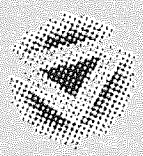
Representation and Parameterisation Issues in Genetic Algorithms

Jason Edward Price

Doctor of Philosophy



The University of Aston in Birmingham

December 1996

THE UNIVERSITY OF ASTON IN BIRMINGHAM

REPRESENTATION AND PARAMETERISATION ISSUES IN GENETIC ALGORITHMS

Jason Edward Price

Doctor of Philosophy

1996

Thesis summary

A multi-chromosome GA (Multi-GA) was developed, based upon concepts from the natural world, allowing improved flexibility in a number of areas including representation, genetic operators, their parameter rates and real world multi-dimensional applications.

A series of experiments were conducted, comparing the performance of the Multi-GA to a traditional GA on a number of recognised and increasingly complex test optimisation surfaces, with promising results. Further experiments demonstrated the Multi-GA's flexibility through the use of non-binary chromosome representations and its applicability to dynamic parameterisation. A number of alternative and new methods of dynamic parameterisation were investigated, in addition to a new non-binary 'Quotient crossover' mechanism.

Finally, the Multi-GA was applied to two real world problems, demonstrating its ability to handle mixed type chromosomes within an individual, the limited use of a chromosome level fitness function, the introduction of new genetic operators for structural self-adaptation and its viability as a serious real world analysis tool.

The first problem involved optimum placement of computers within a building, allowing the Multi-GA to use multiple chromosomes with different type representations and different operators in a single individual.

The second problem, commonly associated with Geographical Information Systems (GIS), required a spatial analysis location of the optimum number and distribution of retail sites over two different population grids. In applying the Multi-GA, two new genetic operators (addition and deletion) were developed and explored, resulting in the definition of a mechanism for self-modification of genetic material within the Multi-GA structure and a study of this behaviour.

Additional keywords: Multi-chromosome, optimisation, spatial analysis, Geographical Information Systems, dynamic parameterisation.

The path is laid – all you have to do is walk it.

For my mother,

with whom I wanted to share so much.

Acknowledgements

Acknowledgement sections seem to vary between a form of 'editorial comment' and a seemingly never ending list of names. This will be no exception.

I have to thank my supervisor, Dr. Alan Harget. Alan's constant encouragement, support, ideas and guidance throughout my time at Aston, through thick and thin, have undoubtedly been critical to my completing the Ph.D. I also have to thank Alan for suggesting the idea of research to me in the first place, without which I would never have taken this path.

Cazhaow Qazaz, Phil Barrett, Allister Pattison, Bryn Williams and John van der Rest have provided invaluable help, software support, guidance, reviewing of presentations and discussions during the course of events. Thanks are also due to Shane Murnion for providing the spatial analysis data used in chapter 6 and Mike Tipping for his help in enabling effective visualisation of the results with Matlab.

The Department of Computer Science, The Kathleen and Margaret Elliot Scholarship Trustees and ICGA committee all deserve a special mention for their financial assistance, making it possible for me to attend the 5th and 6th International Conferences on Genetic Algorithms in 1993 and 1995. The discussions with researchers at these conferences provided a wealth of ideas and inspiration embodied in my work.

Finally, I would like to thank my friends and family – particularly my cousins, Charles Adcock and Carol Simpson – along with Graeme Coombes, Katie Allen and especially Jenny Fletcher for providing the weekend breaks, friendship and laughter that kept my sanity (almost) intact over the last four years.

Most of all, I wish to thank my mother, who will regrettably never see this.

Jason E. Price

December 1996.

List of Contents

List of Tables & Figures

Chapter 1: Introduction

1.1 The place of Evolution in Computing

"Where shall I begin, please your Majesty?" he asked. "Begin at the beginning," the King said, gravely, "and go on till you come to the end; then stop."

Lewis Carroll (1865), Alice's Adventures in Wonderland.

Since man invented the wheel, human beings have been seeking to create devices to make life easier and solve some of the problems encountered in an ever more intricate world. As the world and the activities of man have become more complex, so have the requirements of the devices that man attempts to create to reduce this complexity. The early inventors such as Charles Babbage – hindered by an unfortunate lack of equipment – had to invent the tools they needed before they could go on and build the machines they had in mind. Babbage's ultimate invention has finally led to the computer – possibly one of the most significant technological advances made by humanity.

However the computer itself is only the start, with the tool having made possible advancements seemingly restricted only to the imaginations of science-fiction writers. Using the computer, Artificial Intelligence (A.I.) has come ever closer to reality and research in A.I. has followed a number of different directions. These approaches are constantly developing, with innovations like expert systems and neural networks continually emerging. Yet there are still problems that evade solution or prove difficult for current A.I. methods. Whilst the computational speed given by the computer is helpful, combinatorially explosive problems are still very much in existence.

Search and optimisation problems – those with a large number of potential solutions – are a prime example, providing a constant test for computing approaches. One new methodology – namely Genetic Algorithms (GAs) – draws on nature, looking to the inspiration provided by theories about the origins of species.

