding. However, Antonisse (1989) and Radcliffe (1992) have questioned these arguments. Additionally, empirical comparisons between binary encodings and multiple character encodings have shown better performance for the latter Michalewicz (1994). At present there are no conclusive findings that suggest the use of a particular encoding.

Other issues related to the encoding is the loci at which genes should be placed on the chromosome. Goldberg (1989a) suggests that genes should be strategically placed on the chromosome so that their positions can be exploited by the recombination operator. This is further discussed in the next chapter (Section 3.4).

Having established the encoding for the problem a population of candidate solutions must be created and the question is *how many (the population size)?* In most GAs a fixed sized population is used and is a fundamental decision faced by GA practitioners. If too small a population size is chosen, the GA will converge too quickly with insufficient processing of possible solutions. On the other hand, a population with too many members consumes a great deal of processing power with very little payoff in terms of better solutions. Several researchers have investigated the size of the population using binary encodings. Goldberg (1989b) suggests that the optimal population size grows exponentially with the length of the chromosome, which in a practical sense are extremely large populations. Grefenstette (1986) suggests the use of a small population size of 30 individuals. More recently, Reeves (1993) provides a theoretical justification for the use of smaller population sizes. Most usually optimum population sizes are constrained by the available machine resources and thus a strong preference is for small populations.

Having decided the population size p, the starting population is initialised by randomly generating p chromosomes. For the GA cycle to begin the chromosomes must first be evaluated and this is discussed in the next section.

2.3.2 Evaluation

An optimisation problem can be considered as a black box with a series of control dials representing the different parameters. The only output of the black box is a value returned by an evaluation or fitness function. The evaluation of each individual in the population is done using the fitness function, which assigns a figure of merit to a chromosome that is proportionate to the performance of the phenotype. The fitness function is usually given as part of the problem; in our previous example on function optimisation the fitness function is just the value returned by the function. However, it is not always so straightforward.

2.3.3 Selection

GA search is directed by selection with a bias towards areas in the search space that contain better solutions. The purpose of selection is to choose candidate parents for the process of recombination such that prominence is given to fitter individuals. There are many strategies to use for selection. Strong selection can lead to premature convergence, where a large proportion of the population consists of identical chromosomes that represent sub-optimal solutions. Weak selection on the other hand can make the search ineffective, where search makes very little progress in comparison to the resources being used. Numerous selection schemes have been proposed and compared in GA literature (Goldberg and Deb 1991; Thierens and Goldberg 1994; Blickle and Thiele 1995; Bäck 1996). A description of the most popular methods is given below.

2.3.3.1 Fitness Proportionate Selection

In fitness proportionate selection the expected number of offspring of an individual is equal to the fitness of that individual divided by the average fitness of the population. The most common method to implement this is the 'roulette wheel': each individual in the population is assigned a slot sized in proportion to its fitness. The wheel is spun N times (where N is the population size). The individual pointed by the wheel marker is selected.

Given a large enough population, this roulette wheel selection method will theoretically result in the expected number of offspring for each individual. However, the actual number of offspring allocated is far from the expected values. Baker (1987) proposed a different sampling method namely 'stochastic universal sampling' (SUS). This algorithm is analogous to roulette wheel but in this case the wheel is not spun N times but once using N equally spaced pointers. The number of copies an individual receives is then given by the number of pointers that fall in its slot.

Fitness proportionate selection is the oldest and most widely known method in GAs as it was first proposed by Holland. In all experiments in this thesis we use the SUS sampling method (Baker 1987).

2.3.3.2 Tournament selection

Tournament selection chooses some number t (tournament size) of individuals randomly from the population and copies the best individual from this group into the intermediate population. This is repeated N times to make up the N individuals in the population. In most cases the tournament size is two, but larger tournament sizes can be used in order to increase the selection pressure.

2.3.3.3 Ranking Selection

Baker (1985) introduced ranking selection as a way of relaxing the selection pressure to prevent premature convergence. The population is first sorted from best to worst. The expected value of offspring for each individual is a function of its rank. This selection method avoids giving the largest share of offspring to a small group of highly fit individuals and thus reduces the selection pressure.

2.3.3.4 Elitism

Elitism or an elitist strategy, first introduced by De Jong (1975), is a form of selection that is used together with the other selection methods like those described above. This strategy ensures that highly fit individuals in the population are passed onto the next generation without being altered by the recombination or mutation operators. This guarantees that the maximum fitness of the population can never reduce from one generation to the next.

2.3.4 Recombination

Recombination or crossover takes two individuals, and combines their chromosomes to create two offspring. Associated with recombination is the crossover rate - the probability that any individual will experience crossover, and unless it is set to 1, there is a chance some members in the current generation *do not* undergo crossover. The three most popular types of crossover are one-point crossover, two-point crossover and uniform crossover; these are further discussed in Section 3.2.

2.3.5 Mutation

The mutation rate governs the probability, which is normally low, that a gene or bit may experience mutation (some instantaneous change). For chromosomes encoded as bit strings, this would mean a bit change from one to zero or vice versa. Holland (1992) explained that the main purpose of mutation was as an "insurance policy" to avoid fixation on the search space. For example, without mutation, every string in the population may hold a one at the first bit position, and there would be no way of obtaining a zero at the first position. Therefore mutation is what helps provide diversity at a given bit position.

2.4 Choosing Operators and Parameters

In Section 2.3, a description of the components was provided and all but one component (evaluation) has associated with it a choice of operators and/or parameters. An example of possible choices that can be made is given in Table 2.1. There are many more GA operators and parameters; a comprehensive survey can be found in Michalewicz (1994) and Mitchell (1996).

Component	Operator	Parameter
Population		Population size
Selection	Selection Methods	Elitism
	Fitness proportionate selection	Roulette wheel SUS
	Tournament selection	Tournament size
	Ranking selection	
Recombination	Recombination operators	Recombination probability
	One-point crossover	
	Two-point crossover	
	Uniform crossover	
Mutation		Mutation probability

	Table 2.1	: An example	collection of	operators and	parameters associated	with a	a genetic	algorithp
--	-----------	--------------	---------------	---------------	-----------------------	--------	-----------	-----------

The choice of these operators and parameters greatly influences the search capabilities of the GA. They determine whether the algorithm can find a suitable solution and with what efficiency. However, choosing appropriate parameters for a GA is problematic and is in itself an optimisation problem Earlier work has been done on trying to find suitable choices of operators and parameters that will work well on a wide range of problems. De Jong (1975) put considerable effort into finding ideal parameter values, for a traditional GA, that were good for a suite of test problems also proposed in his work. He used an experimental approach and concluded that the following parameters gave reasonable performance:

Population size	50 - 100
One-point crossover rate	0.6
Mutation probability	0.001
Selection strategy	elitist + roulette wheel

In a further study, Grefenstette (1986) used a 'meta-GA' to optimise values for the parameters. He also used De Jong's test suite and evolved a population of 50 GA parameter sets (the same set used by De Jong). His study found parameters:

Population size	30
One-point crossover rate	0.95
Mutation probability	0.01
Selection strategy	elitist + roulette wheel

Note how the parameter settings differ in both studies. An exhaustive study by Schaffer *et al.* (1989) on a small set of numerical optimisation problems and on some functions from De Jong's test suite also arrived at different conclusions.

In these studies, an attempt was made to find an optimal and general set of parameters. However, the current state of GA research utilises a large range of operators and parameters; this makes it additionally difficult to determine the appropriate combinations to use. Moreover, parameters are not independent, making it practically impossible to try all combinations systematically. As stated by Eiben *et al.* (1999) "...*finding good parameter values for an evolutionary algorithm is a poorly structured, ill-defined, complex problem*". A more contemporary view on GAs acknowledges that general principles on parameter settings cannot be formulated *a priori* for general use (Bäck 1996; Mitchell 1996; Eiben *et al.* 1999). Moreover, optimal settings are likely to

differ over the course of a single run as was demonstrated by approaches that adapt recombination and mutation probabilities concurrently with ongoing search (Davis 1989; Bäck 1992a and 1992b; Tuson and Ross 1998). Other adaptive techniques such as adaptive recombination operators have been proposed and are further discussed in Section 3.3.

2.5 Schema Theorem and Building Block Hypothesis

The first theoretical foundation of genetic algorithms was presented by Holland (1992); this is known as the Schema Theorem and assumes a binary encoding of chromosomes. Holland (1992) introduced the concepts of *schemata* and *building blocks*, which have now dominated much of the theoretical analysis and thinking about GAs.

A schema is built by introducing a 'don't care' symbol (*) into the alphabet of genes. For example, in a binary encoding a schema is a string of characters from the alphabet $\{0, 1, *\}$. A schema represents all strings, or "hyperplanes" (subsets of the search space), which match it on all positions other than '*'. For example, a schema H1 = "*0000" is a hyperplane defined by having zeros in its last four positions. All strings with zeros in their last four positions are examples or instances of this schema. Thus schema H1 matches 2 strings and for example:

H2 = "1*1*0" matches 4 strings H3 = "0***0" matches 8 strings H4 = "***1*" matches 16 strings H5 = "****" matches 32 (2^l where *l* is the length of the string) strings

There are two properties used to describe schemata, the *order* and *defining length*. The Schema Theorem is formulated using these two properties and this terminology is also used throughout this thesis. The order of a schema H, denoted by o(H) is the number of non-* symbols in the schema. For example, the following 3 schemata, each of length 5,

H1 = "*01**" H2 = "*0*01" H1 = "01011" have the following orders:

$$o(H1) = 2$$
, $o(H2) = 3$ and $o(H3) = 5$.

The defining length of a schema H, denoted by $\delta(H)$, is the distance between the first and last non-* symbol. For example,

$$\delta$$
 (H1) = 1, δ (H2) = 3 and δ (H3) = 4.

Holland (1992) showed that the analysis of GA behaviour was far simpler if carried out in terms of schemata. He showed that a string of length l is an instance of 2^{l} schemata. In theory a population of P individuals could contain $P \cdot 2^l$ schemata but in general not all schemata will be represented, as there will be some overlap. Holland was able to demonstrate, using the k-armed bandit analogy, the result known as "implicit parallelism". This suggests that the main factor in the success of GAs is their ability to test a large number of possibilities such that a population will usefully process $O(P^3)$ schemata. However the validity of this argument has attracted many criticisms. Firstly the notion of implicit parallelism assumes a uniformly distributed population and in a GA that is true only in the initial population. Grefenstette (1991) states that in order to accurately assess how many hyperplanes are processed it is necessary to consider the dynamic distribution of samples within the population which means taking into account the fitness function and selection algorithm as part of the analysis. Secondly, the assumption that hyperplane competitions can be isolated and solved independently is incorrect owing to high fitness variances (Grefenstette and Baker 1989) and gene interactions (Reeves and Wright 1999). Thus, the fitness of a hyperplane cannot be estimated independently of those with which it interacts. Macready and Wolpert (1996) have also argued, using the 2-armed bandit analogy, that the strategy described by Holland is not an optimal one. Furthermore they also believe there is a fatal flaw in Holland's analysis and its supposed justification for GAs.

The Schema Theorem was formulated for a GA that uses fitness proportionate selection, one-point crossover and mutation. In summary (for a detailed explanation the reader is referred to Holland (1992 pp. 89-111) or Goldberg (1989a pp. 28-33)), the

Schema Theorem provides a lower bound on the expected number of instances of schema H at time t + 1, denoted as E[m(H,t+1)], as being a function of the number of instances of schema H at time t (denoted as m(H, t)), the probability of selecting schema H, and the probability of disrupting H via recombination and mutation (as given below):

$$E[m(H,t+1)] \ge m(H,t)$$
 * probability of selection * (1 – probability of disruption)

The Schema Theorem provides a lower bound because recombination and mutation may also create instances of a schema H. E[m(H,t+1)] is given as:

$$E[m(H,t+1)] \ge m(H,t) \cdot \frac{f(H,t)}{\bar{f}(t)} \cdot \left[1 - p_c \cdot \frac{\delta(H)}{l-1} - p_m \cdot o(H)\right]$$

Where:

f(H,t) represents the mean fitness of individuals that are instances of schema H at time t,

 $\bar{f}(t)$ represents the mean fitness of the population at time t,

- p_c is the recombination rate,
- p_m is the mutation rate.

The Building Block Hypothesis (Goldberg 1989a) is related to the Schema Theorem. This states that GAs work by discovering low-order schemata of high fitness (*building blocks*) and then combining them via recombination to form higher-order fitter schemata. However this still remains unproven and is an article of faith, which for some problems is easily violated. Consider the following a problem that has a global optimum (the fittest individual in the search space) HG = "11111" and a local optimum (a false peak) HL = "00000". Now consider the following schemata that have above average fitness:

H1 = "11***" H2 = "***11"
The combining of these schemata produces a schema H3 = "11*11", which might be *less fit* than H1 and H2 and furthermore might be *less fit* than schemata:

In these cases the GA may have difficulty in converging to HG since it may tend to converge to strings leading to HL. This phenomenon is called deception and is described in more detail in the next section.

2.6 Epistasis and Deception

A central problem in the theory of GAs is the characterisation of problems that are difficult for GAs to optimise. Many attempts to characterise such problems have focused on the notion of epistasis and deception.

Epistasis is the interaction of genes in a chromosome. That is, the influence of a gene on the fitness of the chromosome may depend on the values (alleles) of other genes present on the chromosome. This interaction does not just apply to genes that are grouped together on the chromosome. Interaction can occur between genes at opposite ends of the chromosome or between adjacent genes. Moreover each gene can have different interactions with each other; some genes may not interact with others on the chromosome whilst some may interact with many. Most problems contain epistasis; however, it is difficult to know a priori exactly how much epistasis exists in a problem.

The phenomenon of deception is strongly connected with the concept of epistasis and has been widely studied by Goldberg (1989a, b, c), Whitley (1991), Grefenstette (1993) and Deb and Goldberg (1993). A problem is considered to be deceptive if a combination of alleles or schemata lead the GA away from the global optimum and towards the local optimum. A deceptive problem is not always difficult for the GA to solve (GA-hard). Whitley (1991) and Deb and Goldberg (1993) presented conditions, which class problems as not being deceptive and thus GA-easy to those that are fully deceptive and thus GA-hard. A fully deceptive problem of order k exists when the maximum order of any schema is k and all relevant lower-order hyperplanes lead toward a single schema that is not the global optimum. A partially deceptive problem of order k exists when the maximum order of any schema is greater than k and all relevant hyperplanes of order less than k lead toward a single schema that is not the global optimum.

In our test-suite of benchmark problems (Section 5.3) we use L-MaxSAT problems (Mitchell, Selman and Levesque 1992; De Jong, Potter and Spears 1997) and NK landscapes (Kauffman, 1993) with tuneable epistasis and deceptive trap functions (Deb and Goldberg 1993) with tuneable deception which allows us to evaluate performance of recombination operators under a wide range of conditions.

2.7 Summary

The genetic algorithm is a type of evolutionary algorithm that uses a genetic/evolutionary metaphor. Implementations typically use fixed-length bit chromosomes to represent the genetic information, together with a population of individuals that undergo recombination and mutation in order to find optimal solutions.

The conventional GA has many components each associated with operators and parameters. The number of parameters available to use today are far greater than that of the original GA proposed by Holland (1992) and as the search capabilities of the GA are sensitive to the combination of the parameters chosen, the choice of parameters to use is difficult to decide a priori.

The Schema Theorem (Holland 1992) was the first explanation of how GAs work, which was later supported by the Building Block Hypothesis (Goldberg 1989a). A brief explanation of the underlying concepts of the Schema Theorem and Building Block Hypothesis were provided in this chapter and concepts of epistasis and deception were also introduced.

Chapter 3

Related Work

This chapter is focussed on recombination and begins with a survey of previous work on recombination operators by dividing these operators into two sets: static and adaptive. Early work on genetic algorithms has concentrated on static recombination operators, which are described in Section 3.2, and are widely accepted as a test bed for comparing other recombination operators. More recent work has concentrated on adaptive recombination operators, which are described in Section 3.3 in the hope of creating a more robust optimisation strategy. This survey of recombination operators is followed by a description of existing work that is used to understand the behaviour of static recombination operators in terms of (i) biases imposed on search and (ii) schema survival; these are provided in Sections 3.4 and 3.5 respectively. In Section 3.6 we introduce the linkage problem which is associated with static recombination operators and the encoding of the problem. This section also provides a survey of techniques that have been proposed to try to overcome this linkage problem. Finally, the increasing use of adaptive techniques has prompted the need to classify the different forms of adaptation presented by these techniques to allow future research to make a clear distinction between different adaptive techniques. We summarise all the classifications and their limitations in Section 3.7.

3.1 Introduction to Recombination

Recombination, also known as crossover, has been considered as the primary operator of a GA (Holland 1992; Goldberg 1989a) and is thought to be responsible for the generation and propagation of good solutions. More recently, there have been many studies on the role played by traditional static recombination operators compared with the role of mutation in a GA (Schaffer and Eshelman 1991, Spears 1993 and Wu, Lindsay and Riolo 1997). Recombination operators have also been classified by their usefulness in terms of generating and propagating solutions (Eshelman and Schaffer 1995). There are now many different ways of implementing recombination (Spears 1997). Some recombination operators incorporate adaptive methods and are classed as adaptive recombination operators: those that do not incorporate adaptive methods are classed as static recombination operators.

3.2 Static Recombination Operators

In this section we describe static recombination operators that have been considered for use as general operators. The three most popular recombination operators are one-point, two-point and uniform crossover. The popularity of one-point crossover is due to the foundations of genetic algorithms constructed by Holland (1992). Two-point crossover (De Jong 1975) and uniform crossover (Syswerda 1989) are also popular, as they have been shown to perform better than one-point crossover (Syswerda 1989; Eshelman, Caruana and Schaffer 1989). These three operators have been generally accepted as the test bed for comparing other recombination operators. In this thesis we use two-point and uniform crossover as a comparison with selective crossover.

3.2.1 One-point Crossover

One-point crossover occurs when parts of two chromosomes are exchanged after a randomly selected point, creating two children. The point selected is the same for both parents; shown in Figure 3.1. One-point crossover was used to define the Schema Theorem and the Building Block Hypothesis as mentioned earlier.



Figure 3.1: One-point crossover.

3.2.2 N-point Crossover

N-point crossover (De Jong 1975) is similar to one-point crossover, except that n points $(n \le l-1)$ are randomly selected and the genetic material between the n points is exchanged. Figure 3.2 shows an example of two-point crossover. Two-point crossover is very commonly used because it has been shown to be less disruptive of schemata than one-point crossover and has shown to perform better than one-point crossover (Spears and De Jong 1991a; Eshelman, Caruana and Schaffer 1989).

Parent1:	1	1	1	1	1	1	1	1	1	1	. 1	1
Parent2:	0	0	0	0	0	0	0	0	0	0	0	0
		L		L		I	L	I		I		J
Child1:	1	1	1	0	0	0	0	1	1	1	1	1
Child2	0	0	0	1	1	1	1	0	0	0	0	0

Figure 3.2: Two-point crossover.

3.2.3 Uniform Crossover

Uniform crossover (Syswerda 1989) differs greatly from one-point and two-point crossover. In uniform crossover we decide, with probability P_0 , for each gene, which parent contributes its allele to which child as shown in Figure 3.3. A mask is generated for each child using P_0 . In each mask (Mask1 and Mask2) a '1' indicates that the allele is inherited from Parent1 and a '2' indicates that the allele is inherited from Parent2. Note that Mask2 is a complement of Mask1 and vice versa. Exchange can potentially occur at

each allele and for this reason uniform crossover is considered as an *"allele-based"* recombination operator. This term is used throughout the thesis to differentiate between operators where crossover occurs at each allele and operators that exchange a block of sequential alleles as done in n-point recombination.



Figure	3	.3:	Uniform	crossover.
--------	---	-----	---------	------------

When Syswerda originally proposed this crossover, P_0 was set to 0.5. Uniform crossover has since been extended to *parameterised uniform crossover* (Spears and De Jong 1991b) where P_0 can take alternative values, such as 0.1. Once the value of P_0 has been decided it remains unchanged throughout the algorithm. Spears and De Jong (1991b) showed that uniform crossover (when P_0 is set to 0.5) is more disruptive of schemata and performed better than one-point and two-point crossover in some problems but worse on others (Syswerda 1989; Eshelman, Caruana and Schaffer 1989). The theoretical analysis by Spears and De Jong (1991b) showed that lowering values of P_0 could reduce this disruptive quality. However, these results were limited to a theoretical analysis and no empirical evidence was given to indicate the best setting for P_0 .

3.2.4 Bit-Based Simulated Crossover

Syswerda (1993) identified that uniform crossover is very similar to an operator that uses two parents for recombination and whenever these parents have a bit in common, the bit is copied to the child. All remaining bits are then randomly generated. This is similar to uniform crossover because in uniform crossover both children are guaranteed to inherit the common bits and each remaining bit has a 50% chance of coming from one parent or the other. Since the bits in each parent are different, it is the same as randomly assigning a 0 or 1 to the bit position.

From this idea Syswerda (1993) introduced bit-based simulated crossover (BSC), which produces a single child by using statistics of individual bits across the entire population. For example, a single bit position in a population (a bit column) will contain some ratio of ones and zeros. The individuals that contain these bits have some probability of being selected and these probabilities can be used to create a weighted average for the ones and zeros in each bit column. This produces a probability for each bit as to whether is should be a zero or a one in the offspring. These probabilities are then used to generate new individuals.

BSC was applied to a set of test problems and compared with one-point, twopoint, uniform crossover and mutation only. The results show that BSC was found to be competitive with other operators. A study by Eshelman and Schaffer (1993) on BSC also confirmed these results.

3.3 Adaptive Recombination Operators

In this section we provide a survey of adaptive recombination operators that have been proposed for use as general operators. A thorough description of masked crossover (Louis and Rawlins 1991) and adaptive uniform crossover (White and Oppacher 1994) is provided in Chapter 7 which analyses the similarities and differences to selective crossover.

3.3.1 Schaffer and Morishima (1987)

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. These appended bits represented the crossover positions using a binary alphabet. For example consider an encoded solution of length l; a 1 at locus l + i denotes a crossover position at locus i. The extra bitmap also underwent the same crossover as the corresponding alleles thereby evolving the crossover positions. Inferior children were discarded, through selection, along with their crossover bitmaps. Their evaluation of punctuated crossover was limited to a small set of problems and a comparison to one-point crossover, which showed that punctuated crossover performed better than or as well as one-point crossover. Their analysis of punctuated crossover was limited to observing the population distribution of number of crossover points, which demonstrated that the crossover points increased in line with the number of generations.

3.3.2 Louis and Rawlins (1991)

Louis and Rawlins (1991) proposed *masked crossover*, which is an allele-based recombination operator. It uses an extra binary mask that accompanies each chromosome to direct crossover. The parental binary masks are compared at each bit position; the differing bit values in the masks define the crossover positions. Using parental and fitness correlations, relative fitness information was translated into the binary mask to guide crossover towards local fitness increases. This operator is described in detail in Chapter 7 owing to its close relation to selective crossover. Their evaluation of masked crossover was limited to two problems of circuit design, which are not well studied. A comparison of masked crossover was made with one-point crossover and performed better than one-point crossover on most problems except those that are deceptive.

3.3.3 White and Oppacher (1994)

White and Oppacher (1994) proposed *adaptive uniform crossover* (AUX), which is an allele-based operator. For AUX 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. This operator is described in detail in Chapter 7 owing to its close relation to selective crossover. Their evaluation of AUX against uniform crossover on 23 unknown functions demonstrated that AUX performed better or equally on 19 functions and showed worse performance on the remaining 4 functions.

3.3.4 Eshelman and Schaffer (1995)

Eshelman and Schaffer use a switching mechanism to decide between two recombination operators based on how they perform. The two operators in question are *half-uniform crossover* (HUX) - a variant of uniform crossover which randomly swaps half of the differing bits) and *shuffle crossover* (SHX) - which is like one-point crossover without positional bias (see Section 3.4.1). The switching mechanism is incorporated before a GA is run, so if HUX was used initially and no global solution was found owing to premature convergence, the GA is re-started using SHX. The same process is then also applied to SHX. One drawback to this mechanism is that a global solution is not always known in which case when should we switch to the other operator? Their analysis demonstrated that the switching mechanism was better than using HUX on its own however worse than using SHX on its own.

3.3.5 Spears (1995)

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. 1-bit adaptation was compared against the sole use of two-point and uniform crossover on a set of unimodal and multimodal problems. The results demonstrated that this mechanism worked better than two-point crossover but no better than uniform crossover. Spears analysed 1-bit adaptation by monitoring the number of 1's in the operator column and the difference in the number of 1's from generation to generation. This analysis demonstrated that the number of 1's in the operator column changed more rapidly when using 1-bit adaptation as opposed to two-point or uniform crossover but did not provide any indication to suggest why 1-bit adaptation was demonstrated to be inferior to uniform crossover

3.4 Biases in Static Recombination Operators

The 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. The study of biases in recombination operators allows us to understand the behaviour of recombination operators on problems with specific characteristics.

Eshelman, Caruana and Schaffer (1989) studied biases in static recombination operators (later refined in Eshelman and Schaffer, 1995). Their motivation was to understand the explorative and exploitative behaviour of different recombination operators, and categorise them in terms of their positional bias, distributional bias and explorative power. An explanation of each is given in the following sections.

A study of biases in adaptive recombination operators is provided in Chapter 7 to identify any deleterious biases in these operators and to understand the search behaviour of these operators.

3.4.1 Positional Bias

A recombination operator has a positional bias when the creation of a new individual is dependent upon the location of the alleles in the chromosome. In other words the recombination operator is more likely to propagate adjacent genes together rather than disjoint ones. Booker (1992) showed that, of the *n*-point recombination operators, onepoint 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 parameterised crossover (Spears 1998) has no positional bias.