1.2 The Genetic Algorithm

Genetic Algorithms are, put simply, a set of procedures based upon the ideas of Darwinian evolution and natural selection. Originally proposed by Professor John Holland in his book *"Adaptation in Natural and Artificial Systems"* (Holland, 1975), GAs draw upon the ideas put forward by Charles Darwin – the improving evolution of a species through survival of the fittest members in its

population by reproduction – in order to optimise complex solutions. The high-level ideas and biological principles of evolution are maintained and applied to encoded computing search problems to 'evolve' better and better solutions. GAs have spawned a number of different directions of research based upon this basic principle – survival of the fittest. These research directions are explored in more detail during the literature review carried out in chapter 2.

1.3 The Multi-Chromosome Approach

Current approaches, based upon the traditional GA concept of a single, encoded chromosome string mostly use this linear structure, whilst investigating the application of alternative representations and operators. More recently a number of authors have begun to investigate hybridisation with other techniques, hierarchically organised chromosomes and more structured GAs, but for the most part a number of Holland's original proposals (1975) have gone as yet unimplemented in a non-hybridised GA framework. In addition, consideration of the biological world identified a great deal of complexity within natural creatures, with the potential for a similar parallel to exist in GAs - demonstrated to a degree in some other fields of Evolutionary Computation research.

In studying both the move of current GAs towards more complex, multi-dimensional problems and representations, along with Holland's (1975) work, the natural world and structural research undertaken in other areas of EC, a multi-chromosome GA (Multi-GA) was devised, allowing the incorporation of several chromosomes within an individual and a consequent increase in flexibility and operational potential.

1.4 Structure of the thesis

It is the multi-chromosome structure, its development, testing and application to real world problems that are presented in this thesis, with the following structure:

1.4.1 Review

Chapter 2 provides an introduction to GA research, covering the principles behind genetic algorithms and a brief introduction to the numerous research directions which are being investigated. There then follows a more detailed discussion of the representational and structural research that has been carried out, exploring in depth recent work investigating alternative representations,

hierarchically structured and split-chromosome GAs and the initial steps towards the use of more than one chromosome to tackle a problem.

1.4.2 *Multi-chromosome approach*

Chapter 3 describes the multi-chromosome GA developed here, from the biological precedents and existing research which provided the inspiration, through to the design and implementation of the multi-GA approach both in conceptual and software development terms.

1.4.3 *Comparative testing*

Chapter 4 investigates the application of the Multi-GA in a traditional GA manner on a number of test optimisation surfaces. Details of the problems, the rationale behind the testing and the results of experiments carried out on both the Multi-GA and traditional GA are given, along with a brief summary of the findings.

1.4.4 *Dynamic parameterisation*

Studies were also carried out to explore some of the current GA research areas identified as potentially benefiting from the flexibility of the Multi-GA structure. In particular, experiments involving dynamic parameterisation and alternative chromosome representations were undertaken, with the results presented in chapter 5. This chapter also contains details of a newly developed real valued crossover method – Quotient crossover – and a brief discussion concerning the problem surfaces used, in the light of the results obtained. Chapters 4 and 5 contain a description of the studies carried out to determine the scope of performance of the basic Multi-GA. More complex tests were then undertaken and these are described in chapter 6.

1.4.5 *Applications*

Chapter 6 investigates two applications problems designed to more thoroughly explore the features of the Multi-GA not investigated by previous tests. Specifically, network minimisation and spatial site location problems were used.

Chapter 6 also includes a brief introduction to Geographical Information Systems (GIS) – an area making particular use of spatial analysis – and presents the Multi-GA results in the context of its usefulness as a new GIS tool. In addition, it outlines two new operators that provide genetic manipulation within

the multi-chromosome structure, along with a study of their behaviour and application during the site selection problem.

1.4.6 *Conclusions and future work*

Chapter 7 provides a summary of the work presented in the thesis, outlining the main conclusions of the preceding chapters. A discussion of the future potential of the Multi-GA approach is included, with details of additional work that could be carried out, both following directly from results presented in this thesis and in areas seen to hold potential, but unexplored due to the constraints of time.

1.4.7 *Appendices and references*

The thesis concludes with a list of the references consulted during the thesis and appendices presenting some additional data referenced in chapters 5 and 6.

Chapter 2: Review

2.1 The Evolutionary Cycle

Taking a novel approach to problem solving, GAs follow a simple, cyclical pattern in their operation, using terminology that relates to the sources of their biological inspiration. A GA works by randomly encoding a number of possible solutions to the problem being attempted. Each generated solution takes the role of an *individual* in a *population* of possible answers. The randomly generated solutions are then assessed for *fitness*, which is a measure of how good a possible solution is at solving the specified problem. At this point, the evolutionary parallel is applied and a new population of solutions created, by a process of selection and reproduction from the current population. A number of genetic operations are applied to generate the new population, which then goes forward to replace all, or some, of the old population in the next generation. The entire cycle is then repeated until appropriate terminating conditions are satisfied, as illustrated diagrammatically in figure 2.1 overleaf.