3.4.2 Distributional Bias

Distributional bias exists if the amount of material being exchanged, during recombination, is concentrated toward a mean value. If the distribution of the number of 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 (1992) found that the distributional bias of *n*-point recombination tends to increase as *n* increases, as the distribution becomes less and less uniform. Spears (1998)

extended the work by Eshelman *et al.* to include population homogeneity (similarities between individuals in the population). Rana (1999) empirically analysed the distributional biases of static recombination operators using Hamming distance. Both Spears and Rana confirmed the results of Booker and Eshelman *et al.*, that one-point and two-point crossover do not have distributional bias, whereas uniform 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.

3.4.3 Exploration and Exploitation

An issue that is of great concern in the GA community is the balance between exploration and exploitation. An efficient optimisation algorithm is one that uses two strategies: exploration to investigate new and unknown areas in a search space and exploitation to make use of knowledge acquired by exploration to reach better positions on the search space. Pure random search is good at exploration, but has no exploitation. Hill climbing is good at exploitation but has little exploration. Genetic algorithms combine both strategies, but recombination operators have varying degrees of exploration and exploitation (Eshelman, Caruana and Schaffer, 1989).

Exploratory power is defined to be the number of different individuals that can be created by a single application of the recombination operator. One-point crossover, for instance, can potentially create any one of 2(l-1) different individuals in a single event. Eshelman, Caruana and Schaffer (1989) assumed complete diversity when making the calculations, but population homogeneity also effects the exploratory power of a GA (Spears, 1998). The explorative power of any recombination operator increases as diversity increases. Therefore the exploratory power of one-point crossover at most is 2(y - 1), where y is the number of differing alleles in the parents. One-point crossover has very low exploratory power and two-point crossover somewhat higher, therefore the exploratory power of n-point recombination increases as n increases. For uniform crossover there are up to 2^y possible recombination events thus having the potential of reaching the maximum exploratory power.

3.5 Theoretical Static Recombination Analysis

Spears (1998) constructed a schema survival probability for n-point and uniform crossover. In this section we provide an overview of the schema survival probability for uniform crossover, as this will be used to construct a schema survival probability for selective crossover in Chapter 8. For the schema survival probability for n-point crossover the reader is referred to Spears and De Jong (1991a) and Spears (1998).

3.5.1 Framework

Spears (1998) constructed a schema survival probability for uniform crossover assuming that individuals are of a fixed length *l*. A schema or hyperplane of order *k* can be denoted by H_k . Given a binary encoding, H_k represents $2^{l\cdot k}$ possible strings where the strings match on the *k* defining positions. For example, $H_2 = \text{``**}00$ '' is second-order hyperplane that represents the four strings that contain zeros in their two positions.

3.5.2 Schema Survival Probability for Uniform Crossover

In uniform crossover alleles are exchanged between two parents with probability P_0 . Spears (1998) identified that a schema can survive in either offspring, under uniform crossover, if all k defining positions of H_k are exchanged or if all k defining positions of H_k are not exchanged. This can be described in terms of a bit mask with k ones or k zeros. Thus schema survival under uniform crossover can be represented as:

$$P_{s,uniform}(H_k, P_0) = P_0^k + (1 - P_o)^k$$
(3.1)

Note for traditional uniform crossover (Syswerda 1989) where $P_0 = 0.5$ the schema survival probability is simply $(1/2)^{k-1}$.

Spears (1998) identified that a schema can **also** survive due to population homogeneity; where parents share identical alleles. Suppose that crossover results in x of the k defining positions being exchanged, then a hyperplane survives if the parents' alleles:

- 1. Match at the x positions being exchanged.
- 2. Match at the k x positions not being exchanged.



Figure 3.4: An example of hyperplane survival - Parent1 is a member of the third-order hyperplane $H_3 = "**111"$ and Parent2 is an arbitrary string. Recombination is performed to produce two offspring.

For example consider a hyperplane $H_3 = "**111"$ in Figure 3.4 where parent1 is a member of the hyperplane and parent2 is some arbitrary string; recombination will produce two offspring. Now suppose exchange takes place at locus 3 only. H_3 would survive if the two parents have matching alleles at locus 3 or if they have matching alleles at loci 4 and 5. In either situation the hyperplane H_3 survives on one offspring or the other. The probability that the alleles will match depends on the population homogeneity and is given as follows:

$$P_{h} = P_{eq}^{x} + P_{eq}^{k-x} - P_{eq}^{k}$$
(3.2)

Where:

- P_{eq}^{x} represents the probability that the two parents will match on x alleles being exchanged.
- P_{eq}^{k-x} represents the probability that the two parents will match on the k x alleles not being exchanged.
- P_{eq}^{x} is the joint probability that both parents match on all k alleles and is subtracted.

Extending equation 3.1 to include population homogeneity would correspond to those bit masks which are not either all zeros or all ones at the defining positions of the hyperplanes. Thus for all possible bit mask *combinations* the schema survival probability for uniform crossover is finally given as:

$$P_{s}(H_{k}, P_{0}) = \sum_{x=0}^{k} {\binom{k}{x}} P_{0}^{x} (1 - P_{0})^{k-x} (P_{eq}^{x} + P_{eq}^{k-x} - P_{eq}^{k})$$
(3.3)

Spears (1998) graphed this probability for different values of P_{eq} against the defining length of schema and showed that:

- a. The survival probability is not affected by the defining length of schema and thus confirmed that uniform crossover has no positional bias. However, schema survival *is* affected by the order of a schema. The probability of a schema surviving is directly proportional to the order of a schema.
- b. As the population becomes more and more homogeneous the schema survival probability increases as is expected.

3.6 Encoding and Linkage in Genetic Algorithms

A set of genes is said to be 'linked' if they are epistatically connected (see Section 2.6) and are not separated by recombination. This is known as *genetic linkage* and is related to the encoding of the problem.

The "Building Block Hypothesis" (Goldberg, 1989) suggests that genes that are situated relatively close to each other on the chromosome are less likely to be disrupted in a canonical genetic algorithm (one that uses one point crossover). Further studies on gene positions and crossover by Eshelman, Caruana and Schaffer (1989) showed that recombination operators with high positional bias (see Section 3.4.1), such as one-point and two-point crossover, are less disruptive against adjacent genes. Holland (1992), Goldberg (1989a) and Goldberg, Korb and Deb (1989) suggest that genes, which are thought to be epistatically connected, should be encoded into the chromosome so that they are positioned near each other to prevent them from being separated by recombination operators that have high positional bias.

To produce such an encoding is difficult without knowing ahead of time which genes are important and related to each other in forming useful schemata. This is known as the 'linkage problem' (Mitchell 1996). Choosing such fixed encodings without a priori knowledge of the problem can be difficult for the GA user; how is one to decide the best encoding for one's problem and one's GA?

Many techniques that adapt the encoding, as an alternative to the use of fixed encodings, have been proposed to try and overcome the linkage problem and are outlined in the following sections.

3.6.1 Holland (1992)

Holland (1992) proposed "*inversion*", a reordering operator. Inversion works by giving each allele an index indicating its actual position in the chromosome (for evaluation purposes). Two points are then chosen in the string and the bits between them are reversed to produce a new ordering. Crossover then occurs on this new ordering, thereby producing more orderings. The purpose of reordering was to find orderings in which beneficial schemata are more likely to survive under one-point crossover. This technique has been applied in early work but did not produce any improvements in performance (Goldberg 1989a).

3.6.2 Goldberg, Korb and Deb (1989)

Goldberg, Korb and Deb (1989) proposed the "messy GA" that evolves the encoding. The messy GA (mGA) uses a variable length encoding where each allele has an index indicating its actual position in the chromosome, but all loci do not have to be specified in the chromosome and loci can be specified more than once. The mGA has two phases; a primordial phase – where building blocks of a particular order are generated and a juxtapositional phase – where building blocks are recombined using cut and splice operators that mimic one-point crossover. To use mGAs we are faced with the same problem of not having a priori knowledge of the problem. What is a useful schema order for the primordial phase? Also when evaluating strings where all loci are not specified,

how can you compute the true fitness? If the individual contains loci that are interacting, missing alleles are crucial in defining fitnesses for individuals.

3.6.3 Kargupta (1996)

Further extensions have been made to the mGA called the "gene expression messy GA" (GEMGA) (Kargupta 1996) where each gene has a position, value, weight and a linkage set. Kargupta demonstrated that GEMGA has better performance than the mGA but it also requires an optimal schema order in the initial population.

3.6.4 Harik (1997)

The "linkage learning GA" (LLGA) developed by Harik (1997) used alleles that were also tagged with their actual positions. An exchange operator similar to two-point was used for recombination. In his study he compared LLGA on problems constituted by a number of non-overlapping building blocks of a maximum size k and on uniformly-scaled problems (problems where all building blocks give the same contribution to the fitness e.g. the one-max problem). Unfortunately the LLGA did not work well for easy uniformly scaled problems and the study was limited to non-overlapping building blocks.

3.6.5 Smith (1998)

Smith (1998) proposed the "*LEGO*" operator that worked by considering the population of solutions as a gene pool comprised of 'blocks' of genes defined over certain loci. These blocks can vary in size from a single gene to an entire chromosome. This was achieved by associating two boolean flags to each gene that determine whether it links to the genes to its left and right. Two adjacent genes are linked if the appropriate flags are set to *true*. A new individual is created by holding tournaments to fill its loci. The LEGO operator did not perform any better than static recombination operators on simple optimisation problems like the one-max problem.

3.7 Classification of Parameter Adaptation in Genetic Algorithms

The run of a GA is intrinsically a dynamic and adaptive process but the use of constant parameters or static operators contradict the general evolutionary spirit and thus elicits sub-optimal performance from the GA. To overcome this contemporary GAs use adaptive strategies that modify parameters or manipulate operators during the run. The increased use of adaptive strategies has prompted the need for classifying different types of adaptation.

Angeline (1995) provided a classification based on levels of adaptation and types of update rules used. He defined three levels adaptation that can exist within the GA – population, individual and component level of adaptation. Population level adaptation occurs when parameters global to the population are adjusted dynamically. Individual level adaptation adjusts strategy parameters held within individuals and the values undertaken by these parameters affect only the corresponding individuals. Component level adaptation adjusts strategy parameters local to some component of an individual.

Angeline also defined two distinct types of update rules for parameters in an adaptive evolutionary algorithm – absolute and empirical update rules. Absolute rules are predetermined and specify how modifications are made. In contrast empirical update rules allow the competitive process in evolutionary algorithms to determine if changes in parameters are advantageous. A *self-adaptive* evolutionary algorithm was defined as one that evolves the values of its adaptive parameters. Angeline's classification considered evolutionary algorithms as a whole with little attention given to the different components that exist and can be adapted, for example, in a genetic algorithm.

Hinterding *et al.* (1997) extended the classification provided by Angeline by defining types of adaptation that are possible from the update mechanism used and introduced an extra level of adaptation. They define two general types of adaptation, static and dynamic adaptation. A canonical GA is adaptive but exhibits static adaptation because the operators are controlled externally and are decided before a run. Dynamic adaptation occurs when some mechanism, other than an external source, modifies the functionality of the operators. An example of dynamic adaptation is a self-adaptive GA, which has encoded on the chromosome the parameters that are to be adapted and these parameters will undergo recombination and mutation along with the chromosome.

52

7

Dynamic adaptation was further sub-divided into three classes: deterministic, adaptive and self-adaptive. Deterministic dynamic adaptation was defined to take place if the value of the parameter is altered by some deterministic rule that does not use any feedback from the evolutionary algorithm e.g. changing the probability of mutation at specified generations. Adaptive dynamic adaptation exists if feedback from the evolutionary algorithm is used to determine the direction and/or magnitude of the change to the strategy. Self-adaptive adaptation is described as done by Angeline, where parameters evolve along with the evolutionary algorithm. The adaptive parameters are encoded into the chromosome of the individual and also undergo mutation and recombination; however, these encoded parameters do not effect the individual's fitness. Hinterding *et al.* also introduced environment level adaptation in addition to population and individual level adaptation, where the response of the environment (in terms of the fitness function) to the individual is changed.

This classification provided by Hinterding *et al.* does not consider adaptation in different components of the evolutionary algorithm. Also this classification of different types of adaptation is not as clear-cut. For example, some of the adaptive recombination operators described earlier fall in more than one category. Some features are adaptive e.g. the credit assignment mechanism used. On the other hand some are self-adaptive e.g. the encodings used that also evolve along with the chromosomes.

Smith (1998) provides a comprehensive classification based on three criteria: *what* is being adapted, the scope of adaptation and the basis for change. Smith's classification captures the different components that exist in an evolutionary algorithm by analysing what is actually being changed (e.g. encoding, recombination operators, mutation or recombination rates, selection etc). This classification allows us to locate where a specific adaptive mechanism takes effect. The scope of adaptation was described using Angeline's terminology on the three levels of adaptation: population, individual and component level. The third criterion, the basis of change, was sub-divided into two categories: the evidence on which the change is based (e.g. population statistics), and the rules that define how changes are affected.

A recent study by Eiben *et al.* (1999) merges some elements of the classification provided by Hinterding *et al.* and Smith. Their classification is more general and allows more methods of adaptation to be classified. They identified two main criteria:

- What is changed (Smith 1998) This captures the type of component or control in the evolutionary algorithm that is undergoing change.
- 2. How the change is made (Hinterding *et al.* 1997) This is dictated by three categories: deterministic, adaptive and self-adaptive.

Their classification is more general and applicable to all evolutionary algorithms. It does not include the scope or level of adaptation because it is not always intuitive to decide whether a parameter acts at the individual level or the population level; for example, in Spears' (1995) adaptive operator, 1-bit adaptation, where an extra bit was used to determine which recombination operator to use. This technique can be characterised as adaptation at the individual level. However, considering the distribution of this bit over the entire population the adaptation is also at the population level. Therefore there is no precise distinction between the scope or level of adaptation.

Eiben *et al.* defines that the type of change (the how-aspect) can be classified into one of three categories: deterministic, adaptive and self-adaptive. However, the analysis of selective crossover in Chapter 6 demonstrates that adaptive strategies can employ more than one method of change and thus can not be uniquely classified into one category. Chapter 6 will provide a new taxonomy for the three categories of change, which allows many diverse adaptive strategies to be classified.

3.8 Summary

Recombination is a distinct operator associated with GAs and one-point crossover was the first recombination operator proposed for use in GAs. Since then many recombination operators have been proposed. Early work in GAs concentrated on static recombination operators and analyses in terms of biases and schema survival have been done to understand the behaviour of these operators. More recently, adaptive recombination operators have been proposed in the hope of overcoming some limitations of static recombination operators and creating a more robust optimisation strategy. These adaptive operators have been evaluated in terms of performance in comparison with other techniques; this is usually done using a test suite of problems or a real world application. However, little effort has been given to analyse "how?" and "what?" provides this increased performance. This form of analysis has been limited to two studies: (i) punctuated crossover (Schaffer and Morishima 1987), which analysed the distribution of crossover locations in the population, (ii) 1-bit adaptation (Spears 1995) which monitored the operator choice encoded within individuals. The lack of analyses limits our understanding of these new adaptive operators, and provides us with little justification for their use on other problems. For these reasons we have conducted four different evaluations of selective crossover; each evaluation is given in Chapters 5, 6, 7 and 8.

Many linkage learning GAs have also been proposed to overcome the linkage problem associated with some static recombination operators. However these techniques either showed very little improvement or required specific knowledge about the problem, which in some cases, as stated earlier, is difficult to obtain.

Owing to a recent increase in adaptive strategies many classifications have been proposed to allow future research to make a clear distinction between different techniques. An overview of each classification and its limitations was provided. The classification provided by Eiben *et al.* (1999) is more general and allows more methods to be classified than those provided by others. However their classification also has limitations, which are overcome in Chapter 6 by providing a new taxonomy to classify adaptive strategies.

Chapter 4

Selective Crossover

There are now many different methods for recombination (Spears 1997) but for a given problem, two-point and uniform crossover are generally those that are commonly used. Even with just three crossover operators it has been difficult to decide a priori which form of recombination operator is more efficient to use as the synergy of these operators, other GA parameters and the problem be solved does not always yield optimum performance. We know from "No Free Lunch" theorems (Wolpert, and Macready 1997) that an algorithm that is suited for all problems cannot exist, since for fixed parameter/operator sets there will be problems for which they are optimal and other problems for which their performance is poor. However, is it possible to devise an adaptive recombination operator that dynamically adapts to problem characteristics to use for a *wide range* of problems in which hitle is known about the problem space being searched? This is our motivation for designing "selective crossover" an adaptive recombination operator.

This chapter is based on work in Vekaria and Clack (1998a, 1998b, 1999b, 2000). It provides a detailed description of selective crossover and emphasises its key properties. A description of how this operator is incorporated into the GA is also presented.

4.1 Inspiration

The inspiration for a new adaptive recombination operator ("selective crossover") comes from nature, specifically Dawkins' model of evolution and dominance characteristics in nature. Dawkins' (1989) model of evolution is based on the gene. In his theory the gene is considered to be the fundamental unit of natural selection. Since natural selection acts on the individual and the individual consists of unique chromosomes, these chromosomes actually have a life span of one generation. However, the chromosome consists of many genes and a subset of these genes last for many more generations because it is the genes that are passed onto the offspring not the entire chromosome. Thus, natural selection actually favours the gene.

Dominance in nature is associated with genetic material represented using diploid chromosomes. In the diploid form a genotype carries one or more pairs of chromosomes, each containing information for the same functions. The alleles contained in one chromosome can be regarded as a direct alternative to the alleles in the other chromosome. When building the organism the alleles in one set compete with those in the other set. Alleles that are dominant are expressed in the phenotype of an organism and those that are less likely to be expressed are recessive. The relationship between a dominant and recessive gene is complex: some genes that have been known to be dominant have become more recessive in successive generations and vice versa. These dominance characteristics have evolved over generations. Merrell (1984) suggests that these shifts in dominance are in response to changes in the environment. Thus, those genes that increased an individual's fitness have become more dominant; however, a precise model to show this is not available.

Selective crossover was designed using both the analogy of dominance, where alleles in a chromosome compete with those on the other chromosome, and the analogy of evolution of dominance. The aim is to see if recombination of genes in a haploid GA can be evolved where alleles in one parent compete with those on the other parent chosen for crossover. Here the alleles are competing to be retained in a fitter individual and the use of correlations between parental and offspring fitnesses would allow the means of discovering beneficial alleles.

Selective crossover is very much like "dominance without diploidy". It uses an extra vector that accompanies the chromosome to accumulate knowledge of what happened in previous generations and uses this memory to bias and combine successful alleles (individual bits) during recombination onto the next generation.

57

4.2 Terminology

- 1. Genes are considered here as single bits with associated positions called *loci*.
- An allele is the value taken up by the gene. A bit representation is used throughout this thesis thus the alphabet defined for an allele is given as the set {0,1}.
- 3. The *chromosome* is also referred to here as a *gene vector*.
- 4. The extra real valued vector used by selective crossover has been named a *dominance vector* and each locus has a *dominance value*. "Dominance" is not really an appropriate word to use because in natural genetics (as mentioned in the previous section) dominance is associated with diploid organisms and the research presented here is on a haploid genetic algorithm. However our use of dominance is to indicate that *an allele x 'dominates' an allele y during recombination*.

4.3 Key Properties

Selective crossover is designed with three interdependent key properties:

- 1. **Detection** It detects alleles that were changed during recombination to identify modifications to the candidate solution.
- 2. Correlation It uses correlations between parental and offspring fitnesses as a means of discovering beneficial alleles.
- 3. **Preservation** It preferentially preserves alleles at each locus, during recombination, according to their previous contributions to beneficial changes in fitness.

These three properties work together to form selective crossover. Correlations between parents and offspring together with the detection of alleles (inheritance of alleles) are used to update the dominance values. The dominance values in turn dictate the inheritance and preservation of allele combinations.

4.4 Implementation

Selective crossover (Vekaria and Clack, 1998a, 1998b) is an adaptive recombination operator that evolves better individuals by using a dominance vector to bias alleles during recombination. Each chromosome is accompanied by a dominance vector, which consists of continuous real values such that each allele in the chromosome has an associated dominance value. We assume that dominance in nature is not bounded by limits and therefore the dominance values here are also unbounded and can potentially take up any positive real number. A dominance value is used to accumulate the fitness contribution of an allele with respect to the fitness of the entire individual and parental fitnesses. Hence, the dominance vector accumulates knowledge of what happened in previous generations and uses that to bias successful or beneficial alleles during recombination, by preventing them from crossing-over.

To incorporate selective crossover into the genetic algorithm, the cyclic process of the genetic algorithm shown in Figure 2.1 changes to the one depicted in Figure 4.1.



Figure 4.1: A Generational Cycle of a Genetic Algorithm using Selective Crossover

Population

As mentioned earlier each member of the population must have an associated dominance vector; an example of an individual is shown in Figure 4.2. This dominance vector will undergo the processes of recombination and selection along with the gene vector (chromosome) of the individual. When the population is initialised the dominance values are also randomly initialised with values that are initially constrained to lie in the range [0,1] but eventually can take on any positive real number.

Dominance vector	0.4	0.3	0.01	0.9	0.1	0.2
Gene vector (chromosome)	1	0	0	1	0	0

Figure 4.2: An individual in selective crossover consists of an additional dominance vector.

Recombination

Recombination with selective crossover uses two parents to create two children and the recombination rate (see Section 2.3.4) decides how many individuals in the population will be subjected to recombination. Having selected two parents, 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 Child 1 *along with the associated dominance value* and Child 2 inherits the allele with the lower dominance value. If both dominance values are equal then crossover does not occur at that position. The loci where exchange occurred, such that the alleles differ, are also recorded. Figure 4.3 gives an example of selective crossover: the shaded genes have a higher dominance value than its competing gene.

Mutation

Mutation acts on the gene vector as described in Section 2.3.5. If an allele is mutated no change is made to the corresponding dominance value.

Parent 1 - fitness = 0.36

0.4	0.3	0.01	0.9	0.1	0.2
1	0	0	1	0	0

Parent 2 - fitness = 0.30

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

Child 1

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

Child 2

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

Child 1 - fitness = 0.46

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

Child 2 - fitness = 0.20

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

Increase	dominance	values

Child 1 - fitness = 0.46

0.4	0.3	0.5	0.9	1.0	0.3
1	0	1 · ·	1	1 .	0

Child 2 -fitness = 0.20

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

Figure	4.3:	Recombination	with	Selective
Crossov	ver			

Figure 4.4: Updating Dominance Values

Evaluation

After recombination the new individuals in the population are evaluated. The dominance vectors do not contribute to an individuals' fitness and thus are not used to evaluate an individual. The fitness of each individual is calculated as described in Section 2.3.2.

Updating Dominance Values

Having assigned fitness values to the new individuals during evaluation, their dominance values now need to be updated. For each individual (child) that is now

a result of recombination, we compare its fitness to both parents' fitnesses. If the child's fitness is greater than the fitness of either parent, the dominance values (of only those genes that were exchanged during crossover) are increased proportionately to the fitness increase. This is done to reflect the alleles' contribution to the fitness increase (as shown in Equation 4.4 in the next section). Figure 4.4 gives an example using fitness values, which follows on from the selective crossover example given in Figure 4.3. In Figure 4.4, only Child 1 has an increase in fitness of 0.1 (compared with the fittest parent) hence its dominance values get updated. In Figure 4.3 the bit values of Parent 1 and Parent 2 at loci 1 and 2 did not get exchanged during crossover and the bit values at loci 4 and 6 are the same. Thus, after selective crossover, the genes that caused a change in the chromosome are only those held at loci 3 and 5. Since the change of those genes at loci 3 and 5 resulted in an increase in fitness, only their dominance values get increased by 0.1 in Child 1 (shaded in Figure 4.4).

Selection

Selection is achieved as described in Section 2.3.3. When an individual is selected both the gene vector and associated dominance vector is copied to the new population. Those individuals that do not get selected are discarded along with their dominance vector.

4.5 Mathematical Representation

Let us consider an l bit representation and let $\Omega = \{0,1\}^l$ be the search space. Each individual in selective crossover has a gene vector G and a dominance vector D.

$$G = (g_1, ..., g_i, ..., g_l) \text{ where } g_i \in \{0, 1\}$$
$$D = (d_1, ..., d_i, ..., d_l) \text{ where } d_i \in \Re^+$$

Recombination occurs between two parents $(G^{P_1} \text{ and } G^{P_2})$ and produces two children $(G^{C_1} \text{ and } G^{C_2})$.

$$G^{A_{k}} = \left(g_{1}^{A_{k}}, \dots, g_{i}^{A_{k}}, \dots, g_{l}^{A_{k}}\right) \text{ where } g_{i}^{A_{k}} \in \{0, 1\}$$

$$D^{A_{k}} = \left(d_{1}^{A_{k}}, \dots, d_{i}^{A_{k}}, \dots, d_{l}^{A_{k}}\right) \text{ where } d_{i}^{A_{k}} \in \Re^{+}$$

$$A \in \{P, C\} \text{ and } k \in \{1, 2\}$$

P indicates a parent vector and C indicates a child vector. k refers to the parent or child number (either 1 or 2).

Two parents are randomly chosen from the population for recombination, Parent 1 (P_1) and Parent 2 (P_2), with gene vectors G^{P_1} and G^{P_2} respectively. Their corresponding dominance vectors are D^{P_1} and D^{P_2} . A crossover can be represented by inheritance masks M^1 and M^2 .

$$M^{k} = (m_{1}^{k}, \dots, m_{i}^{k}, \dots, m_{i}^{k}) \text{ where } m_{i}^{k} \in \{0, 1\} \text{ and } k \in \{1, 2\}.$$

Crossover will produce two children Child 1 (C_1) and Child 2 (C_1), with gene vectors G^{C_1} and G^{C_2} respectively. The inheritance mask for Child 1 is M^1 and is given in Equation (4.1). The inheritance mask is created by comparing the parent dominance vectors. In simple terms, each element m_i^1 in M^1 is 1 if the element $d_i^{P_1} \ge d_i^{P_2}$ in the parent dominance vectors. The inheritance mask M^2 of Child 2 is a complement of M^1 and can also be given by $M^2 = (1, ..., 1) - M^1$.

$$M^{1} = S\left(S\left(D^{P_{1}} - D^{P_{2}}\right) + (1,...,1)\right)$$
(4.1)

Where *S* is a sign function operating component-wise as defined below:

Sign Function

Given vectors M and N of length *l* where:

$$M = (m_1, \dots, m_i, \dots, m_l) \text{ where } m_i \in \Re$$
$$N = (n_1, \dots, n_i, \dots, n_l) \text{ where } n_i \in \{0, 1\}$$

$$S(M) \rightarrow N \mid \forall i, n_i = \begin{cases} 1, & if \quad m_i > 0\\ 0, & if \quad m_i = 0\\ -1, & otherwise \end{cases}$$

Using the inheritance mask the resulting gene vector for Child k (where $k \in \{1,2\}$), after recombination, is therefore:

$$G^{C_{k}'} = M^{k'} \cdot (G^{P_{1}} - G^{P_{2}}) + G^{P_{2}'}$$
(4.2)

Where X' represents a transposition of vector X.

The dominance vectors of each new Child k are updated to reflect fitness increase \mathcal{FS}^k with respect to its parents. We only update the dominance values of those genes that were changed during crossover, so we take the Hamming distance of the parent and child to update the appropriate dominance values in the dominance vector D^{C_k} . The Hamming distance is computed by the exclusive-OR operator \oplus . Therefore the resulting dominance vector for Child k is:

$$D^{C_{k}'} = M^{k'} \cdot (D^{P_{1}} - D^{P_{2}}) + D^{P_{2}'} + \mathcal{F}^{k} \cdot (G^{C_{k}} \oplus G^{P_{k}})'$$
(4.3)

Where:

$$\mathcal{FS}^{k} = \begin{cases} \phi(G^{C_{k}}) - \phi(G^{P_{k}}) &, \text{ if } \phi(G^{C_{k}}) > \phi(G^{P_{k}}) \\ \max\left(0, \left(\phi(G^{C_{k}}) - \phi(G^{P_{2}})\right)\right) &, \text{ if } k = 1 \\ \max\left(0, \left(\phi(G^{C_{k}}) - \phi(G^{P_{1}})\right)\right) &, \text{ if } k = 2 \end{cases}$$

(4.4)

The fitness function is $\phi(G)$ and \mathcal{FS}^k is the fitness increase, which is computed by taking the difference between fitness values of Child k and its corresponding Parents. If there is no fitness increase, the dominance values are not updated (i.e. $\mathcal{FS}^k = 0$). To obtain the fitness increase the child is first compared with the fitness of its corresponding parent (i.e. Child 2 and Parent 2). If there is no fitness increase then the child is compared with the other parent. For example, Child 2 is first compared with Parent 2; if there is no fitness increase it is then compared with Parent 1 and the fitness increase (if any) is used to update the dominance values.

4.6 **Review of Properties**

In this section we provide a review of the properties of selective crossover stated earlier (detection, correlation and preservation) and their presence in the representation given in the previous section.

The first property, *detection*, is fulfilled in Equation 4.3 where the Hamming distance is taken of the Child and its corresponding parent. This detects alleles that were changed during recombination and thus identifies modifications made to the candidate solution.

The second property, *correlation*, is a combination of Equation 4.4 and 4.3. Using the fitness differences between the new offspring and the parents (Equation 4.4) the dominance values are updated to reflect fitness increase and thus potentially beneficial alleles (Equation 4.3). Therefore correlations between parental and offspring fitnesses are used to discover beneficial alleles.

The third property, *preservation*, where alleles are preferentially preserved at each locus according to their previous contributions to beneficial changes in fitness is fulfilled in the construction of inheritance masks using the dominance values (Equation 4.1) and the construction of the new genotypes for the offspring (Equation 4.2). The dominance values hold knowledge about the previous contribution of alleles, as explained above in the second property, it is this knowledge that dictates where crossover will occur. Therefore, high dominance values restrict crossover to occur at corresponding loci.

4.7 Examples of Recombination

In this section a worked example is provided that examines the different combinations of parental fitnesses. This example shows how the dominance values are updated. We start with two parents chosen at random to undergo recombination:

D^{P_1} Dominance vector for Parent 1	2.4	0.2	1.01	0.9	0.1	0.6
G^{P_1} Gene vector for Parent 1	1	0	0	1	0	1

D^{P_2} Dominance vector for Parent 2	0.01	0.2
G^{P_2} Gene vector for Parent 2	0	1

2	0.01	0.2	2.4	0.2	1.3	0.9
	0	1	1	1	0	0

We get inheritance mask M^1 from Equation 4.1 and M^2 is simply an inverse of M^1 . For example:

$$M^{1} = S(S(D^{P_{1}} - D^{P_{2}}) + (1,...,1))$$

= $S(S(2.39 \ 0 \ -1.39 \ 0.7 \ -1.2 \ -0.3) + (1,...,1))$
= $S((1 \ 0 \ -1 \ 1 \ -1 \ -1) + (1,...,1))$
= $S(2 \ 1 \ 0 \ 2 \ 0 \ 0)$
= $(1 \ 1 \ 0 \ 1 \ 0 \ 0)$

M^1 Inheritance mask for Child 1	1	1	0	1	0	0
M^2 Inheritance mask for Child 2	0	0	1	0	1	1

Using the inheritance masks the resulting gene vectors for the offspring are given using Equation 4.2. For example:

$$G^{C_1'} = M^{1'} \cdot (G^{P_1} - G^{P_2}) + G^{P_2'}$$

= $M^{1'} \cdot (1 - 1 - 1 \ 0 \ 0 \ 1) + G^{P_2'}$

$$= \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 1 \end{pmatrix} + G^{P_2'}$$

$$= \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

The resulting dominance vector, after applying selective crossover, is given by using the first part of Equation 4.3. For example:

$$D^{C_{1}'} = M^{1'} \cdot (D^{P_{1}} - D^{P_{2}}) + D^{P_{2}'}$$

$$= M^{1'} \cdot (2.39 \quad 0 \quad -1.39 \quad 0.7 \quad -1.2 \quad -0.3) + D^{P_{2}'}$$

$$= \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \cdot (2.39 \quad 0 \quad -1.39 \quad 0.7 \quad -1.2 \quad -0.3) + D^{P_{2}'}$$

$$= \begin{pmatrix} 2.39 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.01 \\ 0.2 \\ 2.4 \\ 0.2 \\ 1.3 \\ 0.9 \end{pmatrix} = \begin{pmatrix} 2.4 \\ 0.2 \\ 2.4 \\ 0.9 \\ 1.3 \\ 0.9 \end{pmatrix}$$

 G^{C_2} and D^{C_2} are calculated similarly, giving two new individuals:

- D^{P_1} Dominance vector for Child 1
- G^{P_1} Gene vector for Child 1

2.4	0.2	2.4	0.9	1.3	0.9
1	0	1	1	0	0

- D^{P_2} Dominance vector for Child 2
- G^{P_2} Gene vector for Child 2

0.01	0.2	1.01	0.2	0.1	0.6
0	1	0	1	0	1

After recombination the fitness of each offspring is calculated. If the offspring shows a fitness increase, the Hamming distance of the children and their parents is used to update the dominance values. For each child the Hamming distance is required to ensure the correct alleles' (only alleles that were changed during recombination) dominance value gets updated. For example:

 $G^{C_1} \oplus G^{P_1} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \oplus \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$ $= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$

The Hamming distances, in this example, are therefore:



The updating of the dominance values is dependent on any fitness increase a child has with respect to the parents. For each child there are four possible fitness combinations that can occur. Using Child 1 (C_1) as an example, where P_1 and P_2 stands for Parent 1 and Parent 2 respectively, we have:

- 1. Fitness of C_1 > Fitness of P_1 and Fitness of C_1 > Fitness of P_2
- 2. Fitness of C_1 > Fitness of P_1 and Fitness of $C_1 \le$ Fitness of P_2
- 3. Fitness of $C_1 \leq$ Fitness of P_1 and Fitness of $C_1 >$ Fitness of P_2
- 4. Fitness of $C_1 \leq$ Fitness of P_1 and Fitness of $C_1 \leq$ Fitness of P_2

Case 1: Fitness of C_1 > Fitness of P_1 and Fitness of C_1 > Fitness of P_2

Given that Child 1 has a higher fitness than Parent 1 by a difference of a and it also has a higher fitness than Parent 2. Child 1 differs from Parent 1 at loci 3 and 6. This indicates that the change at loci 3 and 6 increased the fitness of that individual. Therefore, from Equation 4.3 the dominance values at loci 3 and 6 are increased by a. In this case the dominance values of Child 1 are increased using Equation 4.3 as follows:

$$D^{C_{k}'} = M^{k'} \cdot (D^{P_{1}} - D^{P_{2}}) + D^{P_{2}'} + \mathcal{F} \mathcal{Y}^{k} \cdot (G^{C_{k}} \oplus G^{P_{k}})'$$

$$= \begin{pmatrix} 2.4 \\ 0.2 \\ 2.4 \\ 0.9 \\ 1.3 \\ 0.9 \end{pmatrix} + a \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 2.4 \\ 0.2 \\ 2.4 + a \\ 0.9 \\ 1.3 \\ 0.9 + a \end{pmatrix}$$

The resulting individual is given below:

 D^{C_1} Dominance vector for Child 1 G^{C_1} Gene vector for Child 1

2.4	0.2	2.4+ <i>a</i>	0.9	1.3	0.9+ <i>a</i>
1	0	1	1	0	0

Case 2: Fitness of C_1 > Fitness of P_1 and Fitness of $C_1 \leq$ Fitness of P_2

In this case, Child 1 has a higher fitness than Parent 1 by a difference of a but its fitness is lower than Parent 2. This also indicates that the change at loci 3 and 6 was beneficial and therefore the dominance values of Child 1 are increased as in Case 1.

Case 3: Fitness of $C_1 \leq$ Fitness of P_1 and Fitness of $C_1 >$ Fitness of P_2

In this case, Child 1 has a lower or equal fitness than Parent 1 but it has a higher fitness

than Parent 2 by a difference of b. Here the change that occurred at loci 3 and 6 reduced the fitness. However the individual is still fitter than Parent 2. This provides some indication that there may be epistasis (and possibly some deception) amongst the genes. To reflect this possibility of interaction, the dominance value at loci 3 and 6 belonging to Child 1 are updated by the fitness difference b (difference related to Parent 2). Therefore, from Equation 4.4 the fitness increase that is used is b. In this case the dominance values of Child 1 are increased as follows:

 D^{C_1} Dominance vector for Child 1 G^{C_1} Gene vector for Child 1

2.4	0.2	2.4+ <i>b</i>	0.9	1.3	0.9+ <i>b</i>
1	0	1	1	0	0

Case 4: Fitness of $C_1 \leq$ Fitness of P_1 and Fitness of $C_1 \leq$ Fitness of P_2

In case 4, Child 1 has a lower or equal fitness than Parent 1 and Parent 2, which indicates that the change at locus 3 was undesirable. Therefore the dominance values of Child 1 do not get updated and remain as follows:

 D^{C_1} Dominance vector for Parent 1 G^{C_1} Gene vector for Parent 1

2.4	0.2	2.4	0.9	1.3	0.9
1	0	1	1	0	0

4.8 Review of Strategies Used

This section explains the reasoning behind the strategies used: creation and retaining of Child 2, increasing but not decreasing dominance values, recombination at each gene rather than blocks of adjacent genes and not updating dominance values as a result of mutation.

1. Child 2 inherits the alleles that have a lower dominance value. Child 2 is needed to keep diversity in the population so that genetic diversity is not lost in early generations when more exploration is required than exploitation. That way if Child 2 was to produce an increase in fitness to that of its parents, then its genes will get preserved during recombination. Selection will bias the fitter individuals by retaining them in the population and losing the least fit and their dominance vectors.

- 2. The dominance values get increased when the fitness increases, so it follows that one should decrease the dominance values when fitness decreases. We chose not to do this because we prefer not to introduce a strong negative bias during the early (highly explorative) generations. If we were to decrease the dominance values the corresponding alleles may never get positively biased during recombination, hence introducing a negative bias in early generations; this form of bias is left for selection. By not decreasing the dominance values we still allow the alleles to compete with other alleles (at the same locus in the population). This is further justified in Chapter 7 by the analysis of biases in selective crossover.
- 3. Unlike one-point or two-point crossover, selective crossover is not biased against schemata with high defining length, which is later demonstrated in Chapter 8. Selective crossover propagates good schemata regardless of their defining length for example, if a schema consists of interacting genes at the two extremes of the chromosome, it can be propagated as easily as a schema which consists of interacting genes located adjacent to each other. Selective crossover can be considered as an extension of uniform crossover. With selective crossover the probability of crossing over at a position is dependent on what happened in previous generations whilst in uniform crossover the probability it fixed throughout (traditionally at 0.5). For this reason selective crossover is classed as an allele-based adaptive recombination operator (crossover positions are determined at each allele).
- 4. When an allele is mutated no change is made to the corresponding dominance value because this study is limited to investigating the effects of updating dominance values due to recombination.

4.9 Summary and Conclusions

This chapter described a new adaptive crossover operator, selective crossover, for use with genetic search. Its design was motivated by intuition abstracted from Dawkins' theory of natural evolution to exploit and express good characteristics. In retrospect we
found that this strategy is similar to one that exists in natural genetics and is termed as "assortative segregation" (Altenberg 1998). In assortative segregation, recombination shuffles together the genetic material by separating and then combining advantageous genetic material (Bergman, Otto and Feldman, 1995a; Bergman, Otto and Feldman, 1995b).

Selective crossover is an allele-based adaptive recombination operator, which is similar to uniform crossover in that crossover occurs at each gene position, rather than at blocks of genes adjacent to each other. However, selective crossover has three distinctive features that make it different from uniform crossover: detection of alleles that were changed during recombination, use of correlations to discover and preserve beneficial alleles during recombination.

Selective crossover evolves better individuals by using a dominance vector to bias (prevent alleles from crossing-over) alleles that have increased an individual's fitness in previous generations. It uses this vector as a means of storing knowledge about what happened in previous generations and exploiting this knowledge during recombination. A dominance vector accompanies each chromosome such that each allele in the chromosome has an associated dominance value. It is these dominance values that dictate where crossover should occur.

The next chapter evaluates selective crossover in terms of performance and compares it with two-point and uniform crossover.

72

Chapter 5

Performance Evaluation of Selective Crossover

In the previous chapter we provided a detailed description of selective crossover; in this chapter we conduct the first evaluation of selective crossover (part of this work is published in Vekaria and Clack, 1998b). As a first step in verifying the hypothesis of this thesis, we empirically evaluate selective crossover by comparing its performance with that of two-point and uniform crossover on five well-studied benchmark problems: One Max, Royal Road, L-MaxSAT, NK Landscapes and Deceptive Trap problems. Each of these problems have unique characteristics of epistasis, deception and fitness landscapes that together provide a good variation of features present in real-world problems to evaluate selective crossover. The One Max problem represents simple optimisation problems, the Royal Road represents problems containing tightly coupled schemata, the L-MaxSAT represents problems with variable epistasis, the NK landscapes and the Deceptive Trap represents problems with variable deception.

5.1 Experimental Method

In this section we describe the experimental conditions used to evaluate selective crossover. These conditions have been applied to all experiments conducted in this thesis and are described below:

• The performance measure is the number of evaluations taken to find the global solution or the number of evaluations taken to find the best solution after x

generations. The choice of this performance is because the vast majority of computation involved in a GA is during the evaluation of individuals. All individuals in the population are evaluated in the initial generation. However, in subsequent generations some individuals may not be subjected to recombination or mutation owing to the recombination rate (the probability that individuals are recombined) and the mutation rate (probability that a gene will be mutated), as mentioned in Sections 2.3.4 and 2.3.5 respectively. Thus, the GA only re-evaluates individuals that underwent crossover or mutation.

- Selective crossover is compared against two-point and uniform crossover because these two operators have been generally accepted as a test bed for comparing other recombination operators (Syswerda 1993; White and Oppacher 1994; Spears 1995). Moreover, two-point crossover and uniform crossover have been shown to perform better than one-point crossover (Syswerda 1989; Eshelman, Caruana and Schaffer 1989). To make this a strict comparison of recombination operators all other GA operators and parameters are kept constant.
- All three recombination operators are applied to the same problems. Since it is
 impossible to make a comparison on all problems we have selected five well
 studied and different benchmark problems that have been previously used to
 study GA behaviour and which also have characteristics present in practical
 problems.
- The comparison of the recombination operators is done across 50 independent runs, which is our sample size. A larger sample size (more runs) would provide us with slightly more precision in our comparisons, but the trade off would be the extensive use of computational resources. A small sample size on the other hand, would provide us with less precision. Hence the choice of 50 was made because it provides a reasonable sample size from which a fair comparison can be made (Rice 1995) and also saves computational resources.
- For all 50 independent runs the same set of seeds (for the random number generator) are used for each of the recombination operators being tested under

the assumption that the initial population of each GA run will have a significant effect on the population.

Statistical hypothesis testing is used to compare the performance of selective crossover with two-point and uniform crossover. Statistical hypothesis testing is done using a *t*-test. Given our choice of the sample size, we do not know the variance of the population and it has to be estimated. The *t*-test allows for uncertainty introduced by estimating the variance of the population means and can be used with our sample size. A *t*-test is carried out at a 5% significance level. The significance level reflects residual uncertainty in our conclusions (Cohen 1995). The *null hypothesis* H₀ and an *alternative hypothesis* H₁ are used to compare selective crossover with two-point and uniform crossover and are defined below:

H₀: $\mu_s = \mu_A$ (of equal performance) H₁: $\mu_s < \mu_A$ (selective crossover shows better performance)

Where μ_s is the population mean (mean performance) of selective crossover and μ_A is the population mean of the alternative recombination operator (the operator it is being tested against).

Given our assumption that the initial population for each GA run will have a significant effect on the performance (the assumption of dependence of results), the appropriate test of the hypothesis of no difference is the use of the 'paired *t*-test'. Using the paired *t*-test initial conclusions were drawn; however, a test of the correlations between the operators on the One Max problem demonstrated that the results are actually independent. Thus, the appropriate test of hypothesis is the 'two sample *t*-test'[†]. It is the results of this test that is reported in Section 5.4; however, in these experiments the use of either the paired or two sample *t*-test did not alter the significance of the conclusions.

[†] In each experiment the *F*-test was used to test the equivalence of the two sample variances; where appropriate Welch's *t*-test was used.

5.2 GA Parameters

To make this a strict comparison of recombination operators all other GA operators and parameters were kept constant and a generational GA was used throughout this study. To be consistent all experiments and analyses in this thesis were conducted using these conditions, which represent one variation of the many different forms of genetic algorithms. As mentioned in Section 2.4, the optimal setting for each parameter differs from problem to problem and is also influenced by the other operators and parameters used in the GA. Hence, the optimal settings for our benchmark problems are not known. However, using suggestions by De Jong (1975), Grefenstette (1986) and Schaffer *et al.* (1991) we use the parameters listed below:

Crossover rate:	0.6
Mutation rate:	0.01
Selection:	SUS algorithm (Baker 1987)

5.3 Test Suite of Benchmark Problems

In our experiments we use a set of well-studied benchmark problems that provide a good variation of features present in real-world problems. The features considered in this study are epistasis, deception and fitness landscapes. The problems used are:

- One Max problem a linear fitness landscape with no epistasis and no deception.
- Royal Road function a staircase fitness landscape with epistasis and no deception.
- L-MaxSAT problems many flat regions in the fitness landscape with tuneable epistasis.
- NK landscapes tuneable ruggedness of fitness landscape with tuneable epistasis.
- Deceptive Trap functions a two-peak fitness landscape with tuneable deception.

The following sub-sections describe the properties of each test problem used and summarise the features of epistasis, deception and fitness landscapes in each problem.

5.3.1 One Max Problem

The One Max problem is a simple bit counting problem (Ackley 1987), where each bit that is set to '1' in the chromosome contributes an equal amount to the fitness, thus all combinations of 1 bits are good schemata which makes the function linear with no epistasis or deception. The fitness function is given below:

$$f(chrom) = \sum_{i=0}^{l} x_i$$
(5.1)

where l is the length of the chromosome. This problem is used to test how selective crossover handles simple fitness landscapes where a hill-climber will tend to outperform a traditional GA. The chromosome length is 50 and the population size is 100. The results are shown in Section 5.4.1.

The properties of the One Max problem are summarised below:

- It represents simple optimisation problems.
- It has a linear fitness landscape (in terms of Hamming distance).
- There is no epistasis and no deception.
- The maximum fitness of an individual is the length of the chromosome.

5.3.2 Royal Road Function

Royal Road problems are a class of functions that were designed for studying GA behaviour on landscapes that contain tightly coupled 'building blocks' (Mitchell, Forrest and Holland 1991). A building block in the context of Royal Road functions refers to a low order, low defining length group of bits on an individual that together contribute to that individual's fitness. The Royal Road functions were specifically designed to be 'GA-easy'; however empirical studies demonstrated that standard GAs find them difficult to

solve in comparison with other heuristic methods such as 'random mutation hill-climbing' (Forrest and Mitchell, 1993).

The Royal Road function considered in this study consists of partially specified schemata, s_i . These schemata are hierarchically structured and are predefined with corresponding fitness values as depicted in Figure 5.1. The shape of the fitness landscape is very much like a staircase consisting of four levels[†]. The lowest level (level 0) schemata are the shortest in order and in defining length. Level 1 schemata comprise a combination of adjacent level 0 schemata. Level 2 schemata comprise a combination of adjacent level 1 schemata and so on.

The fitness is calculated as the sum of all schema fitnesses that exist in the individual (by considering the instances of each schema in the individual). A chromosome x is said to be an instance of a schema $s, x \in s$, if x matches' s in the defined positions. For example chromosome (1111....1) is an instance of all schemata in Figure 5.1. The fitness f(x) of a chromosome x is given below:

$$f(x) = \sum_{i=1}^{14} \delta_i(x) o(s_i), \quad \text{where} \quad \delta_i(x) = \begin{cases} 1 & \text{if} \quad x \in s_i \\ 0, & \text{otherwise} \end{cases}$$
(5.2)

where $o(s_i)$ is the number of defined bits in s_i . For example if x is an instance of exactly two level 0 (order-8) schemata, then f(x) = 16. The Royal Road function is not deceptive but does contain epistasis, which is defined by the specific collection of bits (schemata) that contribute to the fitness; all other schema combinations that are not defined do not contribute to the fitness of an individual.

In our experiments we used the same parameters as Mitchell and Forrest (1996). The chromosome length is 64 and the population size is 128. The results are shown in Section 5.4.2.

[†] The number of levels can be altered by either removing or adding levels and by changing the steepness of the steps (by increasing or decreasing the fitness difference between levels).

Si	Schema Instance	$O(S_i)$
S ₁	11111111*******************************	8
<i>S</i> ₂	*******11111111************************	8
S 3	***************************************	8
S4	***************************************	8
S5	***************************************	8
<i>S</i> ₆	***************************************	8
S7	***************************************	8
S8	***************************************	8
S9	1111111111111111***********************	16
S10	******************111111111111111111111	16
S11	***************************************	16
S12	***************************************	16
S13	111111111111111111111111111111111******	32
S14	***************************************	32

Figure 5.1: Simple Royal Road Function

The properties of the Royal Road function are summarised below:

- It represents problems that contain tightly coupled building blocks.
- It has a structured ('staircase') fitness landscape.
- It contains epistasis but no deception.
- All the desired schemata of the problem are known in advance as they are explicitly built into the fitness function.
- The maximum fitness of an individual is 192.

5.3.3 Random L-MaxSAT Problems

The Boolean satisfiability (SAT) problem is a well-known constrained satisfaction problem, which consists of variables or negated variables that are combined together to form clauses using **and** (\wedge) and **or** (v). Typically SAT problems are presented in conjunctive normal form. The goal is to find an assignment of 0 and 1 values to the variables such that the boolean expression is true. The random L-MaxSAT problem generator (Mitchell, Selman and Levesque 1992; De Jong, Potter and Spears 1997) is a boolean expression generator. It creates random problems in conjunctive normal form subject to three parameters V (number of variables), C (number of clauses) and L (the length of the clauses). Each clause is generated by randomly selecting L of the V variables and negating each variable with probability 0.5. Hence, each variable occurs (on average) in CL/V clauses. By increasing the number of clauses, the number of instances of a variable in each clause also increases. As a result epistasis increases and creates more constraints in finding an assignment of 0 and 1 values to the variables such that the boolean expression is true. By using this problem generator we can see how the recombination operators react to the change in epistasis when the fitness landscape contains many flat regions (Rana and Whitley, 1998).

The L-MaxSAT problems are encoded as binary bit chromosomes, where each bit represents a boolean variable. The fitness function for the L-MaxSAT Problem is given below:

$$f(chrom) = \frac{1}{C} \sum_{i=1}^{C} f(clause_i)$$
(5.3)

where *chrom* consists of C clauses, $f(clause_i)$ is the fitness contribution of each clause and is 1 if the clause is satisfied or 0 otherwise. Since the problem generator randomly generates problems on demand, there is no guarantee that such an assignment to the expression exists. The difficulty of a SAT problem increases as a function of the number of boolean variables and the complexity of the boolean expression.

In our experiments we used the same parameters as De Jong, Potter and Spears (1997). We keep V and L fixed and we change the number of clauses C to vary the amount of epistasis. The number of variables V is set to 100 and the clause length L is set to 3. The number of clauses C is varied from 200 (low epistasis) to 1200 (medium epistasis) to 2400 (high epistasis). The chromosome length is 100 (the number of variables) and the population size is 100. The GA was allowed to run for 600 generations and the number of evaluations administered was recorded at the end of each run or when a global solution was found. The results are shown in Section 5.4.3.

The properties of the L-MaxSAT problem are summarised below:

- It represents problems that contain epistasis.
- It allows tuning of epistasis.
- It allows tuning of problem difficulty.

- There are many flat regions in the search landscape (Rana and Whitley, 1998).
- The maximum potential fitness of an individual is 1, although there is no guarantee that such an individual exists.

5.3.4 NK Landscape Problems

Kauffman's NK landscapes are a class of problems used in theoretical biology to study rugged fitness landscapes (Kauffman, 1993). The NK landscape models epistatic couplings and the effects of these couplings between genes. N is the number of genes (in our case the number of bits) and K is the number of epistatic interactions each gene has to the other genes. When K is increased the number of peaks in the Hamming landscape also increases and thus the landscape becomes more rugged and uncorrelated. By increasing K, we are increasing the number of epistatic interactions and as a result more conflicting combinations of alleles arise (an allele's value may increase the fitness contribution of its corresponding gene, but also reduce the fitness contribution of other genes). It is these conflicts that make the landscape rugged and therefore there will be low correlation between the fitness of individuals and the similarity of their genotypes (Kauffman, 1993).

The fitness of the entire individual is computed by the average of the fitness contributions from each locus as shown in Equation 5.4. The fitness contribution of each gene to the whole individual depends on its own allele plus the alleles of the K other genes (allele combinations). Thus, the fitness depends on the alleles present in K+1 genes and for a bit encoding the total number of allele combinations is 2^{k+1} . To ensure epistatic interactions, each gene in each of the possible individuals is randomly assigned a fitness contribution between 0.0 and 1.0. The fitness function for the NK landscape problem is given below:

$$f(chrom) = \frac{1}{N} \sum_{i=1}^{N} f(locus_i)$$
(5.4)

where $f(locus_i)$ is the fitness contribution of locus *i*. These fitness contributions can be stored in a look-up table, Figure 5.2 and Table 5.1 shows an example of a NK landscape where N=4 and K=2. In Figure 5.2, A, B, C and D represent the genes in the

chromosome, which is of length N. Each gene interacts with its adjacent genes; those genes at the ends of the chromosome interact with each other. Table 5.1 shows the fitness contribution of each locus, which is determined by the corresponding allele and the alleles of the two genes that are linked to it (the allele combinations, which are given as a substring in the first column). To evaluate the fitness of the whole chromosome an average is taken the fitness contribution for each allele. For on example, $f(0110) = \frac{1}{4}(0.3 + 0.9 + 0.7 + 0.5) = 0.6$. The fitness contribution of gene A is 0.3 because (i) its allele value is 0, (ii) the allele of the gene linked to its left is 0 and (iii) the allele of the gene linked to its right is 1. These alleles concatenated in this order formulate the sub-string 001. Therefore, looking at sub-string 001 and the entry at A, its fitness contribution is 0.3. The fitness contribution of gene B is 0.9 because (i) its allele value is 1, (ii) the allele of the gene linked to its left is 0 and (iii) the allele of the gene linked to its right is 1. These alleles concatenated in this order formulate the sub-string 101. Therefore,

looking at sub-string 101 and the entry at B, its fitness contribution is 0.9. The others are obtained in the same way.



Figure 5.2: An example of epistatic interactions for an NK landscape where N=4 and K=2.

Sub-string	Α	В	С	D
000	0.6	0.3	0.5	0.7
001	0.3	0.9	0.6	0.5
010	0.7	0.6	0.4	0.5
100	0.2	0.4	0.7	0.8
011	0.4	0.8	0.1	0.9
101	0.8	0.9	0.1	0.6
110	0.1	0.5	0.7	0.9
111	0.4	0.7	0.9	0.1

Table 5.1: A table of fitness contributions for the NK landscape given in Figure 5.2.

A practical constraint, in using this problem, is the amount of computational space required to store the tables used to compute the fitness. The amount of storage required grows exponentially as K increases. Therefore in our implementation the landscape is generated on the fly. In our experiments 50 landscapes are generated randomly and thus the global optimum is unknown. We keep N fixed to 32, which also is the length of the chromosome. We vary the number of interactions K from 4 (with little epistasis and a landscape with few peaks) to 20 (with medium epistasis and a rugged landscape) to 31 (high epistasis with an uncorrelated rugged landscape). The GA was allowed to run for 150 generations and the number of evaluations administered was recorded at the end of each run. The results are shown in Section 5.4.4.

The properties of the NK landscape problem are summarised below:

- It represents problems that contain epistasis at adjacent genes.
- It allows tuning of epistasis.
- It allows tuning of problem difficulty.
- Increasing K makes the fitness landscape more rugged.
- The maximum fitness of an individual is not always known.

5.3.5 Deceptive Trap Functions

Unitation u is the number of 1s in a bit string and Deceptive Trap functions (Ackley 1987) are linear functions of unitation. They depend only on the number of 1s in an individual and not on the positions of the 1s. A Trap function divides the search space into two peaks; one peak is the global optimum and the other peak is a local optimum. An example of a trap function is given in Figure 5.3.



Figure 5.3: A Deceptive Trap function where a bit string of all 1s is the global optimum and a bit string of all 0s is the local optimum.

The fitness function for a Deceptive Trap problem is:

$$f(chrom) = \begin{cases} \frac{a}{z}(z-u), & \text{if } u \le z; \\ \frac{b}{l-z}(u-z), & \text{otherwise;} \end{cases}$$
(5.5)

where a and b are constants, l is the length of the chromosome, u is the number of 1s in the string and z is the slope change location.

A Deceptive Trap function is an example of a deceptive problem that GAs find difficult to solve because the GA converges to the local optimum and is unable to find the global optimum; however, some Trap functions are more deceptive than others. Deb and Goldberg (1993) analysed the deception in trap functions; they define a parameter r to be the ratio of the local and global optimal function values (r = a/b) and used r to define boundaries between trap functions that are not deceptive, partially deceptive and fully deceptive (see Section 2.6). They define an order k deceptive trap function as one where kof the l bits in the chromosome (not necessarily situated together) are deceptive and lead the GA away from the global optimum.

In our experiments we use these boundaries to vary the amount of deception in a trap function. This would allow us to determine the levels of deception at which a recombination operator can successfully find the global optimum.

For this study we have used a partially Deceptive Trap problem where no tight ordering exists and a single trap function, rather than sub-functions, is represented in the chromosome (Deb and Goldberg 1993). The fitness value is scaled so that the maximum fitness is 1.0, thus b = 1. The parameters a and z are varied to increase or decrease deception. Experiments are carried out on partially deceptive trap functions of order 10, 15 and 20, which show the typical behaviour as deception is increased. For order 10, r = a = 0.21 and z = 30. For order 15, r = a = 0.35 and z = 30. For order 20, r = a = 0.6 and z = 29. The chromosome length l is 50 and the population size is 1000. The results are shown in Section 5.4.5.

The properties of the Deceptive Trap problem are summarised below:

- It represents deceptive problems.
- It allows tuning of deception.
- Its has a two peak fitness landscape.
- The maximum fitness of an individual is b, which is set to 1 in our experiments.

5.4 Experimental Results

For all problems the performance evaluation is presented in graphical and tabular form. As mentioned in Section 5.1 the performance measure is the number of evaluations taken to find the global solution or the number of evaluations taken to find the best solution after x generations. The results were taken from 50 independent runs.

5.4.1 One Max Problem

Table 5.2 shows the mean number of evaluations taken to find the global solution for the One Max problem. The standard deviation is shown in parentheses. Figure 5.4 shows a comparison of the mean performances presented by selective, two-point and uniform crossover.

	One Max - the global solution is 50			
	Mean evaluations	Mean solution		
Selective crossover	3557 (1118)	50		
Two-point crossover	3733 (484)	50		
Uniform crossover	3686 (449)	50		

Table 5.2: Results for the One Max problem. Mean number of evaluations taken to solve the One Max problem. The standard deviation is shown in brackets.



Figure 5.4: Results for the One Max problem - performance comparison of selective, two-point and uniform crossover.

In Table 5.2 the number of evaluations taken to find the global solution in the One Max problem show that selective crossover took the least number of evaluation on average at 3557, followed by uniform crossover at 3686 and two-point at 3733.

From Figure 5.4 it seems that selective crossover shows a great deal of improvement in mean performance. However, tests for significance at the 5% level show that there was no significant difference in the mean performances of the three operators. Hence, selective, two-point and uniform crossover show equal performance on the One Max problem.

5.4.2 Royal Road Function

Table 5.3 shows the mean number of evaluations taken by selective, two-point and uniform crossover to find the global solution for the Royal Road function (with intermediate steps at four levels). The standard deviation is shown in parentheses. Figure 5.5 shows a comparison of the mean performances presented by selective, two-point and uniform crossover.

	Royal Road - the global solution is 192			
	Mean evaluations	Mean solution		
Selective crossover	64598 (34192)	192		
Two-point crossover	38287 (16894)	192		
Uniform crossover	74128 (27469)	192		

Table 5.3: Results for the Royal Road function. Mean number of evaluations taken to solve the Royal Road function. The standard deviation is shown in brackets.



Figure 5.5: Results for the Royal Road function - performance comparison of selective, two-point and uniform crossover.

The Royal Road function was designed for one-point crossover, which has a high positional bias hence it was expected that two-point crossover, which also has a high

positional bias will work well with this function. From Table 5.3 and Figure 5.5 we can see that two-point crossover out-performed selective and uniform crossover, where uniform crossover took the most number of evaluations. Tests for significance were carried out and showed that two-point crossover is significantly better than selective crossover when solving the Royal Road problem.

5.4.3 Random L-MaxSAT Problems

Table 5.4 shows the mean number of evaluations taken and the mean solution found by selective, two-point and uniform crossover on the Random L-MaxSAT problems with low, medium and high epistasis. Recall from Section 5.3.1 that there is no guarantee that an assignment to the boolean expression exists; thus, there is no guarantee that a global solution (with a fitness value of 1.0) exists. The standard deviation is shown in parentheses. Figure 5.6 shows a comparison of the mean performances presented by the three operators.

	Low epistasis		Medium epistasis		High epistasis	
	Mean	Mean	Mean	Mean	Mean	Mean
	evaluations	solution	evaluations	solution	evaluations	solution
Selective	24098	0.997	28214	0.937	27573	0.916
crossover	(6430)	(0.002)	(509)	(0.002)	(484)	(0.002)
Two-point	29206	0.999	37188	0.932	37275	0.911
crossover	(1064)	(0.005)	(69)	(0.002)	(61)	(0.002)
Uniform	29621	0.999	38306	0.931	38310	0.911
crossover	(5782)	(0.003)	(42)	(0.003)	(43)	(0.002)

Table 5.4: Results for the Random L-MaxSAT problems. Mean number of evaluations to find the best solution for low, medium and high epistasis. The standard deviation is shown in brackets.



Figure 5.6: Results for the Random L-MaxSAT problems - performance comparison of selective, two-point and uniform crossover.

At all levels of epistasis, the comparison of performance in Figure 5.6 indicates that selective crossover performed better than two-point and uniform crossover with the least number of evaluations taken. Moreover, the results in Table 5.4 indicate that as epistasis increased selective crossover is able to find a better solution than two-point and uniform crossover. When epistasis is low, the standard deviation of the mean evaluations is larger than those shown in medium and high epistasis because when epistasis was low the GA was able to find the global solution (1.0) on some occasions and as a result the GA terminated earlier than 600 generations, which is when all the other runs terminated.

Tests for significance, using the t-test with a 5% significance level, were carried out, to compare the difference between the means. This showed that with low, medium and high epistasis the performance of selective crossover is significantly better than two-point and uniform crossover; on low, medium and high epistasis selective crossover shows an improvement of 19%, 26% and 28% respectively. This demonstrates that selective crossover works well with epistatic problems.

5.4.4 NK Landscape Problems

Table 5.5 shows the mean number of evaluations taken and the mean solution found after 150 generations by selective, two-point and uniform crossover on the NK landscape problems, where N was fixed at 32 and K was modified with values 4, 20 and 31. Recall from Section 5.3.4 that all landscapes were generated randomly and as a result the global optimum for each landscape was unknown. Figure 5.7 shows a comparison of the mean performances presented by selective, two-point and uniform crossover.

	K = 4		K = 20		K = 31	
	Mean	Mean	Mean	Mean	Mean	Mean
	evaluations	solution	evaluations	solution	evaluations	solution
Selective	11658	0.747	7931	0.687	10631	0.729
crossover	(832)	(0.015)	(1312)	(0.017)	(1526)	(0.010)
Two-point	14158	0.749	16642	0.701	15690	0.748
crossover	(549)	(0.009)	(92)	(0.013)	(110)	(0.014)
Uniform	17516	0.736	18430	0.703	18295	0.743
crossover	(383)	(0.010)	(18)	(0.013)	(43)	(0.010)

Table 5.5: Results for the NK Landscape problems. Mean number of evaluations to find the best solution for K=4, 20 and 31. The standard deviation is shown in brackets.



Figure 5.7: Results for the NK Landscape problems - performance comparison of selective, two-point and uniform crossover.

On all values of K, selective crossover showed superior performance, with the least number of evaluations, by a statistically significant amount and uniform crossover showed the worst performance with the most evaluations. The quality of the solution found by the three recombination operator did not differ significantly.

From Figure 5.7 as K increases and epistasis increases, the performance of twopoint and uniform crossover decreased, whereas the performance of selective crossover is consistent (approximately around 10000 evaluations) on all three values of K. Compared to two-point crossover, selective crossover shows an improvement of 52% when K=20 and an improvement of 14% when K=4. This demonstrates that selective crossover works well with epistatic problems that also have rugged fitness landscapes.

5.4.5 Deceptive Trap Functions

Table 5.6 shows the mean number of evaluations taken by selective, two-point and uniform crossover to find the global solution for order 10 and 15 trap functions. All three operators were unable to find the global solution on an order 20 trap function, thus the GA was terminated at 600 generations. Figure 5.8 shows a comparison of the mean performances presented by selective, two-point and uniform crossover.

	Deceptive Trap (the global solution is 1.0)					
	Order 10		Order 15		Order 20	
	Mean	Mean	Mean	Mean	Mean	Mean
	evaluations	solution	evaluations	solution	evaluations	solution
Selective	10856	1.0	15033	1.0	131519	0.6
crossover	(2058)		(2909)		(3070)	
Two-point	12735	1.0	15168	1.0	167100	0.6
crossover	(1372)		(3286)		(1855)	
Uniform	11313	1.0	15833	0.93	221950	0.6
crossover	(1059)		(5317)	(0.01)	(1402)	

Table 5.6: Results for the Deceptive Trap problems. Mean number of evaluations completed for order 10, 15 and 20 partially deceptive trap functions. The standard deviation is shown in brackets. The solution quality is given in square brackets.



Figure 5.8: Results for the Deceptive Trap problems - performance comparison of selective, two-point and uniform crossover.

Our initial expectation was that selective crossover was too exploitative (exploration is achieved solely through the generation of Child 2 and, of course, mutation) and would always be misled by the local optimum in all experiments. In Table 5.6, the number of evaluations taken to reach the solution on an order 10 trap function indicates that selective crossover took the least number of evaluations on average at 10856, but this is not a significant amount compared with uniform crossover. In an order 15 trap function selective crossover again took the least number of evaluations as well as finding a solution in early generations. Uniform crossover did not always find the solution: misleading schemata of order 15 deceived it. Selective crossover, like two-point and uniform crossover, was unable to solve an order 20 trap function.

Tests for significance were carried out at the 5% level show that selective crossover performs better than two-point crossover but the same as uniform crossover on the order 10 deceptive trap function. Selective crossover did not show a significant difference on the mean number of evaluations for order 15 deceptive trap functions. We can conclude that selective crossover was unable to solve the order 20 Deceptive Trap function; however, it performed as well as two-point and uniform crossover on the order 10 and 15 Deceptive Trap Functions.

5.5 Analysis and Discussion of Results

Table 5.7 gives a summary of the results by ranking the three different operators. A 1* indicates that the operator showed a statistically significant improvement and 3 indicates the operator showed worst performance.

	Selective crossover	Two-point crossover	Uniform crossover
One Max	<i>E q</i>	ual performa	nce
Royal Road	2	1*	3
Random L-MaxSAT	1*	2	2
NK Landscapes	1*	2	3
Deceptive Trap	<i>E q</i>	ual performa	nce

Table 5.7: A summary of relative performance of selective, two-point and uniform crossover. A 1* represents best performance.

As shown in Table 5.7 selective crossover generally did well on all problems. Although the experiments were limited to a small set we can still see how the traditional crossover operators differ in performance on different problems.

The empirical results in the previous section demonstrated that selective crossover can efficiently solve problems that contain epistasis. It does this successfully on a search landscape that has many flat regions (L-MaxSAT problems) and on rugged landscapes (NK landscapes). Selective crossover successfully allows exploitation of genes, which are not necessarily situated together in the chromosome.

On the deceptive trap problems, like two-point and uniform, selective crossover cannot overcome the problem of being misled to the local optimum. This is due to the correlation property used to bias alleles (this is further discussed in Chapter 7). The correlation used is fitness increase, and in these problems the fitness increase does not always lead to the global optimum, thus selective crossover may end up following a rapid ascent to the local optimum. In conclusion, all three operators are able to solve problems with some deception but unfortunately the operators are unable to solve deceptive problems of order 20.

For the Royal Road problem, two-point crossover outperformed selective crossover; we suspect this is due to two reasons (both of which are further analysed in Chapter 8). The first is the positional bias inherent in two-point crossover; it has a better chance of propagating blocks of genes adjacent to each other. The second possibility is that selective crossover finds it difficult to construct and exploit schemata. In the Royal Road problem, selective crossover first has to construct a desired schema before it can bias the alleles during recombination, but as crossover occurs at each gene it may find it more difficult to construct the schemata than two-point crossover. In conclusion, if a problem is known to be similar to the Royal Road problem then the best recombination operator to use is two-point crossover. However, such detailed information about the problem is not usually known.

From these experiments we can conclude that selective crossover can give practitioners more confidence than two-point and uniform crossover as it performs well on all but one problem - the Royal Road, where selective crossover was out-performed by two-point crossover. However, further investigation is required to understand the mechanisms that make selective crossover a beneficial strategy to use.

5.6 Summary and Conclusions

This chapter evaluated selective crossover in terms of performance, which was measured as the number of evaluations taken to find a solution. Selective crossover was compared with two static recombination operators: two-point and uniform crossover. Our test suite was a set of well studied benchmark problems that display varying degrees of epistasis and deception. This test suite of benchmark problems ranged from simple to difficult problems and are listed below:

- 1. One Max problem a linear fitness landscape with no epistasis and no deception.
- 2. Royal Road function a staircase fitness landscape with epistasis and no deception.
- 3. L-MaxSAT problems many flat regions in the fitness landscape with tuneable epistasis.

- 4. NK landscapes tuneable ruggedness of fitness landscape with tuneable epistasis.
- 5. Deceptive Trap functions a two-peak fitness landscape with tuneable deception.

The results demonstrated that selective crossover can efficiently solve problems that contain epistasis as well as simple problems. The performance of selective crossover is significantly better than two-point and uniform crossover on L-MaxSAT and NK landscapes and worked as well as two-point and uniform crossover on the Deceptive Trap and One Max problem. Selective crossover was outperformed by two-point crossover on one occasion, the Royal Road problem, however it still performed better than uniform crossover. Two-point crossover succeeds because it is highly biased towards adjacent genes and thus can exploit the tightly coupled schemata inherent in the encoding of the Royal Road. This is further investigated in Chapter 8, which demonstrates that selective crossover out-performs two-point crossover when the encoding of the Royal Road function *does not* contain tightly coupled schemata.

We conclude that the results suggest that selective crossover may be a viable strategy as an adaptive recombination operator in which GA practitioners can have more confidence in using for best overall performance than two-point and uniform crossover. The evaluation provided in this chapter produced positive results that provide a justification to further investigate selective crossover. The first study is provided in the next chapter with an evaluation of the adaptive properties in selective crossover, which analyses the dominance values during a GA run.

Chapter 6

Features of Adaptation in Selective Crossover

The first evaluation of selective crossover in Chapter 5 demonstrated that selective crossover works better than two-point and uniform crossover on most problems used in that study. The aim of this chapter is to analyse the adaptive nature of selective crossover.

This chapter identifies, in Section 6.1, that the taxonomy used by Eiben *et al.* (previously described in Chapter 3) to classify the type of change adopted by a strategy is too rigid and does not allow selective crossover to be classified. Eiben *et al*'s taxonomy assumes that a strategy may only contain one feature of change and we show that selective crossover adopts two different features of change; both adaptive and self-adaptive. Section 6.1.4 presents a new taxonomy to classify the type of change, which is more flexible and allows selective crossover to be classified as well as other strategies that may adopt one or more features of change.

The remainder of this chapter analyses the adaptive behaviour in selective crossover. The adaptive and self-adaptive features are empirically analysed, in Section 6.2, to understand the contributing features behind the performance of selective crossover. Section 6.3 empirically analyses the adaptive behaviour of selective crossover in terms of the distribution of dominance values in the population.

6.1 Classification of Selective Crossover

Eiben et al. (1999) provided a general classification of adaptive techniques used in evolutionary computation (outlined in Section 3.7). To classify a technique, they

concluded that the two main criteria are "what is being changed?" and "how the change is made? To classify selective crossover we begin by identifying which component of the GA is being changed. It is clear that selective crossover is changing the way recombination occurs. All other parameters of the GA remain fixed throughout a run.

Identifying and classifying how the change is made (Eiben *et al*'s second criterion) is not so trivial. Eiben *et al*. define that the type of change can only be classified into one of three categories: deterministic, adaptive and self-adaptive; as shown in Figure 6.1. These categories are relevant in classifying selective crossover and are discussed in the following three sub-sections.



Figure 6.1: Taxonomy of parameter control by Eiben et al. (1999)

6.1.1 Deterministic Features

Eiben et al. define deterministic parameter control as:

"This takes place when the value of a strategy parameter is altered by some deterministic rule. This rule modifies the strategy parameter deterministically without using feedback from the search. Usually, a time-varying schedule is used, i.e., the rule will be used when a set number of generations have elapsed since the last time the rule was activated."

From this definition selective crossover has no deterministic features because it uses feedback from the search to modify the dominance values and does not use any time-varying schedule.

6.1.2 Adaptive Features

Eiben et al. define adaptive parameter control as:

"This takes place when there is some form of feedback from the search that is used to determine the direction and/or magnitude of the change to the strategy parameter. The assignment of the value of the strategy parameter may involve credit assignment, and the action of the evolutionary algorithm may determine whether or not the new value persists or propagates throughout the population."

From this definition selective crossover is adaptive as the fitness correlation between parents and offspring is used as the feedback from the search and this is used to update the dominance values. The updating of the dominance values is very much like credit assignment; an allele's dominance value is increased by the fitness increase. However the methods used by selective crossover are also self-adaptive as shown in the next section.

6.1.3 Self-Adaptive Features

Self-adaptive parameter control according to the following definition provided by Eiben *et al.* (which was refined from earlier definitions by Angeline 1995 and Hinterding *et al.* 1997) is:

"The idea of the evolution of evolution can be used to implement the self-adaptation of parameters. Here the parameters to be adapted are encoded into the chromosomes and undergo mutation and recombination. The better values of these encoded parameters lead to better individuals, which in turn are more likely to survive and produce offspring and hence propagate these better values."

In selective crossover the dominance values are encoded into the chromosome as a dominance vector, which also undergoes recombination. A higher dominance value indicates a potentially beneficial allele and the recombination of these alleles with high dominance values may lead to better individuals. Additionally, both the chromosome and dominance vector of an individual are subjected to selection. The individuals that survive retain their dominance values to propagate onto the next generation. Those individuals that do not survive are discarded along with their associated dominance values. Thus, the dominance vector is used to evolve the crossover positions during recombination. This indicates that selective crossover, by Eiben *et al*'s definition, also contains a self-adaptive feature.

Selective crossover uses a combination of two types of change: adaptive and selfadaptive and thus cannot be classified in the taxonomy proposed by Eiben *et al*. The need for classification is still necessary for future research to understand what methods have been employed for a particular strategy. Hence, a more flexible taxonomy is required for the three types of change. In the next section we refine the classification provided by Eiben *et al*.

6.1.4 A New Taxonomy for Type of Change

Analysis of selective crossover in the previous sections indicated that the methods adopted by an adaptive strategy could not always be uniquely identified into Eiben *et al*'s taxonomy. A strategy can take on many different methods that make it adaptive. To overcome the rigid boundaries used in their classification, a better taxonomy for the type of change is the use of a Venn diagram where each category (deterministic, adaptive and self-adaptive) represents a mathematical set as depicted in Figure 6.2. This allows methods adopted by adaptive strategies to be classified into one or more categories if necessary. The definition of deterministic, adaptive and self-adaptive remain the same as originally in Eiben *et al.* (1999).

Given this new taxonomy selective crossover is classified in the intersection of the adaptive and self-adaptive types of change. The adaptive feature is the use of fitness correlations to update dominance values and the self-adaptive feature is the use of a dominance vector associated with each individual to evolve crossover positions. The adaptive feature holds the key properties of selective crossover: detection, correlation and preservation (see Section 4.3). Although preservation actually occurs during recombination this property is based on preserving alleles according to previous contributions to fitness. Thus, it is part of the adaptive feature because that is where correlation is used to store knowledge about previous fitness contributions.



Figure 6.2: New taxonomy for type of parameter control.

This new taxonomy suggests that a strategy may consist of all three features, but is it possible for a strategy to be deterministic, adaptive and also self-adaptive? Let us first consider an example of a deterministic feature; a method that alters the probability of mutation so that it changes with the number of generations is a deterministic feature (Hinterding *et al.* 1997). Similarly the probability of recombination can be altered with the number of generations or an entirely different crossover operator can be applied at fixed time intervals. For example consider selective crossover which currently has an adaptive and self-adaptive feature. If we were to include an additional method into selective crossover which applied uniform crossover at fixed intervals, say every 100 generations then this will be a deterministic feature. Hence with this additional method selective crossover would now have an adaptive, self-adaptive and a deterministic feature. This example is used to illustrate that it is possible for a strategy to have these three features; whether the additional deterministic feature is beneficial to selective crossover or not is left for future work.

Given this new taxonomy we can classify the adaptive recombination operators outlined in Section 3.3 as shown in Figure 6.3.



Figure 6.3: Classification of other adaptive recombination operators using this new taxonomy.

The adaptive strategy used by Eshelman and Schaffer (1995) (see Section 3.3.4) is deterministic because they switch between two recombination operators at a pre-defined time interval. The strategies used by Schaffer and Morishima (1987) and Spears (1995) (see Section 3.3.1 and 3.3.5 respectively) are self-adaptive because their strategy parameter, crossover positions and choice of recombination operator respectively, are encoded within the individual and are allowed to evolve. White and Oppacher (1994), like selective crossover use an adaptive and self-adaptive method similar to selective crossover. The strategy used by Louis and Rawlins (1991) (see Section 3.3.2) is also similar to selective crossover in that it uses feedback from the search and the use of an extra vector to determine where crossover should occur. However, its uses pre-defined rules to dictate how this extra vector is inherited by the children and thus this extra vector is not allowed to evolve through recombination.

In summary this new taxonomy allows adaptive techniques to be classified more precisely than the originally taxonomy provided by Eiben *et al.*, by accommodating for strategies that adopt one or more methods of change.

6.2 Analysis of the Adaptive and Self-Adaptive Features

Section 6.1 identified a combination of two types of change in selective crossover: adaptive and self-adaptive. The adaptive feature uses parental and offspring fitness correlations to update dominance values and thus bias alleles; the self-adaptive feature uses a dominance vector associated with an individual to dictate where crossover should occur.

In Chapter 5 selective crossover was evaluated in terms of performance on different problems and demonstrated to work better than two-point and uniform crossover on most problems. However it is not clear whether this increase in performance is due solely to the adaptive feature or the self-adaptive feature or the combination of both. This section empirically analyses the adaptive and self-adaptive features in selective crossover.

6.2.1 Experiments

The adaptive and self-adaptive features (identified in Section 6.1.4) in selective crossover are interconnected. The adaptive feature updates the dominance values that belong to the dominance vector (associated with each individual) and this dominance vector constitutes the self-adaptive feature. The adaptive feature cannot be analysed on its own because it updates the dominance values, which are part of the self-adaptive feature. By removing the self-adaptive feature there is no strategy parameter for the adaptive feature to update. On the other hand, the adaptive feature can be removed because the dominance values do not need to be updated for the self-adaptive mechanism to work. By removing the adaptive feature we would also be removing the three key properties of selective crossover: detection, correlation and preservation.

To understand which feature caused the increased performance the adaptive feature is removed from selective crossover. In this case the dominance values are assigned randomly as usual (see Section 4.4) and are never changed to reflect fitness increase; they remain fixed. This method is very similar to Punctuated Crossover (Schaffer and Morishima, 1987) described in Section 3.3.1, where the crossover positions were initialised randomly and were allowed to evolve through recombination. The GA cycle depicted in Figure 4.1 is thus updated to Figure 6.4. In this GA cycle the component that updates the dominance values is omitted.



Figure 6.4: GA Cycle of selective crossover that does not change the dominance values.

In this strategy the dominance values accompany the chromosome and recombination is conducted as usual (see Figure 4.3). The dominance values still dictate where crossover should occur, such that Child1 inherits the higher dominance value at each locus along with its associated allele value and Child2 inherits all other dominance and allele values. As usual both the dominance and gene vectors are subjected to crossover and selection.

This strategy was applied to One Max, Royal Road and NK landscapes (K = 4, 20, and 31) which served to illustrate the phenomenon. The same GA parameters were used as described in Chapter 5 and the results were taken for 50 independent runs.

6.2.2 Results and Analysis

For 50 independent runs, Table 6.1 shows the results of applying selective crossover with only the self-adaptive feature to One Max, Royal Road and NK landscapes (K = 4, 20, 31).

		Without A	daptive Feature	With Adaptive Feature	
	•	Mean	Standard	Mean	Standard
		solution	deviation	solution	deviation
One Max		37	1	50	0
Royal Road		56	16	192	0
NK Landscapes	NK Landscapes $K = 4$		0.018	0.747	0.015
K = 20		0.659	0.019	0.687	0.017
K = 31		0.612	0.017	0.729	0.010

Table 6.1: Results of applying selective crossover without its adaptive feature to the One Max, Royal Road and NK landscapes. Previous results with adaptive feature are also recalled in this table.

The results show that on all 50 runs of the One Max and Royal Road problems selective crossover without the adaptive feature was never able to find the global solution. The mean solution found for the One Max and Royal Road was 37 and 56 respectively (the global solution was 192; see Section 5.3.2). For the NK landscape the average solution found was 0.63 for all values of K. In comparison the results in Chapter 5 (also recalled in Table 6.1) showed that the original selective crossover i.e. with both the adaptive and self-adaptive features was able to find the global solution for the One Max and Royal Road on all 50 runs. It was also able to find better solutions for all values of K on the NK landscapes.

These results indicate that the performance of selective crossover deteriorates when the adaptive feature is removed, i.e. when the three key properties (detection correlation and preservation) are removed. This suggests that it is these three properties that are beneficial to selective crossover.

In conclusion, the adaptive and self-adaptive features in selective crossover are interconnected. The adaptive feature relies on the presence of the self-adaptive feature; however, this relationship is not reflexive. The results in this section indicate that the performance of selective crossover is due to the combination of the adaptive and self-adaptive features and not solely due of the self-adaptive feature. For this reason we term selective crossover as an "adaptive" recombination operator. All future experiments using selective crossover make use of both its adaptive and self-adaptive features.

6.3 Analysis of the Dominance Values in the Population

This section analyses the fluctuation of dominance values in the population to determine whether selective crossover is analogous to uniform crossover.

We identified in Chapter 4 that selective crossover is very much like uniform crossover, where the difference is that in uniform crossover there is a 50% chance of crossover occurring at each locus whilst, in selective crossover the dominance values dictate where crossover should take place. These dominance values undergo change using feedback from the search; however, if these dominance values were uniformly distributed across the population then selective crossover would be analogous to uniform crossover and the use of correlations would not actually be preserving alleles during recombination. We empirically observed the dominance values in the population to determine if this was true.

6.3.1 Experiments

To test if selective crossover is analogous to uniform crossover we tracked the distribution of dominance values at each generation on the One Max, Royal Road and NK landscape (N = 32 and K = 8), which served to illustrate the results. The fitness function of the One Max problem was normalised so that the maximum fitness is 1. This was done to limit the range of the distribution. All other GA parameters remained unchanged and were those used as in Chapter 5. For the One Max and Royal Road the GA was terminated when the global solution was found. For the NK landscapes the GA was terminated after 150 generations.

The distribution of dominance values were recorded at four intervals The intervals were chosen as the first generation and then at 33% (a third) of the run, 66% (two thirds) of the run and the final generation when the solution was found. For each interval the mean, mode, standard deviation and skewness of the distribution was recorded.

6.3.2 Results and Analysis

Figure 6.5, Figure 6.6 and Figure 6.7 show the distribution of dominance values for the One Max, Royal Road and NK landscape (N = 32 and K = 8) respectively.

Figure 6.5 shows the distribution of dominance values for the One Max problem at intervals of 1, 18, 36 and 53 generations on a single run. In the first generation the distribution of dominance values is uniform because the dominance values are randomly initialised to be in the range 0 to 1. However in subsequent generations the distribution of dominance values is constantly changing between generations and is far from uniform. At generation 18, in Figure 6.5, the distribution is completely different with a mean of 0.70 and the mode is a plateau between 0.45 and 1.0. As the GA run continues the distribution tends towards a normal distribution. By the final generation (generation 53) the distribution is symmetrical and approximately normal where the mean and mode are approximately the same. The standard deviation is 0.24.

For the Royal Road function in Figure 6.6, the distribution starts off uniformly but by generation 484 the distribution is positively skewed (skewed to the right), the mode is 28, the mean is 76 and the standard deviation is 57. The skewness of the distribution tends to increase as the generations increase; moreover, the distribution becomes very rugged by the final (1454th) generation. On the final generation the mode is still 28, the mean is now 118 and the standard deviation is 104.42.

For the NK Landscape, the distribution starts off uniformly but by generation 50 the distribution is negatively skewed (skewed to the left). The mean is 0.82, standard deviation is 0.33 and the mode is 0.9. On successive generations the distribution tends to be positively skewed and the positive skewness continues to increase. At the final generation the mean is equivalent to the mode at 0.9 and the standard deviation is 0.19.

The analysis of dominance values on all three problems shows that the dominance values are initially uniformly distributed, due to the random initialisation; however in subsequent generations the distribution of dominance values is constantly changing between generations and is not uniformly distributed. This demonstrates that selective crossover is not analogous to uniform crossover and shows dynamic behaviour.









Figure 6.5: The evolution of the distribution of dominance values in the population, for the One Max problem.








Figure 6.6: The evolution of the distribution of dominance values in the population, for the Royal Road function.









Figure 6.7: The evolution of the distribution of dominance values in the population, for the NK Landscape N=32 K=8.

From this analysis it is interesting to note that the distribution of dominance values is different for each problem. Moreover, in the Royal Road function the mode does not change on successive generations, whilst it does for the One Max and NK Landscape. This may be an indication that selective crossover behaves differently for each problem. Also the rugged distribution in the Royal Road may possibly reflect the structured search landscape inherent in the problem. However the results are not conclusive enough to generalise that the distribution of the dominance values reflects the structure of the landscape. A further quantitative analysis of these dominance values is required that would observe many runs across different problems and perform a cross-correlation of the distributions to find any re-occurring patterns. This additional analysis is beyond the scope of this thesis and is thus left for future work.

Another interesting point in the Royal Road function is that the fraction of alleles with dominance values less than 1 in the population remains constant after generation 484. As the minimum fitness increase is 8 one would expect these dominance values to increase by at least a factor of 8, but they remain unchanged with the value that was assigned to them during initialisation. Note that these values range from 0.8 to 1 and are located near or at the higher boundary of the initialisation range (at 1). This could be due to the bias selective crossover has towards higher dominance values. Selective crossover biases alleles with higher dominance values as the higher dominance values are intended to reflect beneficial alleles; however, in the initial generation such information about alleles is not known and the dominance values assigned will not reflect the true fitness contribution of the allele. Thus initialising dominance values can possibly have a negative effect on the recombination mechanism adopted by selective crossover; this is further investigated in the next chapter.

6.4 Summary and Conclusions

This chapter has identified two different features of change in selective crossover: adaptive and self-adaptive. The adaptive feature holds the three key properties of selective crossover: detection, correlation and preservation by use of parental and offspring fitness correlations to update the dominance values. The self-adaptive feature, in selective crossover, is the use of a dominance vector associated with each individual to evolve crossover positions. Owing to these two features, selective crossover could not be classified using the taxonomy provided by Eiben *et al.* The taxonomy they use assumes that a strategy can only adopt a single method of change. Using selective crossover as an example we prove that an adaptive strategy can adopt more than one method of change. The taxonomy provided by Eiben *et al.* was refined, in this Chapter, to form a new taxonomy represented as a Venn diagram where each category (deterministic, adaptive and self-adaptive) is a mathematical set. This new taxonomy accommodates strategies that adopt one or more methods of change.

An analysis of the adaptive and self-adaptive features was done by removing the adaptive feature from selective crossover. This was done to see whether the performance of selective crossover was due solely to the self-adaptive feature. This strategy was applied to the One Max, Royal Road and NK Landscapes and demonstrated that selective crossover with only the self-adaptive feature performs poorly. On all problems it was unable to find the global solution. This indicates that the performance of selective crossover is due to the combination of the adaptive and self-adaptive methods employed in selective crossover.

Empirical analysis on the adaptive behaviour was conducted by observing the distribution of dominance values at regular generational intervals. This was done for different problems. The results indicated that the distribution of dominance values in selective crossover are not uniformly distributed and thus it is not analogous to uniform crossover. This analysis also provides initial indications that selective crossover adapts to different problems by exhibiting different distributions and thus behaviour that corresponds to the problems. However, for more conclusive results, a further quantitative analysis of these dominance values is required, which would observe many runs across different problems and perform a cross-correlation of the distributions to find any reoccurring patterns. This additional analysis is beyond the scope of this thesis and is left for future work.

This analysis also identified that initialising dominance values may have a negative effect on the recombination mechanism in selective crossover. This is further addressed in the next chapter by analysing the biases inherent in selective crossover.

Chapter 7

Biases in Adaptive Recombination Operators

The previous chapter studied the adaptive properties in selective crossover: this chapter provides a novel study of biases on alleles present in allele-based adaptive recombination operators - selective crossover, masked crossover (Louis and Rawlins, 1991) and adaptive uniform crossover (White and Oppacher, 1994). This work is published in Vekaria and Clack, (1999a and 1999c).

The aim of this study is to understand how each of these three adaptive operators influences GA search and thereby identify any limitations and possible ways of enhancing these operators. Moreover, this would allow us to determine their suitability for specific problems.

This chapter begins by providing a detailed description of masked crossover and adaptive uniform crossover in Section 7.1, which also highlights the similarities and differences between these two operators and selective crossover. This is followed by Section 7.2, which performs three functions: first it identifies and describes four key biases on alleles (directional bias, credit bias, initialisation bias, and hitchhiker bias), which are present in the three adaptive recombination operators; secondly, it analyses the relationship between these biases; thirdly it characterises the three adaptive recombination operators in terms of these biases by analysing the specific methods used, in each operator, to guide recombination. Section 7.3 then introduces a known phenomenon in GAs called hitchhiking, which is a consequence of selection. We show that selective crossover is susceptible to this form of hitchhiking as well as its own hitchhiker bias, and thus the hitchhiker bias maybe a hindrance to GA search using selective crossover. The

remaining sections empirically analyse, in terms of performance, the effects of the initialisation and credit biases, which were identified as the cause of the hitchhiker bias in selective crossover.

7.1 Existing Allele-Based Adaptive Recombination Operators

Besides selective crossover there are two other allele-based adaptive recombination operators: masked crossover (Louis and Rawlins, 1991) and adaptive uniform crossover (White and Oppacher, 1994). This section provides detailed descriptions of these operators, so that their similarities and differences to selective crossover can be analysed.

7.1.1 Masked Crossover

Masked crossover (Louis and Rawlins, 1991) uses relative fitness information to guide recombination. To do this each chromosome has an associated bit vector (a binary mask) such that each locus has an associated bit value. On initialisation the binary masks are set randomly and are the primary element that dictates crossover. To describe how recombination occurs, let us 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 7.1 and as defined below:

```
for i=1 to ChromosomeLength

if Ml_i == 1 and M2_i == 0

copy i bit from Parent1 to Child1 and Child2

else if Ml_i == 0 and M2_i == 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 that are separate from those used for recombination. Figure 7.2 and Figure 7.3 depict examples of binary mask inheritance given parents shown in Figure 7.1. Children are categorised into three types; Good (fitter than best parent), Average (fitness

within range of parents) and Bad (less than or equally fit as worst parent). With two children produced by recombination 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 (see Figure 7.2) then the parents' masks are left unchanged and the masks of the children are given by (i) ORing the masks of the parents; (ii) for any 0's left in the children's masks randomly decide their value. In contrast, if both children are bad then the children's 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 7.3). Louis and Rawlins (1991) do not use generational replacement; instead they use a steady state population. This means that parents will still remain in the population after crossover and if their masks of both parents also undergo a change to reflect detrimental effects.

Par	rent1	L					Par	rent2				
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						
Ch	ild1						Ch	ild2				
0	1	1	1	0	1	Alleles	0	0	0	1	0	1

Figure 7.1: Recombination using masked crossover.

Child1						Ch	ild2					
1	?	1	1	1	1	Binary Masks	1	?	1	1	1	1
0	1	1	1	0	1	Alleles	0	0	0	1	0	1

Figure 7.2: Masked crossover – creation of new binary masks if both children are 'good'. (? Denotes randomly generated).

Child1							Ch	ild2				
1	0	?	1	0	1	Binary Masks	0	0	1	0	1	0
0	1	1	1	0	1	Alleles	0	0	0	1	0	1

Figure 7.3: Masked crossover – creation of new binary masks if both children are 'bad'. Assume fitness of Parent2 > Parent1. (? Denotes randomly generated).

The similarities between masked crossover and selective crossover are outlined below:

- Both operators use parental and offspring fitness information to bias alleles during recombination.
- Both operators are allele-based recombination operators because recombination is decided at each allele in the chromosome.

The differences between masked crossover and selective crossover are outlined below:

- Masked crossover uses a binary mask; thus, 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 equal to an allele that contributes more to the fitness increase.
- In masked crossover the updating of the binary mask does not take into account population homogeneity (when both parents have matching alleles at locus *i*). The binary mask is updated on the basis of which parent contributed to create a good or bad child and not on the differing bits that were exchanged to form the new child (Hamming distances between parents and offspring). For example, in Figure 7.1 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 1.
- The changes in the binary mask do not reflect the magnitude of the fitness increase. The binary mask is updated in the same manner whether there is a small or large increase or decrease in fitness, thus allowing very little competition amongst the alleles in the population.
- The binary mask does not undergo recombination; the masks are inherited or changed independently using pre-defined external rules.
- 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.

7.1.2 Adaptive Uniform Crossover

Adaptive uniform crossover (AUX) (White and Oppacher, 1994) also uses relative fitness information to guide recombination. In AUX each locus in the chromosome has an associated linear automaton with an odd number of states (the set of states $S = \{s_0, s_1, ..., s_N\}$), where N is fixed by the user. Hence, for each individual of length l there are l automata associated with it, see Figure 7.4. The * in each automaton indicates the current state of the automaton and each state maps to the probability of crossover (bit exchange) at that location.

When bits are exchanged their associated automata are also passed onto the child. States are changed using pre-defined rules that have been invoked as a result of the fitness differences between parents and offspring. Examples of rules that can be invoked are given in Table 7.1.

Offspring fitness	Reward bits from Parent1	Reward bits from Parent2
	with probability	with probability
>Parent1	$+P_{superiorReward}$	0
>Parent2	0	+P _{superiorReward}
<parent1< td=""><td>-PinferiorPenalty</td><td>0</td></parent1<>	-PinferiorPenalty	0
<parent2< td=""><td>0</td><td>-P_{inferiorPenalty}</td></parent2<>	0	-P _{inferiorPenalty}

Table 7.1: Updating rules for adaptive uniform crossover (AUX). $+P_{superiorReward}$ is the probability with which an automaton will be rewarded – moved to the next state. $-P_{inferiorPenalty}$ is the probability with which an automaton will be penalised – moved to the previous state.

White and Oppacher (1994) set $P_{superiorReward}$ to 1.0 and $P_{inferiorPenalty}$ to 1.0. Hence, a reward implies that the automaton moves from state *i* to *i*+1 and a penalty implies that the automaton moves from state *i* to *i*-1. If the automaton was in state *N* (the last state) prior to a reward then no change is applied to the automaton. This similarly applies when the automaton is in state 0 (the first state) prior to a penalty. The initial states of the automata are set randomly.

The rules in Table 7.1 are used to update the automata in both children created under recombination. For example, assume that in Figure 7.4 the fitness of Child1 is greater than both its parents' fitnesses; therefore, according to the rules in Table 7.1 the states of all automata get updated by moving to i+1, if the current state is i. For Child2, assume that its fitness is less than both parents; therefore, the states of all automata will get updated by moving to state i-1, if the current state is i.



Figure 7.4: Recombination using adaptive uniform crossover (AUX). The automata belonging to the offspring are updated here given that the fitness of Child1 > Parent1, Parent2 and fitness of Child2 < Parent1, Parent2.

The similarities between AUX and selective crossover are outlined below:

- Both operators use parental and offspring fitness information to bias alleles during recombination.
- Both operators are allele-based recombination operators because recombination is decided at each allele in the chromosome.
- In both operators the inheritance of the strategy parameter (the automata in AUX and the dominance vector in selective crossover) is determined by recombination and not external rules (as done in masked crossover).

The differences between AUX and selective crossover are outlined below:

- AUX uses a finite automaton; therefore the automaton cannot move to another state if the final state is reached; if there is then a fitness increase, stagnation ensues.
- In AUX, each automaton state maps to a probability, thus recombination is still probabilistic whereas in selective crossover recombination is deterministic.
- The change in states of the automaton, like masked crossover, does not reflect the magnitude of the fitness increase. Hence the automaton changes state if there is a small or large increase in fitness.
- AUX does not take into account population homogeneity (when both parents have matching alleles at loci i). The algorithm only rewards or penalises automata on the basis of which parent contributed to which child and not on the differing bits that were exchanged to form the new child. For example, in Figure 7.4 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 1.
- AUX penalises alleles when there is a fitness decrease but owing 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 also get penalised.

7.2 Analysis of Biases in Allele-Based Adaptive Recombination Operators

This section provides a novel study of biases in allele-based adaptive recombination operators. It identifies and describes four key biases on alleles, which are inherent in masked crossover, adaptive uniform crossover and selective crossover: directional, credit, initialisation and hitchhiker biases. These biases are an inherent part of the above three operators and are additional to the general bias of the GA (implemented by selection according to fitness).

7.2.1 Directional Bias

Directional bias exists if alleles are favoured (or not favoured) for their fitness contribution. This bias determines the direction the GA is likely to follow and benefits GA search as it pushes the GA towards particular regions in the search space to aid convergence.

All three operators have directional bias as they favour alleles exchanged during recombination that show a fitness increase in the individual relative to its parents. This form of bias means that the operators follow regions of the search space where fitness is likely to increase; therefore, the GA can potentially be misled and is unable to solve highly deceptive problems.

Masked crossover and AUX consider fitness increase and decrease when biasing an allele whilst selective crossover only considers fitness increase. Therefore selective crossover has a higher directional bias towards fitness increase than masked crossover or AUX. The results in Chapter 5 demonstrated that selective crossover, like two-point and uniform crossover, was unable to solve trap functions that are partially deceptive at order 20 and above. Thus selective crossover is not ideal for highly deceptive problems due to its directional bias.

7.2.2 Credit Bias

The credit bias is the degree with which an allele gets favoured with respect to its fitness contribution. Credit bias is used as a means for accumulating fitness information in 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 high credit for any degree of fitness increase.

Masked crossover has high credit bias as it makes use of a binary mask to accumulate fitness information. 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 7.1.1). This allows very little competition amongst the alleles to be retained in a fitter individual during recombination.

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 whether there is a fitness increase of 5 or 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). For these reasons, AUX also has high credit bias.

In the case of selective crossover, different variations in the fitness increase are captured as the fitness increase determines the degree to which the dominance values get increased (see Figure 4.4, page 61). Selective crossover uses real values; there is no restriction on how big these values can get, other than computational restrictions. These unbounded continuous values allow more competition amongst the alleles (as identified in Chapter 6). Hence, selective crossover has less credit bias than masked crossover and AUX.

7.2.3 Initialisation Bias

A recombination operator has an initialisation bias if alleles are favoured during initialisation. This means that alleles are exploited without any knowledge of their contribution to fitness. This form of exploitation occurs prior to any exploration and therefore restricts the explorative power of the GA.

The effect of the initialisation bias is greatly influenced by the population size. For example, in selective crossover the population size governs the distribution of the dominance values at each locus in the population. In Chapter 6 a distribution of the initial dominance values was plotted, which depicted a uniform distribution because it considered all dominance values in the population. However, the distribution of the dominance values at each locus or "*column*" of the population is not always uniform due to the stochastic noise in a small population size. For example, consider a population of 4 individuals of length 4 represented in Figure 7.5; the distribution of dominance values at each locus or each *column* does not show that these values are uniformly distributed because the population size is too small and therefore causes an initialisation bias. Given a "large enough" population this bias will not exist because the distribution will be evenly spread and therefore averaged out. As ideal population sizes are unknown for different problems (previously mentioned in Section 2.3.1), this bias can potentially have a negative affect on GA performance.

1	0.59	0.18	0.33	0.20				
	1	0	0	0				
2	0.77	0.98	0.14	0.16				
	0	1	1	1				
•			•	<u></u>				
3	0.4	0.83	0.21	0.85				
5	1	0	0	1				
4	0.85	0.65	0.5	0.82				
	0	1	0	1				

Figure 7.5: Population of 4 individuals, which demonstrates that the dominance values are not uniformly distributed at each locus due to the small population size. As the population size increases the distribution of the dominance values at each locus is more likely to be uniform.

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]. This implies that in 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 and in the case of adaptive operators we hope for guarantees in behaviour.

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

7.2.4 Hitchhiker Bias

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

The hitchhiker bias is inherent in selective crossover because the dominance values of alleles are increased if they were exchanged during recombination and as a result there is 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 schemata 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 will get increased by the same amount (the fitness increase). Hence in following generations the 0 will have a high dominance value and may get passed down to future generations. For the One Max problem such an event is not desirable. For example, in Figure 7.6 the shaded dominance values show four alleles that were exchanged to create Child1. Given the One Max problem, if Child1 has a fitness increase of 3, the 0 that was also exchanged at locus 2 will get biased. The dominance values of all four alleles (shaded) will increase by 3. This means that the 0 at locus 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.

Child1 - fitness = 4

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

Figure 7.6: Example of hitchhiker bias in selective crossover when applied to the One Max problem.

Both masked crossover and AUX have 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 Figure 7.2 and Figure 7.4. Also both operators penalise alleles that were exchanged and caused a fitness decrease but this has a potential for error as those alleles that are not the cause of the fitness decrease get penalised too.

Due to the continuous nature of the dominance values in selective crossover, compared with masked crossover and AUX, there is increased competition amongst the alleles thereby reducing the effect of the hitchhiker bias. For example, Figure 7.6 shows only a single recombination event whereas other individuals may contain a '1' at locus 2 with a higher dominance value; thus it is suspected that selective crossover has a better chance of discarding hitchhikers in the population.

7.2.5 Interdependencies Between Biases

The directional, credit and hitchhiker biases are not independent biases and each one is a direct result of another. Figure 7.7 shows the relationship amongst the four biases. The directional bias is a high level bias and can be considered as the general direction required by the GA. The credit bias is at 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. The credit bias can be assigned in many ways as shown by selective crossover, masked crossover and adaptive uniform crossover. 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 7.7 we can see how the directional, credit and 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 may be harmful to GA search.



Figure 7.7: Interdependencies amongst biases in allele-based adaptive recombination operators.

	Initialisation	Directional	Credit Bias	Hitchhiker Bias
	Bias	Bias		
Masked	High	Medium	High	High
crossover				
Adaptive uniform	High	Medium	High	High
crossover (AUX)				
Selective	High	High	Low	Medium
crossover				

Table 7.2: Relative strength of biases present in masked crossover, adaptive uniform crossover and selective crossover.

Table 7.2 provides a summary of the biases imposed by masked crossover, adaptive uniform crossover and selective crossover. All three operators have high initialisation bias because the strategy parameters (binary masks, automata and dominance vectors) are

initialised without knowing the fitness contributions of the corresponding alleles. Both masked crossover and AUX have a medium amount of directional bias because they consider the fitness increase and decrease when updating their strategy parameter, whilst selective crossover has a high amount of directional bias because it only considers fitness increase. Both masked crossover and AUX have a high credit bias because the magnitude of the fitness increase is not considered when updating their strategy parameter, whilst it is considered in selective crossover, which has low credit bias. Selective crossover has medium hitchhiker bias because it updates dominance values using Hamming distances between parent and offspring; thus, alleles that were identical in both parents are not biased as they are not considered to contribute to the fitness increase. Masked crossover and AUX have a high hitchhiker bias because alleles that were identical in both parents are still biased even though they did not contribute to the fitness increase. Moreover, as seen in Figure 7.7, a high hitchhiker bias is also the result of a high credit bias and high initialisation bias.

7.3 More Hitchhiking and Selective Crossover

The previous section identified the presence of a hitchhiker bias in selective crossover. Earlier studies by Schaffer, Eshelman and Offnut (1991) and Forrest and Mitchell (1993) show that GAs (that use one-point crossover) can also be susceptible to another form of hitchhiking due to over-sampling in selection. In this case schemata are sampled, by selection of individuals, at rates that are not justified by their fitnesses. Hence, undesirable schemata have been coupled along with desirable schemata in an above average individual, during recombination, which then gets sampled at a higher rate during selection. This produces more instances of the undesirable schemata together with the desirable schemata in the newly selected population. This phenomenon is known as "spurious correlations" or "hitchhiking" (Schaffer, Eshelman and Offnut (1991) and Forrest and Mitchell (1993)) where the undesirable (less fit) schemata hitchhike along with the more fit schemata. The reader should note that hitchhiking and the hitchhiker bias (described in Section 7.2.4) are two different entities; hitchhiking occurs at the allele level and is a consequence of the adaptive recombination operator.

Schaffer, Eshelman and Offnut (1991) first discovered hitchhiking as a consequence of selection in GAs. In their work they define hitchhiking as spurious correlations that are inherent in GAs and can lead to premature convergence. They explored the effects of two-point and uniform crossover on synthetic problems that are susceptible to hitchhiking. Two-point crossover is known to possess considerable positional bias meaning that schemata are less likely to get disrupted if they are situated together on the chromosome, therefore two-point crossover is not sufficiently vigorous to suppress hitchhikers. However, uniform crossover possesses no positional bias and therefore will tend to disrupt highly fit schemata as well as the hitchhikers. Schaffer *et al.* concluded that hitchhiking could not be eliminated and suggested that using a population-elitist strategy, in which individuals are introduced into the population only by replacing the worst of the population, can compensate for the disruptive behaviour of uniform crossover. This allows good solutions to be retained and reduces the high sampling rate of hitchhikers.

A further study on Royal Road functions (Forrest and Mitchell, 1993) showed that the convergence of the GA was slowed down by the presence of intermediate schemata (R2) (Forrest and Mitchell, 1993). To understand the slow convergence they traced specific schemata during evolution; the results showed that the slow convergence was due to hitchhiking. Their study used one-point crossover, which has a high positional bias and therefore is very unlikely to disrupt hitchhikers.

The above studies indicate that all recombination operators are susceptible to hitchhiking due to over-sampling in selection. Thus selective crossover, AUX and masked crossover are susceptible to hitchhiking *and* the hitchhiker bias on alleles as described in Section 7.2.4. Furthermore the hitchhiker bias at the allele level can also get over-sampled during selection and thus affect both future recombination and convergence. For example, in selective crossover consider an allele whose dominance value is increased when it did not contribute to the fitness increase of the individual (see Figure 7.6). This individual containing this allele now has a hitchhiker bias, which may be over-sampled during selection. Thus the newly selected population will have many instances of this individual and as a result more instances of this hitchhiker bias. This may affect future recombination and convergence, as recombination is dictated by the dominance values and is used to guide the GA towards fitter individuals. *For this reason, we suspect that the hitchhiker bias inherent in selective crossover, AUX and masked crossover can hinder*

the performance of the GA. The effect of this bias is empirically investigated for selective crossover in the next section. A similar study for AUX and masked crossover is left for future work.

7.4 Analysis of the Effect of Biases on Selective Crossover

The previous section identified that selective crossover is susceptible to hitchhiking as a consequence of selection and its own hitchhiker bias. Furthermore, from the relationship in Figure 7.7 it is clear that selective crossover is exposed to two doses of the hitchhiker bias; one from the credit bias and the other from the initialisation bias. In this section we show, using selective crossover as an example, how the credit bias and the initialisation bias affect GA search. This is done by reducing the credit bias and eliminating the initialisation bias. Therefore three variations of selective crossover are compared: 'original selective crossover' (Algorithm 7.1), 'selective crossover with less credit bias' (Algorithm 7.2) and 'selective crossover without the initialisation bias, (Algorithm 7.3).

Algorithm 7.1: Pseudo code for original selective crossover.

Algorithm 7.2: Pseudo code for selective crossover with less credit bias. The main difference to the "original selective crossover" is the way in which dominance values are updated which has been stressed in **bold**.

```
Genetic Algorithm:
     т=0
     Initialise population(P)
     Initialise dominance values to O(P)
     Evaluate(P)
     WHILE NOT finish DO {
           T=T+1;
           Selection(P)
           if T == 1 \{
                Recombination(P) /*uniform crossover*/
           }
           else Recombination(P) /*selective crossover*/
           Mutation(P)
           Evaluate(P)
           Update dominance values(P)
     }
```

Algorithm 7.3: Pseudo code for selective crossover without the initialisation bias. The main difference to the "original selective crossover" are stressed in **bold**.

Since our study is on the biases in selective crossover and how they affect GA performance we use two test problems which serve to illustrate the effects. We use NK landscape problems (N=32 and K= $\{4,20,31\}$) because they allow us to alter both the fitness landscape and difficulty of the problems, which our other test problems could not. We also use the Royal Road function because Chapter 6 identified that the dominance values for this problem retained some original initialisation values throughout the run and the initialisation bias expressed in these original values may affect the performance of selective crossover on the Royal Road problem.

Two sets of experiments were carried out using the NK landscapes and the Royal Road function; one without mutation and the other with mutation. All other GA parameters including the population sizes and chromosome lengths were the same as used in Chapter 5 (page 76). The results were taken for 50 independent runs

The first set of experiments were carried out with no mutation to analyse selective crossover on its own merits. In this case a comparison of the three different algorithms was done using the mean best solution found over 50 runs; the GA was allowed to run until the population had converged (when 95% of the individuals in the population were identical).

In the second set of experiments mutation was included into the GA and the mutation rate was 0.01. The three algorithms were compared in terms of performance (the number of evaluations taken).

7.4.1 Selective Crossover with Less Credit Bias

In Section 7.2.5 we stated that the credit bias causes a hitchhiker bias, which may be harmful for GA search. To determine whether the credit bias affects GA search 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 (as is done in the original selective crossover). The fitness increase is shared between the changed alleles because it is difficult to tell which allele actually contributed to the fitness increase (Figure 7.8). 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.

In the example shown in Figure 7.8 the fitness increase of Child 1 with respect to its parents is 0.1. 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 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 decreasing the magnitude of the credit bias we give to the alleles (the amount the dominance values get increased by). Therefore, we alter the way the dominance values are updated; the method is implemented as shown in Algorithm 7.2.

Parent 1 - fitness = 0.36

0.4	0.3	0.01	0.9	0.1	0.2
1	0	0	1	0	0

Child	1 –	fitness =	0.46
-------	-----	-----------	------

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

Parent 2 - fitness = 0.30

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

Child	1

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

Child 2

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

Child $2 - $ fitness = 0.20	
-----------------------------	--

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

Child 1 - fitness = 0.46

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

Child 2 - fitness = 0.20

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

Figure 7.8: Reducing the magnitude of the credit bias by sharing the fitness increase amongst alleles that were exchanged. Child1 produced a fitness increase of 0.1. Since two alleles were exchanged their dominance values get increased by 0.05 (0.1/2).

7.4.2 Selective Crossover without the Initialisation Bias

To see the effect of the initialisation bias we removed it from selective crossover so that the only biases present were those shown in Figure 7.9. To remove the initialisation bias a new method where dominance values are initialised at zero (Algorithm 7.3) is compared with the 'original selective crossover'.



Figure 7.9: Selective crossover without the initialisation bias.

Selective crossover requires different dominance values for crossover to occur (See Figure 4.3.); if two parents have an equal dominance value at the same loci the bits do not get exchanged. Since 'selective crossover without the initialisation bias' initialises all dominance values to zero the first recombination process using selective crossover cannot be carried out. Therefore in the first generation we use uniform crossover and then after evaluation the dominance values are increased according to the fitness increase presented by the new offspring. This then generates variation amongst the dominance values, which in subsequent generations allows selective crossover to be used. The use of uniform crossover in the first generation makes no difference in the way recombination occurs because it is exactly the same as initialising dominance values randomly; however this method does not cause an initialisation bias. The pseudo code for selective crossover without the initialisation bias is shown in Algorithm 7.3.

7.4.3 Results

For 50 independent runs on each problem, Table 7.3 shows the mean solution found by eliminating and reducing the initialisation and credit biases respectively when there was no mutation in the GA. When mutation is included in the GA, Table 7.4 and Table 7.5 show the mean fitness of the best solutions found and the mean evaluations taken to find the solution. In all tables the standard deviation is shown in parentheses.

GA without mutation							
	Royal Road	NK Landscap	NK Landscapes				
		K=4	K=20	K=31			
Original selective	14.93	0.7447	0.6811	0.7214			
crossover	(7.65)	(0.0122)	(0.0149)	(0.0141)			
Selective crossover	19.20	0.7430	0.6798	0.7218			
without the	(8.42)	(0.0189)	(0.0165)	(0.0160)			
initialisation bias							
Less credit bias	12.35	0.7114	0.6545	0.7060			
	(6.4)	(0.0179)	(0.0102)	(0.0127)			

Table 7.3: Mean fitness of best solution found when analysing the biases in selective crossover without mutation in the GA. Results are from 50 independent runs for the Royal Road and NK Landscape problems. The standard deviation is shown in brackets.

Without mutation all runs on the Royal Road problem were unable to find the global solution; the mean solution found by the original selective crossover was 14.93, by selective crossover without the initialisation bias was 19.20 and by selective crossover with less credit bias was 12.35. These initial results, without mutation, show that 'selective crossover with less credit bias' found poorer solutions for the Royal Road and NK Landscapes; 'selective crossover' on the Royal Road. However there was no significant difference, in the solutions found, between the two algorithms on the NK Landscapes. When mutation is used, selective crossover is able to find the global solution for the Royal Road and better solutions for the NK Landscapes as demonstrated in Chapter 5 and Table

7.4 and Table 7.5 below. This indicates that selective crossover needs mutation, to introduce diversity into the population, to allow it to explore and exploit other regions in the search space.

GA with mutation				
	Royal Road			
	Mean solution	Mean number of		
		evaluations		
Original selective crossover	192.00	64598 (34192)		
Selective crossover without the	192.00	55857 (25749)		
initialisation bias				
Less credit bias	192.00	65146 (40199)		

Table 7.4: Mean solution found and mean evaluations taken when analysing the biases in selective crossover with mutation in the GA. These are results from 50 independent runs for the Royal Road function. On all runs the global solution was found. The standard deviation is shown in parentheses.

GA with mutation								
		NK Landscapes						
	K=4		K=20		K=31			
	Mean	Mean	Mean	Mean	Mean	Mean		
	solution	evaluations	solution	evaluations	solution	evaluations		
Original selective	0.7466	11658	0.6866	7931	0.7285	10631		
crossover	(0.0147)	(832)	(0.0171)	(1312)	(0.0100)	(1525)		
Selective crossover	0.7236	11330	0.6903	7766	0.7317	10333		
without the	(0.0153)	(722)	(0.0160)	(1064)	(0.0177)	(1692)		
initialisation bias	1							
Less credit bias	0.7469	11943	0.6911	7687	0.7191	11066		
	(0.0110)	(827)	(0.0202)	(1414)	(0.0106)	(1376)		

Table 7.5: Mean solution found and mean evaluations taken when analysing the biases in selective crossover with mutation in the GA. Results from 50 independent runs for NK landscapes N=32 K= $\{4,20,31\}$. The standard deviation is shown in parentheses.

The results in Table 7.3 (GA without mutation) suggested that reducing the credit bias has an adverse affect on the GA but without mutation the optimal solution was not found in both the NK Landscapes and the Royal Road function. Thus it was not clear whether this was a result of the credit bias or the lack of mutation. The results shown in Table 7.4 and Table 7.5 show that when mutation is re-introduced in the GA, reducing the credit bias in selective crossover has no significant impact on the performance. Thus 'selective crossover with less credit bias' has no change in performance.

In Table 7.4, the 'original selective crossover', which contained all the biases depicted in Figure 7.7, found the optimal solution with 64598 evaluations on average. In comparison, when the initialisation bias was removed which thus reduced the hitchhiker bias ('selective crossover without the initialisation bias') the GA found the optimal solution in fewer evaluations than the original selective crossover; 55857 was the mean number of evaluations. A *t*-test for significance at the 5% level shows that the 'selective crossover without the initialisation bias' is significantly better than the 'original selective crossover'; the 'selective crossover without the initialisation bias' is significant bias' shows an improvement of 14%. A *t*-test also showed that there is no significant difference between the 'original selective crossover' and 'selective crossover with less credit bias'.

The results in Table 7.5 shows that for all values of K on the NK landscape problems the evaluations taken and the solutions found by 'original selective crossover', 'selective crossover without the initialisation bias' and 'selective crossover with less credit bias' were comparable with no statistical difference.

7.5 Analysis

Selective crossover is exposed to two forms of hitchhiking: (i) hitchhiking due to pejorative sampling rates by selection; (ii) a hitchhiker bias on alleles due to the initialisation bias and credit bias inherent in selective crossover.

Evidence of the deleterious effect of the hitchhiker bias was found when the performance of the 'original selective crossover' was compared with 'selective crossover without initialisation bias'. By eliminating the initialisation bias (Figure 7.9) we have reduced the hitchhiker bias and thereby increased the performance of selective crossover

in problems known to suffer from hitchhiking. For the Royal Road problem the GA without the initialisation bias was able to find the global solution with fewer evaluations than the original selective crossover. However for the NK landscape problems there was no real statistical difference in performance between the two algorithms. On one hand, eliminating the initialisation bias reduced the hitchhiker bias. On the other hand, selective crossover is still exposed to the hitchhiker bias (as mentioned in Section 7.2.5) due to the mechanism adopted by selective crossover where the correlation between parental fitnesses and offspring fitnesses is used to increase an allele's dominance value. The results suggest that the hitchhiker bias restricts search in selective crossover and should be eliminated. However, the hitchhiker bias in selective crossover can not be eliminated completely since the hitchhiker bias is an unavoidable consequence of the credit bias. Section 7.4.3 has shown that reducing the credit bias made no difference to the performance of the GA compared to the original selective crossover.

Selective crossover is susceptible to hitchhiking due to over-sampling in selection as well as its own hitchhiker bias. However, selective crossover is better than uniform crossover, which does not possess the hitchhiker bias and is considered to be more disruptive against hitchhiking (Schaffer, Eshelman and Offnut, 1991). Since neither operator is biased against schemata of high defining length, our intuition is that selective crossover does better than uniform crossover because of its adaptive schema preservation characteristics given by the directional and credit biases. Selective crossover implicitly stores knowledge in the dominance values by using correlations between parental fitnesses and offspring fitnesses, which allows selective crossover to suppress the adverse effects of hitchhiking.

The hitchhiker bias is suppressed in selective crossover because the dominance values are not bounded to predefined limits and are allowed to increase indefinitely with respect to fitness increase unlike masked and adaptive uniform crossover. Moreover, the dominance value of an allele a at position i in one individual is not necessarily the same for allele a at the same position in another individual. Each dominance value is a result of correlations in previous generations. This suggests that there is a great deal of competition amongst alleles, for a position on a chromosome, during recombination. For example in Figure 7.6 for the One Max problem, allele '1' is more superior than allele '0', however the dominance value of '0' at locus 2 was increased because of a fitness increase caused by the three '1's introduced in the chromosome along with the '0'. This is an

example of one recombination event in the entire population. Other recombination events may have combined a superior '1' at locus 2 thereby, in the next generation the hitchhiking '0' is suppressed by the higher dominance values of other superior alleles present at the same loci. The distribution of dominance values in Chapter 6 pictured the range of possible dominance values in the population.

These results suggest that the hitchhiker bias will also suppress the performance of masked and adaptive uniform crossover as both operators have more hitchhiker bias than selective crossover (see Section 7.2.5). Thus, using an alternative method of initialisation and a method of applying credit will increase the performance of these operators. This also suggests that selective crossover will outperform both masked and adaptive uniform crossover is left for future work (see Section 9.4).

7.6 Summary and Conclusions

This chapter has completed three analyses. Firstly it has studied the biases on alleles introduced by allele-based adaptive recombination operators that use fitness information to direct the GA. We have identified four key biases: directional bias, credit bias, initialisation bias and hitchhiker bias. The directional bias exists if alleles are either favoured or not favoured for their fitness contribution. The credit bias is the degree at which an allele gets favoured with respect to its fitness contribution. The initialisation bias exists if alleles get favoured during initialisation without knowing their fitness contribution. The hitchhiker bias exists if alleles get favoured during initialisation without knowing their fitness contribution. The hitchhiker bias exists if alleles get favoured when they do not contribute to the fitness increase.

Secondly, this chapter has analysed the relationship between these biases and has shown that they are not independent: the credit bias is a direct result of the directional bias, and both the credit and initialisation biases contribute to the hitchhiker bias. We have used these four biases to characterise three adaptive recombination operators (masked, adaptive uniform, and selective crossover) and to compare analytically 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. The directional bias in selective crossover makes it unsuitable to use when solving highly deceptive problems.

Thirdly, this chapter has used selective crossover to study the effects of the

hitchhiker bias. Selective crossover possesses two doses of the hitchhiker bias; one from the initialisation method used and the other from the credit bias, which uses correlations between parent and offspring fitnesses. We propose two methods to reduce the hitchhiker bias; (1) by reducing the credit bias and (2) by removing the initialisation bias. We compare the performance of three variations of selective crossover: 'original selective crossover', 'selective crossover with less credit bias' and 'selective crossover without the initialisation bias'. These three algorithms were applied to the Royal Road and NK Landscape problems. The results demonstrate that selective crossover is exposed to the hitchhiker bias and this bias hinders the performance of selective crossover. The results indicate that eliminating the initialisation bias reduces the hitchhiker bias and thus improves the performance of selective crossover. However as the hitchhiker bias is an unavoidable consequence of the credit bias it cannot be eliminated completely without a trade-off in performance (as was demonstrated in Section 6.2). A small reduction in the credit bias ('selective crossover with less credit bias') had no impact on the performance.

In conclusion, 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 sensitive. The hitchhiker bias is a problem for selective crossover, and it cannot be eliminated completely without a trade-off in performance. However the adaptive nature of selective crossover, and the exploitative biases it imposes on search using the dominance vector provides selective crossover with the ability to suppress hitchhiking due to selection and to yield better performance.

Chapter 8

Schema Propagation in Selective Crossover

To fulfil our aim in verifying the hypothesis of this thesis we previously, in Chapters 5, 6 and 7, evaluated selective crossover in terms of the performance, adaptive behaviour and biases respectively. In this chapter we conduct an analysis of schema propagation to provide us with more insight into the underlying behaviour of selective crossover; part of this work is published in Vekaria and Clack (1999b, 2000).

The aim of this study is to evaluate selective crossover, two-point and uniform crossover using different encodings of the Royal Road problem to determine if there is any relationship between the performance of these recombination operators and the way a problem is encoded. We use schema propagation to understand this relationship and to provide us with increased confidence that this relationship is applicable to other problems.

This chapter begins in Section 8.1 by describing three alternative encodings for the Royal Road problems. We then empirically observe, in Section 8.2, how the performance (the mean number of evaluations taken to find the global solution) is affected by the encoding and how the schemata that define these Royal Road encodings are propagated under selective, two-point and uniform crossover. This analysis identifies some similarities between selective crossover and uniform crossover and prompts us, in Section 8.3, to construct a schema survival probability for selective crossover using the schema survival probability constructed by Spears (1998) for uniform crossover.

8.1 Alternative Royal Road Encodings

It is often stated that the encoding is very important in the success of the GA; however, as clearly stated by Mitchell (1996, page 158):

"Choosing a fixed encoding ahead of time presents a paradox to the potential GA user: for any problem that is hard enough that one would want to use a GA, one doesn't know enough about the problem ahead of time to come up with the best encoding for the GA. In fact, coming up with the best encoding is almost tantamount to solving the problem itself!"

To compare how the performance of two-point, uniform and selective crossover is affected by the encoding of a problem we make use of the Royal Road functions, which are synthetic problems invented by Mitchell, Forrest, and Holland (1991). These functions were specifically created to understand the behaviour of a canonical GA that uses one-point crossover. Their simple predefined structure makes them a prime candidate for experiments to help us understand behaviours of other recombination operators. For our study we use the original encoding of the Royal Road function as was described in Section 5.3.2 (Mitchell, Forrest, and Holland 1991; Forrest, and Mitchell, 1993). We construct three new encodings of this Royal Road function so that schema fitnesses and schema orders, that define the Royal Road, remain the same and only the schema defining lengths differ in each encoding. In all encodings, the original encoding (in Figure 8.1) and the three new encodings (in Figure 8.2, Figure 8.3 and Figure 8.4), the schemata are defined as follows:

- S1 to S8 are level 0 schemata that have a fitness value of 8.
- S9 to S12 are level 1 schemata that have a fitness value of 16 and are defined by concatenating (from left to right) two adjacent level 0 schemata.
- S13 to S14 are level 2 schemata that have a fitness value of 32 and are defined by concatenating (from left to right) two adjacent level 1 schemata.

The original encoding consists of a structure of schemata made up of tightly coupled alleles. This tight coupling or linkage is gradually broken in the three new encodings so that the defining length of the level 0 schemata (denoted as δ (level 0)) increases and (by necessity) on other levels too. For example:

- In the second encoding, shown in Figure 8.2, the defining length of level 0 schemata is 14.
- In the third encoding, shown in Figure 8.3, the defining lengths of level 0 and level 1 schemata are 28 and 29 respectively.
- In the fourth encoding, shown in Figure 8.4, the defining lengths of level 0, level 1 and level 2 schemata are 56, 57 and 59 respectively.

		Encoding (A) – Original encoding δ (level 0)=7	Fitness
	S 1	11111111*******************************	8
(S2	*******11111111************************	8
	S3	***************************************	8
el (S4	***************************************	8
ev	S5	***************************************	8
I	S 6	***************************************	8
	S7	***************************************	8
	S 8	***************************************	8
	S9	111111111111111************************	16
el]	S10	***************************************	16
ev	S11	***************************************	16
	S12	***************************************	16
	S13	111111111111111111111111111111111111111	32
	S 14	**********************************111111	32
Sol	ution	111111111111111111111111111111111111111	192

Figure 8.1: Encoding A -the original encoding of a Royal Road function. In this encoding the defining length of level 0 schemata, $\delta(\text{level 0})$, is 7.

		Encoding (B) – Second encoding δ (level 0)=14	Fitness
	S 1	1*1*1*1*1*1*1*1************************	8
	S2	*1*1*1*1*1*1*1*1***********************	8
	S3	*************1*1*1*1*1*1*1*1*1*********	8
el C	S4	******************1*1*1*1*1*1*1*1******	8
ev	S5	*********************************1*1*1*1	8
	S 6	***************************************	8
	S 7	***************************************	8
	S 8	***************************************	8
	S 9	111111111111111111*********************	16
el 1	S10	***************************************	16
ev	S 11	***************************************	16
	S12	***************************************	16
	S13	111111111111111111111111111111111******	32
	S14	***************************************	32
Sol	ution	111111111111111111111111111111111111111	192

Figure 8.2: Encoding B – A new encoding of a Royal Road function. In this encoding the defining length of level 0 schemata, δ (level 0), is 14.

		Encoding (C) - Third encoding δ (level 0)=28	Fitness
	S 1	1***1***1***1***1***1***1***1**********	8
	S2	*1***1***1***1***1***1***1***1*********	8
	S3	**1***1***1***1***1***1***1***1****1****	8
el (S 4	***1***1***1***1***1***1***1***1*******	8
ev	S 5	***************************************	8
	S 6	***************************************	8
Ì	S 7	***************************************	8
	S 8	***************************************	8
	S9	11**11**11**11**11**11**11**11*********	16
el 1	S10	**11**11**11**11**11**11**11**11*******	16
ev	S11	***************************************	16
	S 12	***************************************	16
	S13	111111111111111111111111111111111111111	32
	S14	***************************************	32
Sol	ution	111111111111111111111111111111111111111	192

Figure 8.3: Encoding C - A new encoding of a Royal Road function. In this encoding the defining length of level 0 schemata, δ (level 0), is 28.

		Encoding (D) - Fourth encoding $\delta(\text{level } 0)=56$	Fitness
	S 1	1*****1*****1*****1*****1*****1*****1****	8
	S2	*1*****1*****1*****1*****1*****1*****1****	8
	S3	**1*****1*****1*****1*****1*****1*****1****	8
el (S 4	***1*****1******1*****1*****1*****1*****	8
Leve	S5	****1*****1*****1*****1*****1*****1*****	8
	S 6	*****1******1******1******1*****1*****1****	8
	S 7	*****1*****1*****1*****1*****1*****1****	8
	S 8	******1*****1*****1*****1*****1*****1****	8
Level 1	S 9	11*****11*****11*****11*****11*****11****	16
	S10	**11*****11******11*****11*****11*****11****	16
	S11	****11*****11*****11*****11*****11*****11****	16
	S12	*****11*****11*****11*****11*****11*****	16
2	S13	1111****1111****1111****1111****1111****	32
	S 14	****1111****1111****1111****1111****1111	32
Sol	ution	111111111111111111111111111111111111111	192

Figure 8.4: Encoding D – A new encoding of a Royal Road function. In this encoding the defining length of level 0 schemata, δ (level 0), is 56.

8.2 Analysis of Schema Propagation

This section provides empirical observations on the behaviour of two-point, uniform and selective crossover in terms of performance and schema propagation for different encodings of the Royal Road function. This study looks at the effects of recombination on schema propagation and thus only considers individuals that were subjected to recombination.

8.2.1 Experiments

To determine whether the performance of selective crossover is sensitive to the encoding of the problem we apply it to the four different encodings (the original and three new encodings) of the Royal Road function, presented in Section 8.1. On each encoding we measure:

- Performance (the number of evaluations taken to find the global solution). The measurement of performance will allow us to establish any relationship between the encoding of the Royal Road and the performance of the recombination operators.
- 2. Propagation of level 0 schemata. The measurement of schema propagation will provide us with a better understanding of the relationship between the

encoding and performance. It will also provide more insight into the underlying behaviour of the recombination operators.

We define schema propagation as the number of schemata that are transmitted from a parent to any offspring by recombination. Therefore, a schema present in a parent prior to recombination is considered to be propagated if the same schema exists in either child after recombination.

In these experiments the schema propagation of the level 0 schemata is observed; for any Royal Road encoding there are eight level 0 schemata (S1 - S8), thus the maximum number of level 0 schemata that can be propagated from any one individual is also eight. To calculate the mean number of level 0 schemata propagated by any individual during recombination, the schemata present in each parent are observed and compared with the schemata present in the offspring. For example, in Figure 8.5 Parent1 contains three level 0 schemata: S1, S3 and S7, and Parent2 also contains 3 level 0 schemata S1, S4 and S5. After recombination:

- Child1 contains 3 level 0 schemata: S1, S3 and S4; S1 is inherited from either Parent1 or Parent2, S3 is inherited from Parent1 and S4 is inherited from Parent2.
- Child2 also contains three level 0 schemata S1, S5 and S8; again S1 is inherited from either Parent1 or Parent2 and S5 is inherited from Parent2. Schema S8 in Child2 was not propagated but constructed and schema S7 in Parent1 was not propagated to the offspring.

	S 1	S2	S 3	S 4	S5	S 6	S 7	S8		
Parent1:	11111111*******111111111***************									
Parent2:	11111111*******************************									
Child1:	11111111	******	11111111	111111111	******	*******	******	*****		
Child2:	11111111	******	******	*******1	1111111	*******	********	11111111		

Figure 8.5: Example of schema propagation in a single recombination event. Parent1 contains schemata S1, S3 and S7. Parent2 contains schemata S1, S4, S8. Schemata S1 and S3 were propagated from Parent1 and schemata S1, S4 and S5 were propagated from Parent2.
Thus, the total number of schemata that were propagated by both parents is 5; 2 from Parent1 and 3 from Parent2. In this example, of a single recombination, the mean number of schemata propagated from any one parent is therefore 2.5. When considering all recombinations in the population the mean number of schemata propagated \overline{S} by any individual in the population is given as:

\overline{S} = total number of schemata propagated / total number of parents

For both measures, performance and schema propagation, the GA parameters were the same as those used in Chapter 5 (page 76). The results were taken for 50 independent runs. In these experiments we used selective crossover without random initialisation (Algorithm 7.3, page 128) because Chapter 7 demonstrated that initialising the dominance values to zero and using uniform crossover in generation 1 and selective crossover in subsequent generations increased the performance of selective crossover.

8.2.2 **Results on Performance**

For 50 independent runs on each problem, Table 8.1 shows the performance of selective, uniform and two-point crossover in terms of the number of evaluations taken to find the global solution. The standard deviation is shown in parentheses. In the original encoding (A) of the Royal Road function where $\delta(\text{level 0}) = 7$ (Figure 8.1) two-point crossover is superior; taking on average 38000 evaluations whilst selective and uniform crossover take 56000 and 74000 respectively. However, as the defining length of the level 0 schema increases, the performance of two-point crossover diminishes to 72000 evaluations. In contrast the performance of selective and uniform crossover remain the same. Selective crossover takes approximately 57000 evaluations to find a solution with any encoding. Uniform crossover takes approximately 72000 evaluations to find a solution with any encoding. Tests for significance show that selective crossover is consistently better than uniform crossover demonstrate equal performance on encoding (C); however selective crossover is significantly better than two-point on encoding (D).

(Encoding)	Selective crossover		Uniform crossover		Two-point crossover	
δ(level 0)	Mean	Standard	Mean Standard		Mean	Standard
	evaluations	deviation	evaluations	deviation	evaluations	deviation
(A) - 7	55857	(25749)	74128	(27469)	38287	(16894)
(B) - 14	61007	(39576)	69525	(37003)	42671	(22462)
(C) - 28	57320	(34162)	69376	(35672)	51351	(22172)
(D) - 56	57807	(28367)	75834	(47465)	72077	(35568)

Table 8.1: Mean number of evaluations taken to find the solution for different Royal Road encodings. The standard deviation is shown in parentheses.

8.2.3 Results on Schema Propagation

The graphs in Figure 8.6, Figure 8.7 and Figure 8.8 shows the behaviour, in terms of schema propagation, of two-point, uniform and selective crossover with the four different encodings described in Figure 8.1, Figure 8.2, Figure 8.3 and Figure 8.4.

In Figure 8.6 two-point crossover is less able to propagate schemata effectively as the defining length is increased (as expected from previous studies by Holland (1992), and Spears (1998)). Without genetic linkage embedded within the encoding, two-point crossover is not a reliable recombination operator. Uniform crossover is more consistent in propagating schema with or without genetic linkage (see Figure 8.7), but the payoff is performance; uniform crossover requires more evaluations. Selective crossover is also consistent in propagating schema with the different encodings but its performance is better than uniform crossover for all encodings (see Figure 8.8) and better than two-point crossover on encoding D, where the defining length of level 0 schema is 56.



Figure 8.8: Schema propagation in selective crossover with different Royal Road encodings. DL# is the defining length of level 0 schemata. The results show the number of level 0 schemata propagated per individual at each generation.

8.2.4 Analysis

The results in Sections 8.2.2 and 8.2.3 demonstrate that the performance of two-point crossover is affected by the encoding of the problem. As the defining length of level 0 schemata is increased the performance of two-point crossover deteriorates - we believe this is because it is less able to propagate the level 0 schemata. This is a result of its positional bias as described by Eshelman, Caruana and Schaffer (1989). The performances of both uniform and selective crossover are not affected by the encoding, but selective crossover has an advantage over uniform crossover by demonstrating better performance. From the graph in Figure 8.8 we can clearly state that selective crossover has no positional bias (because the rate at which schemata are propagated is approximately the same for each encoding). The results suggest that selective crossover will perform equally well on any encoding of the problem. No a priori knowledge is required to 'tune' the encoding in the hope of further increasing performance. Selective crossover consistently shows a steady growth in schema propagation. Clearly two-point crossover has an advantage when genetic linkage is embedded in the encodings; however, as mentioned before, it is difficult to know a priori which loci are important in useful schemata, so that they can be placed together in the encoding.

To understand further the differences in schema propagation for two-point, uniform and selective crossover we compared the schema propagation of each recombination operator on each different Royal Road encoding. The comparisons for each different encoding are shown in Figure 8.9, Figure 8.10, Figure 8.11 and Figure 8.12. In all figures the errors bars are omitted as they are small.



Figure 8.9: Encoding A – the original Royal Road encoding where the defining length of level-0 schemata $\delta(\text{level-0}) = 7$. A comparison of schema propagation between two-point, uniform and selective crossover.

For the original encoding (see Figure 8.9), two-point crossover is clearly able to propagate more schemata than uniform and selective crossover. The number of schemata propagated by selective crossover exceeds uniform crossover at generation 600 and approaches that of two-point at generation 1500.

For the second encoding (see Figure 8.10), the number of schemata propagated by two-point crossover has reduced but still shows better than uniform and selective crossover in early generations. The number of schemata propagated by selective crossover rises slowly in early generations but exceeds uniform and two-point crossover at generation 700 and 1250 respectively.



Figure 8.10: Encoding B - a Royal Road encoding where the defining length of level-0 schemata δ (level-0) =14. A comparison of schema propagation between two-point, uniform and selective crossover.



Figure 8.11: Encoding C – a Royal Road encoding where the defining length of level-0 schemata δ (level-0) =28. A comparison of schema propagation between two-point, uniform and selective crossover.

For the third encoding (see Figure 8.11), schema propagation by two-point crossover has further reduced. Again selective crossover shows low propagation in early generations however it exceeds uniform and two-point crossover at generation 500 and 900 respectively.

For the fourth encoding (see Figure 8.12), until generation 400, uniform crossover is able to propagate more schemata than selective crossover, which in turn propagates more than two-point crossover. Between generation 400 and 1000 all three operators show equal propagation. After generation 1000, the schemata propagated by selective crossover exceed two-point and uniform crossover, both of which continue to show equal schema propagation.



Figure 8.12: Encoding D – a Royal Road encoding where the defining length of level-0 schemata δ (level-0) =56. A comparison of schema propagation between two-point, uniform and selective crossover.

It is interesting to see that the growth in schema propagation for selective crossover is slow in early generations. However, in later generations more schemata are propagated than two-point and uniform crossover. This is true for all encodings. This behaviour may be a result of large amounts of exploration in early generations, which is necessary whilst the population is still diverse and helps to reduce the chances of premature convergence. This exploration is a result of the preservation characteristic of selective crossover. For selective crossover to preserve schemata during recombination, the dominance values must reflect the fitness contribution of the alleles in the schema. For the Royal Road, the dominance values can only do this when at least a single level 0 schema is created. Considering there are 8 defining positions in a level 0 schema there are 2⁸-1 alternative bit combinations that can be taken up by the 8 positions. These alternative bit combinations all have no fitness contribution i.e. their fitness is zero. For example, consider schema S0: 1111111**.....**, which is a schema defined for the first eight loci out of 64. The arbitrary schemata 11110111**.....** and 00101100**.....** are alternative schemata for the first eight loci. There are 255 alternative schemata during the dominance values although they may contain a large proportion of 1's. Selective crossover has to rely on randomly finding a level 0 schema before it can begin to exploit the fit schemata. Hence, this requires a great deal of exploration.

8.3 Schema Survival Probability for Selective Crossover

Selective crossover and uniform crossover have two common characteristics. The first is that crossover occurs at each locus and the second is that they have no positional bias. Several studies (Syswerda, 1989; Eshelman, Caruana, Schaffer, 1989; Booker, 1992; Spears, 1998) have confirmed that uniform crossover has no positional bias and the results in the previous section confirmed our original expectation that selective crossover has no positional bias.

Given these two characteristics, Spears (1998) constructed a schema survival theory for uniform crossover, which was outlined in Section 3.5. Since selective crossover also shares these two characteristics the question is 'Can the schema survival theory, presented by Spears, be extended for selective crossover?'

To answer this question we begin by identifying the differences between uniform and selective crossover. Let us consider two individuals, Parent1 and Parent2, chosen for recombination using uniform crossover. For the moment let us ignore the allele values taken up by both parents. The two individuals can be given as:

Locus	А	В	С	D
Parent1				
Parent2				

In uniform crossover the probability of exchange at any locus A, B, C or D is P_0 (0.5). Now let us consider two individuals, Parent1 and Parent2, chosen for recombination using selective crossover. Again let us ignore the allele values taken up by both parents. These two individuals can be given as:

Locus	А	В	С	D
Parent1	$d_A^{P_1}$	$d_B^{P_1}$	$d_{C}^{P_{1}}$	$d_D^{P_1}$
Parent2	$d_A^{P_2}$	$d_B^{P_2}$	$d_C^{P_2}$	$d_D^{P_2}$

In selective crossover the exchange at any locus is deterministic and occurs when the dominance value of a locus in Parent1 is less than the dominance value of the same locus in Parent2. For example, exchange can only occur at locus B when $d_B^{P_1} < d_B^{P_2}$ (using the same notation as in Section 4.4), where $d_B^{P_1}$ is the dominance value at locus B belonging to Parent1 and $d_B^{P_2}$ is the dominance value at locus B belonging to Parent2. Thus in selective crossover, exchange at any loci depends on the dominance values held in Parent1 and Parent2.

The exchange of an allele in uniform crossover is independent of any values held in the parents or more generally the population, whereas the exchange of an allele in selective crossover is entirely dependent on the current dominance values held in the parents or more generally the population. Selective crossover is dependent on the *current* dominance values in the population because the dominance values are changed in each generation to reflect correlations between parental and offspring fitnesses. Thus, the distribution of dominance values in the population is constantly changing with respect to the generations, as was demonstrated in Chapter 6.

To construct a schema survival probability for the current generation, let us assume that individuals are fixed length strings whose characters are from a finite-cardinality alphabet. Let the length be l and the cardinality be G. Therefore if the alphabet is the set of bits in a binary string, then G = 2 and two possible strings of length l = 4 are "1100" and "0000". This gives us G^l possible strings.

Let H_k denote a schema, or hyperplane, of order k. H_k represents G^{l-k} possible strings, where the strings must match on the k defining positions of H_k . (see Section 2.5). For example, $H_2 = *11^*$ is a second-order hyperplane that represents the 4 strings of length four that contain "11" in the middle of the strings, viz. "1111", "1110", "0111" and "0110". These four strings are instances of H_2 .

To construct a survival probability for a hyperplane H_k let us suppose that an individual containing H_k is chosen to be Parent1 from a population of n individuals. For exchange to occur a second parent, Parent2, must be chosen from the remaining n-1 individuals, which is done randomly. Given that the second parent, Parent2, is chosen; there are **at least** two cases when H_k survives:

- 1. When all k defining positions in H_k are exchanged. This takes place when Parent1 has lower dominance values at all k defining positions than Parent2.
- 2. When all k defining positions in H_k are not exchanged. This takes place when Parent1 has higher or equal dominance values at all k defining positions than Parent2.

Since recombination using selective crossover is deterministic and all dominance values are known in the population, we can determine the number of individuals that match the above two cases. For example, consider the mating pool below. To find the survival probability for a second order hyperplane H_2 "*11*", suppose that an individual containing hyperplane H_2 is chosen to be Parent1 from a mating pool of 5 individuals. The individual chosen to be Parent1 is shaded in the mating pool below. As we are only

interested in the hyperplane "*11*" we do not need to know what the other dominance values or allele values at loci A and D are; hence they are depicted with the 'don't care' symbol '*' in the example mating pool.

Individual	Locus					
	А	В	С	D		
1	*	1.4	0.9	*		
	*	1	1	*		
2	*	2.0	2.3	*		
	*	1	0	*		
	L					
3	*	0.1	0.7	*		
	*	0	1	*		
L	l	1				
4	*	1.9	0.6	*		
	*	0	0	*		
	I	1		·		
5	*	1.7	0.7	*		
	*	1	0	*		
			1			

The second parent, Parent2, is chosen randomly and can potentially be any one of the four remaining individuals i.e. 2, 3, 4 or 5. Thus to compute the probability of the survival of H_2 we compare the dominance values of Parent1 with all other individuals in the mating pool and count the number of occasions when Case 1 or Case 2 takes place.

From our example mating pool Case 1 takes place when individual 2 is chosen to be Parent2. Case 2 takes place when individual 3 is chosen to be Parent2. Thus the probability of Case 1 taking place is $\frac{1}{4}$ and Case 2 taking place is also $\frac{1}{4}$. Considering these two cases the probability $P_s(H_2)$ of H_2 surviving selective crossover is $\frac{1}{2}$. However H_2 also survives when individual 5 is chosen to be the second parent because they share common alleles at those positions where exchange occurs, as demonstrated below:

Applying recom	bination	to indiv	vidual 1	and 5.
----------------	----------	----------	----------	--------

	Individual	Locus				
		А	В	С	D	
Parent1	1	*	1.4	0.9	*	
		*	1	1	*	
Parent2	5	*	1.7	0.7	*	
		*	1	0	*	

Offspring after crossover:

	Individual	Locus				
		А	В	С	D	
Child1	1	*	1.7	0.9	*	
		*	1	1	*	
Child2	5	*	1.4	0.7	*	
		*	1	0	*	

Exchange occurred at locus B but H_2 survived in Child1 as individual 5 also contained a 1 at locus B. hence, the survival probability also depends on the common alleles found in the parents – known as homogeneity (Spears 1998). Taking this into consideration the final probability of H_2 surviving $P_s(H_2)$ is ³/₄.

In general suppose that crossover results in x of the k defining positions being exchanged, then a hyperplane H_k survives if any of the following are true:

- 1. All k defining positions in H_k are exchanged i.e. x = k.
- 2. All *k* defining positions in H_k are not exchanged i.e. x = 0.
- 3. The parents' alleles match at the *x* positions being exchanged.
- 4. The parents' alleles match at the k x positions not being exchanged.

The exact survival probability of H_k , $P_s(H_k)$ can be computed for a mating pool of n individuals, by applying selective crossover to the instance of H_k , individual i, and all other (n-1) individuals in the current mating pool and counting the number of survivals. This gives us a survival probability for H_k as:

$$P_{s}(H_{k}) = \frac{1}{n-1} \sum_{\substack{j=1 \ j \neq i}}^{n} SC(i, j, H_{k})$$
(8.1)

Where SC(i, j, k) is a function that applies selective crossover to individuals *i* and *j* at all *k* defining positions and returns a 1 if H_k survives and 0 otherwise, as given below:

$$SC(I^{P_1}, I^{P_2}, H_k) = \begin{cases} 1, & \text{if } m_i^1 = 1 \quad \forall i \mid h_{k_i} \neq `*` \\ 1, & \text{if } m_i^1 = 0 \quad \forall i \mid h_{k_i} \neq `*` \\ 1, & \text{if } g_i^{P_1} = g_i^{P_2} \quad \forall i \mid h_{k_i} \neq `*` \land m_i^1 = 1 \\ 1, & \text{if } g_i^{P_1} = g_i^{P_2} \quad \forall i \mid h_{k_i} \neq `*` \land m_i^1 = 0 \\ 0, & \text{otherwise} \end{cases}$$

Where:

$$H_{k} = (h_{k_{1}}, \dots, h_{k_{i}}, \dots, h_{k_{i}}) \text{ where } h_{k_{i}} \in \{*, 0, 1\}$$

 I^{P_1} refers to an individual with a gene vector G^{P_1} and a dominance vector D^{P_1} (see Section 4.5 for notation).

 m_i^1 is an element at the *i*th position of the inheritance mask (see Section 4.5). A 1 in the inheritance mask indicates a crossover position. As the inheritance mask is generated using the dominance values schema survival is dependent on the current dominance values in the population.

Since schema survival for selective crossover is dependent on the dominance values in the population it will vary from problem to problem because they are updated using the problem's fitness function. It is therefore not possible to produce a schema survival probability that is applicable to all problems as presented by Spears (1998) for uniform crossover.

In summary, the survival probability of a schema, which is chosen to be a parent, is the probability of choosing another parent from the population that allows it to survive. This other parent must either have lower, higher or equal dominance values at all defining positions of a schema or it has common alleles where exchange would take place. Since exchange of an allele under selective crossover depends on the dominance values and these are updated using the fitness function (Equation 4.4 in Chapter 4); the survival probability of a schema is problem dependent. For example, the survival probability of schema "**1*1*1" for the One Max problem will not be the same for the Royal Road function.

8.4 Summary and Conclusions

This chapter described three alternative encodings for the Royal Road function. The original encoding consists of tightly coupled level 0 schemata, which have a defining length of 7. Increasing the defining length of the level 0 schemata to 14, 28 and 56 systematically reduced this tight coupling or genetic linkage.

These alternative encodings were used to demonstrate the effect an encoding has on the performance of two-point, uniform and selective crossover. The results demonstrated that two-point crossover is very sensitive to the encoding, whilst uniform and selective crossover are not. The performance of two-point deteriorated as the defining length of the level 0 schemata increased. The performance of uniform and selective crossover remained consistent regardless of the defining length of level 0 schemata. Both operators guarantee behaviour on any encoding of the Royal Road function; however, the performance of selective crossover is better than uniform crossover.

The Royal Road encodings were also used to study schema propagation under two-point, uniform and selective crossover to:

- 1. Provide us with a better understanding of the relationship between the encoding and performance;
- 2. Provide more insight into the underlying behaviour of the recombination operators.

At each generation, of a run, the mean number of level 0 schemata successfully transmitted to the offspring from a parent by recombination was observed. This showed

that two-point crossover was more successful at propagating schemata which had smaller defining lengths; thus confirming that two-point crossover has high positional bias. Uniform crossover has no positional bias and was able to propagate schemata at equal rates independent of the defining length. Selective crossover was also able to propagate schemata at equal rates independent of the defining length, which indicates that selective crossover also has no positional bias.

The rate of schema propagation was compared with two-point and uniform crossover. This demonstrated that selective crossover propagates less schemata in early generations than two-point or uniform crossover. This gives some indication that selective crossover is more explorative in early generations, showing a lower schema survival rate, which is ideal whilst diversity still exists in the population. However in subsequent generations it is more exploitative with more schemata propagated than two-point and uniform crossover. Thus selective crossover seems to provide a better balance between exploration and exploitation than two-point and uniform crossover.

Section 8.3 constructed a schema survival probability for selective crossover and identified that schema survival under selective crossover is dependent on the current dominance values in the population. The probability of a schema surviving can be computed by applying selective crossover to an instance of the schema and all other individuals in the population and counting the number of survivals. This schema survival probability is the probability of choosing an individual from the population, to be a parent, which permits survival of the schema given its dominance values and common allele values.

In conclusion, recombination operators with high positional bias are less disruptive against adjacent genes; therefore, to exploit this behaviour epistatic genes must be placed close to each other on the chromosome when encoding the problem. The performance of selective crossover is not affected by the encoding of the problem and has no positional bias. Although two-point crossover outperformed selective crossover on the Royal Road function in Chapter 5, its performance is affected by the encoding of the problem as was demonstrated in this chapter. When the defining length of the schema was increased in the encoding the performance of two-point crossover decreased and the rate at which it propagated schemata also decreased. Selective crossover allows exploitation of good schemata regardless of their defining length; hence if a schema consists of interacting genes at the two extremes of the chromosome, it can be propagated as easily as a schema

with interacting genes located adjacent to each other. Selective crossover is very much like uniform crossover in that it has no positional bias but to its advantage it does not have the high disruptive qualities of uniform crossover and is able to provide a better balance between exploration and exploitation.

Chapter 9

Conclusions

The goal of this thesis was to design, implement and evaluate selective crossover, an adaptive recombination operator for use in genetic algorithms. Selective crossover was designed with motivation from natural evolution and as a general recombination operator for use with a wide range of problems as opposed to a specialised recombination operator for use with a small set of problems.

This chapter first provides a summary and discussion of the results obtained in this thesis followed by a restatement of the contributions made and a statement of the limitations of this thesis. Finally several directions for future research are detailed.

9.1 Discussion of Results

In this section we tie together the results and the thesis hypothesis re-stated below:

When little or no knowledge is available about the problem being optimised by a genetic algorithm, a viable strategy is to use an adaptive recombination operator with the following three properties:

- 1. **Detection** It detects alleles that were changed during recombination to identify modifications to the candidate solution.
- 2. Correlation It uses correlations between parental and offspring fitnesses as a means of discovering beneficial alleles.

3. **Preservation** - It preferentially preserves alleles at each locus, during recombination, according to their previous contributions to beneficial changes in fitness.

Selective crossover was designed with these three properties and evaluated in terms of performance, adaptation, biases and schema propagation. To summarise, selective crossover is an allele-based adaptive recombination operator that uses a dominance vector to bias alleles, during recombination, that have increased an individual's fitness in previous generations. A dominance vector accompanies each chromosome such that each allele in the chromosome has an associated dominance value. It is these dominance values that dictate where crossover should occur and act as a means of storing knowledge about parental and offspring fitness correlations in previous generations which is later exploited during recombination to prevent alleles from crossing over.

The performance of selective crossover was evaluated and compared with two static recombination operators, two-point and uniform crossover, that do not possess the above three properties. The results in Chapter 5 demonstrated that selective crossover shows superior or equal performance on nearly all the test problems used. On simple problems, such as the One Max problem, selective crossover worked equally well as two-point and uniform crossover. However, on the Deceptive Trap functions (with tuneable deception), selective crossover, like two-point and uniform crossover, was unable to solve problems with deception of order 20 or above.

Two-point crossover outperformed selective crossover on the Royal Road problem; however, the study on schema propagation in Chapter 8 confirmed that twopoint crossover has a positional bias and is thus sensitive to the encoding of the problem. For two-point crossover to exhibit superior performance and to propagate schemata successfully the interacting genes must be situated together on the chromosome. To do this a priori knowledge is required about the problem being solved; the interacting genes must first be identified and then encoded so that it can be exploited by two-point crossover. In contrast selective crossover has no positional bias. The performance and schema propagation of selective crossover remained consistent with any encoding of the Royal Road function; there is no need to tune the encoding for selective crossover as its behaviour is consistent. A comparison of schema propagation also suggests that selective crossover provides a better balance between exploration and exploitation than two-point and uniform crossover. Selective crossover exhibits more exploration in early generations showing a lower schema survival rate, which is ideal whilst diversity still exists in the population. Moreover, it is more exploitative in later generations with more schemata propagated than two-point and uniform crossover.

All three operators were compared in Chapter 5 using problems with tuneable epistasis, namely NK landscapes and L-MaxSAT problems. The NK landscapes are a well-studied family of landscapes where the tuneable epistasis also allows control over the ruggedness of the search landscape. On the tested levels of epistasis selective crossover outperformed two-point and uniform crossover.

An analysis of the adaptive properties in Chapter 6 identified that selective crossover contains two features of adaptation: adaptive and self-adaptive (using the terminology of Eiben *et al.*, 1999). It is the adaptive feature that holds the three key properties: detection, correlation and preservation. By omitting the adaptive feature in selective crossover, the performance of selective crossover deteriorated and was worse than two-point and uniform crossover. These results suggested that the three key properties are essential to the performance of selective crossover. Chapter 6 also examined the behaviour of the dominance values in the population and suggested that selective crossover behaves differently to reflect the problem being solved; thus demonstrating dynamic behaviour as opposed to the static behaviour of two-point and uniform crossover.

The study of biases in Chapter 7 identified four key biases in selective crossover: directional, credit, hitchhiker and initialisation bias. Directional bias exists if alleles are either favoured or not favoured for their fitness contribution. Credit bias is the degree at which an allele gets favoured with respect to its fitness contribution. Initialisation bias exists if alleles get favoured during initialisation without knowing their fitness contribution. Hitchhiker bias exists if alleles get favoured when they do not contribute to the fitness increase. All four biases are interdependent; the credit bias is a direct result of the directional bias, and the hitchhiker bias is a consequence of the credit and initialisation biases. We demonstrated that eliminating the initialisation bias further increased the performance of selective crossover; thus identifying that the hitchhiker bias is an unavoidable consequence of the credit bias and cannot be eliminated completely without a trade-off in performance (as was demonstrated in Section 6.2); a small reduction in the credit bias had no effect on performance. This study also suggested that selective crossover is unable to solve highly deceptive problems due to the directional bias.

A schema survival probability for selective crossover was constructed in Chapter 8. This demonstrated that the schema survival probability is dependent on the dominance and allele values of the parents chosen for crossover. Since the exchange at any locus is deterministic the exact probability of survival can be calculated given the current population of individuals.

The results of this thesis jointly suggest that selective crossover is a viable strategy to use when little or no knowledge is known about the problem being solved and thus verifies our hypothesis. However, owing to the limitations of this study (outlined in Section 9.3), we do not claim that selective crossover or any other strategy that may possess the three properties is *the best* one to use when no knowledge is available about the problem being solved. In conclusion it is hoped that this work will aid designers of genetic algorithms for the 'real world' in their choice of recombination operators.

9.2 Contributions of this Thesis

This thesis makes six primary contributions:

- 1. The design and implementation of *"Selective Crossover"*, a new adaptive recombination operator that incorporates correlations between parents and offspring as a means of discovering and preserving beneficial alleles at each locus during recombination to produce fitter offspring. This was described in Chapter 4.
- A measurement and comparison of the performance of selective crossover and two traditional recombination operators on a number of different problems. This was provided in Chapter 5.
- 3. An empirical analysis that demonstrates adaptive behaviour in selective crossover. This was provided in Chapter 6.

- 4. An identification of four key biases inherent in selective crossover, a demonstration of the existence of these biases in two other similar operators and an empirical analysis to study the effects of these biases on selective crossover. This was described in Chapter 7.
- 5. An empirical analysis and comparison of the effects an encoding has on the performance and schema propagation in selective crossover and two traditional recombination operators. This was provided in Chapter 8.
- 6. A construction of a schema survival probability for selective crossover; this unavoidably relies on the dominance values, which in turn rely on the problem and demonstrates that schema survival in selective crossover is problem dependent. This was provided in Chapter 8.

This thesis also makes a secondary contribution which was described in Chapter 6; a description of a new taxonomy to classify selective crossover and other adaptive strategies in evolutionary computation, which overcomes the limitations in the existing taxonomy provided by Eiben *et al.* (described in Chapter 3).

9.3 Limitations of this Thesis

There are three major limitations to this work:

• The set of benchmark problems used; although benchmark problems serve as good problems to test new strategies they can only capture limited characteristics that may exist in 'real world' problems. In our experiments we used a set of well-studied benchmark problems that provide a good variation of features in real world problems; these were epistasis, deception and fitness landscapes. However, these problems were limited to binary encodings and for some problems it is most natural to use a non-binary alphabet for their encoding (Davis 1991). Thus, our set of benchmark problems has allowed us to examine a small sample of real-world problems and further studies would allow us to examine the effects of epistasis, deception and fitness landscapes on problems that require non-binary alphabets.

- Selective crossover was only compared against two static recombination operators and was not compared against masked crossover (Louis and Rawlins, 1991) and adaptive uniform crossover (White and Oppacher, 1994). We identified in Chapter 7 the similarities and differences between masked and adaptive uniform crossover; the similarities being (i) all three operators use parental and offspring fitness correlations to bias alleles during recombination and (ii) all three operators are allele-based adaptive recombination operators. The differences between the operators lie in the way they administer the fitness correlations (update their strategy parameter) to bias alleles. An analytical comparison in terms of biases was made between the three operators; however quantitative measures on performances were not made and would provide more insight into the relationship between these operators and their biases. Hence, we do not know if selective crossover is better than masked or adaptive uniform crossover.
- Selective crossover was only compared against two static recombination operators that do not possess the three key properties stated in the hypothesis: detection, correlation and preservation. In Chapter 6 we applied a version of selective crossover that does not possess the three properties and demonstrated that selective crossover needs the three properties to stop it from converging to sub-optimal solutions. However selective crossover was not compared against *other adaptive* recombination operators that do not possess the three properties such as punctuated crossover (Schaffer and Morishima 1987). Thus we cannot conclusively state that the use of the three properties makes selective crossover better than any other adaptive strategy.

9.4 Future Work

The development of a new operator opens many doors for research, some of which have been tackled in this thesis. However, there still remain multiple directions for future work. This proposed work is aimed at overcoming limitations of the original research and enhancing the understanding of selective crossover through additional analyses. There are three main areas of extension to this work: application, comparative analyses and quantitative analyses.

Application

Extending the application of selective crossover to other problems exposes it to other characteristics found in problems that were not covered by the test problems used in this thesis. Application to problems includes those that require non-binary encodings (such as ordering problems) and a real-world problem. This would allow us to determine if the results in this thesis are widely applicable.

Comparative analyses

Extending the comparative analyses allows us to analyse qualitatively the performance of selective crossover in relation to alternative strategies other than static recombination operators. The comparative analyses include comparing performances of selective crossover with:

- Other adaptive recombination operators (such as masked crossover, adaptive uniform crossover and punctuated crossover).
- Other techniques and operators such as those that adapt mutation or recombination probabilities.
- Landscape neighbourhood operators such as steepest ascent hill climbing.
- Other search methods such as simulated annealing.

Quantitative analyses

The use of quantitative analyses would allow us to extend our understanding of the behaviour exhibited by selective crossover. This would include quantitative analyses of:

• The behaviour displayed by the dominance values; this would entail observing the distribution of dominance values across many runs and different problems and performing a cross-correlation between the distributions to identify any reoccurring patterns of behaviour from selective crossover

- The resilience of selective crossover to other parameters of the genetic algorithm such as the recombination rate, mutation rate, selection scheme and population size. As mentioned in Section 2.4, choosing appropriate parameter settings is difficult and greatly influences the search capabilities of the GA. Systematically varying these parameters and applying selective crossover will allow us to understand how the behaviour of selective crossover is affected by other parameter settings.
- The biases in masked and adaptive uniform crossover and their relationship with selective crossover. This would entail: (i) comparing the performances of masked, adaptive uniform and selective crossover, (ii) analysing the effect of applying an alternative credit mechanism (described in Section 7.4.1) to masked and adaptive uniform crossover and (iii) analysing the effect of applying an alternative initialisation method (described in Section 7.4.2) to masked and adaptive uniform crossover.

Bibliography

- Ackley, D. H. (1987) A Connectionist Machine for Genetic Hillclimbing. Boston, Kluwer Academic Publishers.
- Altenberg, L. (1998) Personal Communication.
- Angeline, P.J. (1995) Adaptive and Self-adaptive Evolutionary Computations. In M. Palaniswami et al. (eds.), Computational Intelligence: A Dynamic System Perspective, 152-163, IEEE Press.
- Antonisse, J. (1989) A New Interpretation of the Schema Notation that Overturns the Binary Encoding Constraint. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, 86-91. Morgan Kaufmann.
- Arabas, J., Mulawka, J. and Pokrasniewicz, J. (1995) A new class of the crossover operators for the numerical optimization. In L. Eshelman, editor, *Proceedings of* the Sixth International Conference on Genetic Algorithms, 42-48. Morgan Kaufmann.
- Bäck, T (1992a) The Interaction of Mutation Rate, Selection and Self-Adaptation Within a Genetic Algorithm. In R. Manner and B. Manderick (eds.), Proceedings of the Second Conference on Parallel Problem Solving from Nature, 85-94, Elsevier Science.

Bäck, T (1992b) Self Adaptation in Genetic Algorithms. In F. Varela and P. Bourgine

(eds.), Towards a Practice on Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 263-271. MIT Press.

- Bäck, T (1993) Optimal Mutation Rates in Genetic Search. In S. Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, 2-8. Morgan Kaufmann.
- Bäck, T (1996) Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press.
- Baker, J. E. (1985) Adaptive Selection Methods for Genetic Algorithms. In J. J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, 101-111. Lawrence Erlbaum Associates.
- Baker, J. E. (1987) Reducing Bias and Inefficiency in the Selection Algorithm. In J.J Grefenstette, editor, Proceedings of the Second International Conference on Genetic Algorithms, 14-21. Lawrence Erlbaum Associates.
- Bergman, A., Otto, S. P. and Feldman, M. W. (1995a) On Evolution of Recombination in Haploids and Diploids: 1. Deterministic Models. In *Complexity*, Vol. 1, no. 1, 57-67, John Wiley.
- Bergman, A., Otto, S. P. and Feldman, M. W. (1995a) On Evolution of Recombination in Haploids and Diploids: 2. Stochastic Models. In *Complexity*, Vol. 1, no. 2, 49-57, John Wiley.
- Blicke, T. and Thiele, L. (1995) A Mathematical Analysis of Tournament Selection. In L. Eshelman, editor, Proceedings of the Sixth International Conference on Genetic Algorithms, 9-16. Morgan Kaufmann.
- Booker, L. B. (1992) Recombination Distribution for Genetic Algorithms. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, 29-44. Morgan Kaufmann.
- Caruana, R. A. and Schaffer, J. D. (1988) Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms. In J. Laird, editor, *Proceedings of the*

Fifth International Conference on Machine Learning, 153-161. Morgan Kaufmann.

- Cohen, P. R. (1995) Empirical Methods for Artificial Intelligence. Cambridge, MIT Press.
- Davis, L. (1989) Adapting Operator Probabilities in Genetic Algorithms. In J. D. Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, 61-69. Morgan Kaufmann.
- Davis, L. (1991) (editor) The Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York.
- Dawkins, R. (1989) The Selfish Gene. Oxford University Press, UK.
- De Jong, K. A. (1975) An analysis of the Behavior of a Class of Genetic Adaptive Systems, Doctoral Thesis, University of Michigan, Ann Harbor, USA.
- De Jong, K. A., and Spears, W. M. (1990) An Analysis of the Interacting Roles of Population Size and Crossover. In H.-P Schwefel and R. Männer (eds.), Proceedings of the First International Workshop on Parallel Problem Solving from Nature, 38-47. Springer-Verlag.
- De Jong, K. A., Potter, M. A. and Spears, W. M. (1997) Using Problem Generators to Explore the Effects of Epistasis. In T. Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, 338-345. Morgan Kaufmann.
- Deb K. and Goldberg D. E. (1993). Analyzing Deception in Trap Functions. In L. D. Whitley, editor, Foundations of Genetic Algorithms 2, 93-108. Morgan Kaufmann.
- Eiben, A. E., Hinterding, R., Michalewicz, Z. (1999) Parameter Control in Evolutionary Algorithms. In *IEEE Transactions on Evolutionary Computation*, Vol 3, no. 2, 124-141.
- Eshelman, L. J., Caruana, R. A., and Schaffer J. D. (1989) Biases in the Crossover Landscape. In J. D. Schaffer, editor, *Proceedings of the Third International*

Conference on Genetic Algorithms, 10-19. Morgan Kaufmann.

- Eshelman, L. J. and Schaffer J. D. (1993) Crossover's Niche. In S. Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, 9-14. Morgan Kaufmann.
- Eshelman, L. J. and Schaffer J. D. (1995) Productive Recombination and Propagating and Preserving Schemata. In L. D. Whitley and M. D. Vose (eds.) Foundations of Genetic Algorithms 3, 299-313. San Francisco, Morgan Kaufmann.
- Falkenauer, E. (1994) A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems. *Evolutionary Computation*, Vol 2, no. 2, 123-144, Cambridge, MIT Press.
- Fogel, L. J., Owens, A. J. and Walsh, M. J. (1966) Artificial Intelligence Through Simulated Evolution. John Wiley.
- Forrest, S. and Mitchell, M. (1993) Relative Building-Block Fitness and the Building Block Hypothesis. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, 109-126. San Francisco, Morgan Kaufmann.
- Goldberg, D. E. (1989a) Genetic Algorithms in search, optimization and machine learning. Addison-Wesley.
- Goldberg, D. E. (1989b) Sizing Populations for Serial and Parallel Genetic Algorithms. InJ. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, 70-79. Morgan Kaufmann.
- Goldberg, D. E. (1989c) Genetic Algorithms and Walsh functions: Part I, A Gentle Introduction. *Complex Systems*, Vol. 3. 129-152.
- Goldberg, D. E. (1989d) Genetic Algorithms and Walsh functions: Part II, Deception and Its Analysis. *Complex Systems*, Vol. 3. 153-171.
- Goldberg, D. E. and Deb, K. (1991) A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, 69-93. Morgan Kaufmann.

- Goldberg, D. E., Korb, B. and Deb, K. (1989) Messy Genetic Algorithms: Motivation, Analysis, and First Results. In *Complex Systems*, Vol. 3. 493-530.
- Goldberg, D. E. and Lingle, R. (1995) Alleles Loci and the Travelling Salesman Problem.
 In J. J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, 154-159. Lawrence Erlbaum Associates.
- Grefenstette, J. J. (1986) Optimization of Control Parameters for Genetic Algorithms. In *IEEE Transations on Systems, Man and Cybernetics*, Vol. 16, no. 1, 122-128.
- Grefenstette, J. J. (1991) Conditions for Implicit Parallelism. In G. Rawlins, editor, Foundations of Genetic Algorithms, 252-261. Morgan Kaufmann.
- Grefenstette, J. J. (1993) Deception Considered Harmful. In L. D. Whitley, editor, Proceedings of Foundations of Genetic Algorithms 2, 75-91. Morgan Kaufmann.
- Grefenstette, J. J. and Baker, J. E. (1989) How Genetic Algorithms Work: A Critical Look at Implicit Paralellelism. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, 20-27. Morgan Kaufmann.
- Harik, G. R. (1997) Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms. Doctoral dissertation, University of Michigan, Ann Harbor, USA.
- Hinterding, R., Michalwicz, Z. and Eiben A. E. (1997) Adaptation in Evolutionary Computation: A Survey. In Proceedings of the IEEE International Conference on Evolutionary Computation, 65-69. IEEE Press.
- Holland, J. H. (1992) Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, 1992. First published by University of Michigan Press 1975.
- Kargupta, H. (1996) The Gene Expression Messy Genetic Algorithm. In Proceedings of the IEEE International Conference on Evolutionary Computation, 814-819 IEEE Press.

- Kauffman S. A. (1993) The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press.
- Koza, J. R. (1992) Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press.
- Louis, S. J. and Rawlins, G. J. E. (1991) Designer Genetic Algorithms: Genetic Algorithms in Structure Design. In R. Belew and L. Booker (eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, 53-60. Morgan Kaufmann.
- Macready, W. G. and Wolpert D. H. (1996) On 2-armed Gaussian Bandits and Optimization. Technical Report SFI-TR-96-03-009, Santa Fe Institute, Santa Fe, New Mexico.
- Merrell, D. J. (1994) *The Adaptive Seascape: The Mechanism for Evolution*. University of Minnesota Press, Minneapolis.
- Michalewicz, Z. (1994) Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag. First Published by Springer-Verlag in 1992.
- Mitchell, D., Selman, B. and Levesque, H. (1992) Hard and Easy Distributions of SAT Problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 459-465. MIT Press.
- Mitchell, M. (1996) An Introduction to Genetic Algorithms. MIT Press.
- Mitchell, M. and Forrest, S. (1996) Fitness Landscapes: Royal Road Functions. In T. Bäck, D. B. Fogel and Z. Michaelewicz (eds.), *Handbook of Evolutionary Computation*. Oxford University Press.
- Mitchell, M. and Forrest, S. and Holland, J. H. (1991) The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance. In F. J. Verala and P. Bourgine (eds.), Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on artificial Life, 245-254. MIT Press.

Radcliffe, N. J. (1992) Non-Linear Genetic Representations. In R. Manner and B.

Manderick (eds.), Proceedings of the Second Conference on Parallel Problem Solving from Nature, 259-268, Elsevier Science.

- Rana, S (1999) The Distributional Biases of Crossover Operators. In W. Banzhaf et al. (eds.), Proceedings of the Genetic and Evolutionary Computation Conference, 549-556. Morgan Kaufmann.
- Rana, S. and Whitley, D. (1998) Genetic Algorithm Behavior in the MAXSAT Domain.
 In A. E. Eiben et al. (eds.), Proceedings of the Fifth Conference on Parallel Problem Solving from Nature, 785-794. Springer-Verlag.
- Reeves, C. R. (1993) Using Genetic Algorithms with Small Populations. In S. Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, 92-99. Morgan Kaufmann.
- Reeves, C. R. and Wright, C. C. (1999) Genetic Algorithms and The Design of Experiments. In L. D. Davis *et al.* (eds.), *Evolutionary Algorithms*: IMA Volumes in Mathematics and its Applications, Vol. 111, 207-226. Springer-Verlag.
- Rice, J. A. (1995) Mathematical Statistics and Data Analysis 2nd Edition. Duxbury Press.
- Schaffer, J. D., Caruana, R. A, Eshelman, L. J. and Das, R. (1989) A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimisation. In J.D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, 51-60. Morgan Kaufmann.
- Schaffer, J. D. and Eshelman L.J. (1991) On Crossover as an Evolutionary Viable Strategy. In R. Belew and L. Booker (eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, 61-68. Morgan Kaufmann.
- Schaffer, D. J., Eshelman, L.J. and Offnut, D. (1991) Spurious Correlations and Premature Convergence in Genetic Algorithms. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, 102-112. Morgan Kaufmann.

Schaffer, J. D. and Morishima, A. (1987) An Adaptive Crossover Distribution Mechanism

for Genetic Algorithms. In J. J Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, 36-40. Lawrence Erlbaum Associates.

- Schwefel, H. P. (1981) Numerical Optimisation of Computer Models. John Wiley and Sons.
- 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. In L. D. Whitley, editor, *Proceedings of Foundations of Genetic Algorithms 2*, 221-237. Morgan Kaufmann.
- Spears, W. M. (1995) Adapting Crossover in Evolutionary Algorithms. In J. McDonnell,
 R. Reynolds and D. Fogel (eds.), *Proceedings of the Fourth Annual Evolutionary Programming Conference*, 367-384, MIT Press.
- Spears, W. M. (1997) Recombination Parameters. In T. Bäck, D. B. Fogel and Z. Michaelewicz (eds.), *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, USA.
- Spears, W. M. and De Jong, K. A. (1991a). An Analysis of Multi-Point Crossover. In J.
 E. Rawlins, editor, *Proceedings of Foundations of Genetic Algorithms*, 301-315.
 Morgan Kaufmann.
- Spears, W. M. and De Jong, K. A. (1991b). On the Virtues of Parameterized Uniform Crossover. In R. Belew and L. Booker (eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, 230-236. Morgan Kaufmann.
- Starkweather, T., McDaniel, S., Mathias, K., Whitley, D. and Whitley, C. (1991) A Comparison of Genetic Sequencing Operators. In R. Belew and L. Booker (eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, 69-79. Morgan Kaufmann.

- Syswerda G. (1989) Uniform Crossover in Genetic Algorithms. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, 10-19. Morgan Kaufmann.
- Syswerda G. (1993) Simulated Crossover in Genetic Algorithms. In L. D. Whitley, editor, Foundations of Genetic Algorithms 2, 239-255. Morgan Kaufmann.
- Thierens, D. and Goldberg, D (1994) Convergence Models of Genetic Algorithm Selection Schemes. In L. Davidor, editor, *Proceedings of the Third Conference Parallel Problem Solving from Nature Conference*, 119-129. Springer Verlag.
- Tuson, A. and Ross, P. (1998) Adapting Operator Settings in Genetic Algorithms. In Evolutionary Computation, Vol. 6, no. 2, 161-184.
- Vekaria K. and Clack C. (1998a). Selective Crossover in Genetic Algorithms. In Koza et al. (eds.), Genetic Programming 1998: Proceedings of the Third Annual Conference, 609. Morgan Kaufmann
- Vekaria, K. and Clack, C. (1998b) Selective Crossover in Genetic Algorithms: An Empirical Study. In A. E. Eiben et al. (eds.), Proceedings of the Fifth Conference on Parallel Problem Solving from Nature, 438-447. Springer-Verlag.
- Vekaria K. and Clack C. (1999a). Biases Introduced by Adaptive Recombination Operators. In W. Banzhaf et al. (eds.), Proceedings of the Genetic and Evolutionary Computation Conference, 670- 677. Morgan Kaufmann.
- Vekaria K. and Clack C. (1999b). Schema Propagation in Selective Crossover. Late breaking papers at the Genetic and Evolutionary Computation Conference, July 13-17, Orlando, Florida. pp. 268-275
- Vekaria K. and Clack C. (1999c). Hitchhikers Get Around. In Evolution Artificielle (EA) 1999, November 3-5, LIL, Universite du Littoral, Dunkerque, France.
- Vekaria K. and Clack C. (2000). Royal Road Encodings and Schema Propagation in Selective Crossover. In Y. Suzuki et al. (eds.), Soft Computing in Industrial Applications. Springer Verlag (Presented in the 4th Online World Conference on

Soft Computing in Industrial Applications (WSC4), September 21-30, 1999.

- White, T. and Oppacher, F. (1994) Adaptive Crossover Using Automata. In L. Davidor, editor, Proceedings of the Third Conference Parallel Problem Solving from Nature Conference, 229-238. Springer Verlag.
- Whitley, L. D. (1991) Fundamental Principles of Deception in Genetic Search. In G.
 Rawlins, editor, *Foundations of Genetic Algorithms*, 221-241. Morgan Kaufmann.
- Wolpert, D. H. and Macready, W. G. (1997) No Free Lunch Theorems for Optimization. In *IEEE Transactions on Evolutionary Computation*, Vol 1, no. 1, 67-82.
- Wu, S., Lindsay, R.K. and Riolo, R.L. (1997) Empirical Observations on the Role of Crossover and Mutation. In T. Bäck, editor, Proceedings of the Seventh International Conference on Genetic Algorithms, 362-369. Morgan Kaufmann.

Appendix A: Publications

- Vekaria K. and Clack C. (2000). Royal Road Encodings and Schema Propagation in Selective Crossover. In Y. Suzuki *et al.* (eds.), Soft Computing in Industrial Applications. Springer Verlag (Presented in the 4th Online World Conference on Soft Computing in Industrial Applications (WSC4), September 21-30, 1999.
- Vekaria K. and Clack C. (1999c). Hitchhikers Get Around. In Evolution Artificielle (EA) 1999, November 3-5, LIL, Universite du Littoral, Dunkerque, France.
- Vekaria K. and Clack C. (1999b). Schema Propagation in Selective Crossover. Late breaking papers at the Genetic and Evolutionary Computation Conference, July 13-17, Orlando, Florida. pp. 268-275
- Vekaria K. and Clack C. (1999a). Biases Introduced by Adaptive Recombination Operators. In W. Banzhaf et al. (eds.), Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17, Orlando, Florida. pp. 670-677. Morgan Kaufmann.
- Vekaria, K. and Clack, C. (1998b) Selective Crossover in Genetic Algorithms: An Empirical Study. In A. E. Eiben et al. (eds.), Proceedings of the Fifth Conference on Parallel Problem Solving from Nature, Amsterdam, The Netherlands. pp. 438-447. Springer-Verlag.
- Vekaria K. and Clack C. (1998a). Selective Crossover in Genetic Algorithms. In Koza et

al. (eds.), Genetic Programming 1998: Proceedings of the Third Annual Conference, July 22-25, University of Wisconsin, Madison. pp. 609. Morgan Kaufmann.

- Clack C., Vekaria K. and Zin N. (editors) Emerging Technologies 1997: Theory and Application of Evolutionary Computation Proceedings of the 2nd Emerging Technologies Workshop (ET'97), University College London, December 15, 1997.
- Vekaria K. and Clack C. (1997). Haploid Genetic Programming with Dominance. Research Note RN/97/121, University College London, Gower Street, London WC1E 6BT.
- Vekaria K. and Clack C. (1997). Genetic Programming with Gene Dominance. In J. Koza (editor). Late Breaking Papers at the Genetic Programming 1997 Conference, pp. 300. Stanford CA:Stanford University Bookstore.