The effect of this evolutionary process is to produce a convergence around areas of interest on a problem surface, directing members of the population to the parts of the surface that best fulfil the criteria of the fitness function defining the problem. Observation of individuals shows a migration towards the current best areas of fitness as the population finds increasingly better solutions. This is a simple overview – naturally a great deal of research has been conducted in a number of areas and these are explored in more detail later on. Whilst the specifics of the various areas of GA research may differ slightly, the essential nature of the systems is that illustrated by the figure overleaf.

The main driving processes that contribute to the success of this cycle are the operations applied to the members of the population. These operators utilise inherent genetic building blocks known as *schemata*, and it is these schemata that facilitate the evolution of improved solutions over time. In the remainder of this section, research into the items and processes fundamental to the basic genetic cycle are examined in more detail. In addition, brief considerations of the place of the genetic metaphor in the real world and the different research directions that have been followed are undertaken.

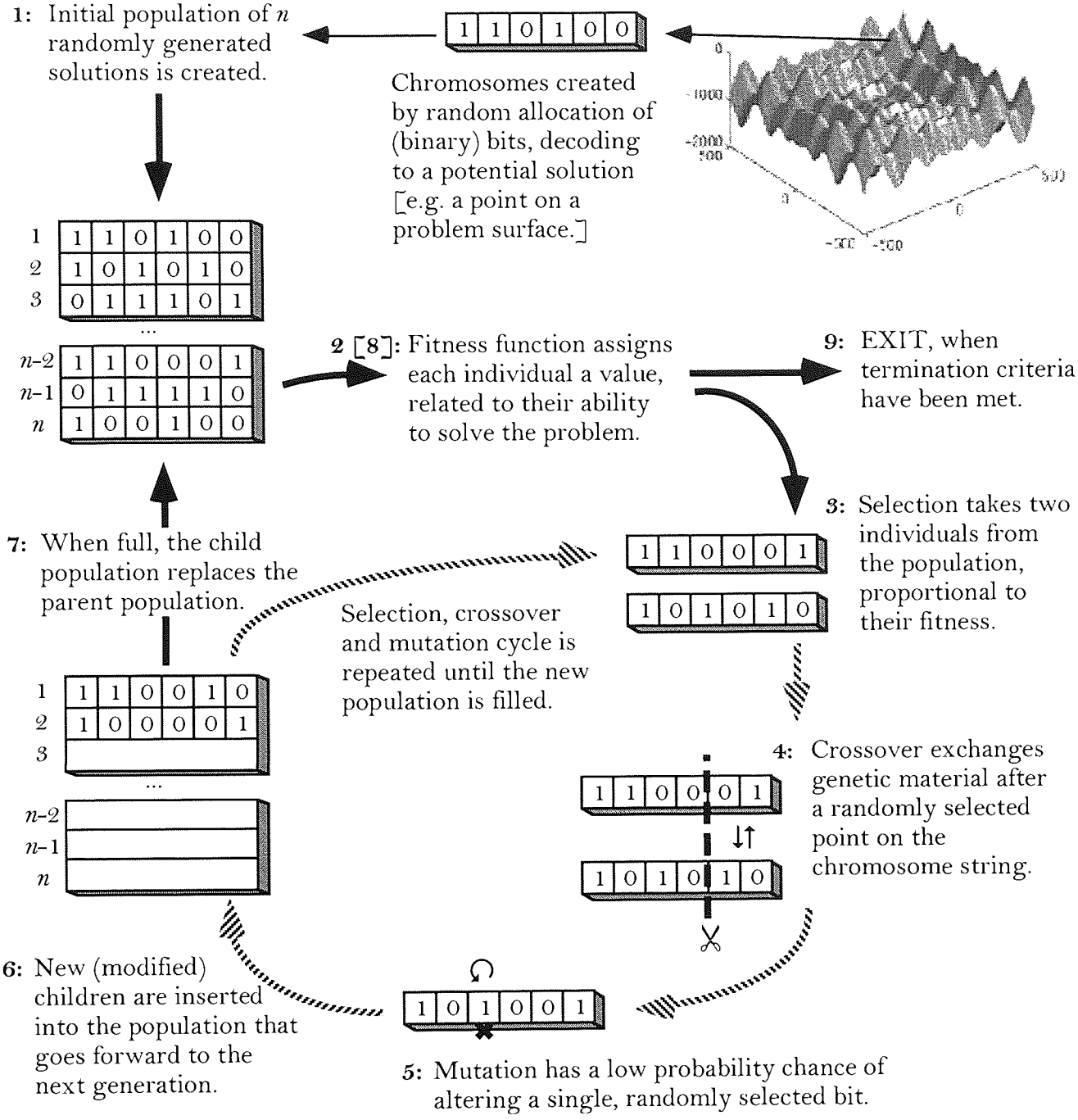


Figure 2.1: The Evolutionary Cycle

The major items covered in this section are:

- Schemata
- Selection
- Crossover
- Mutation
- Other genetic operators.
- Applications
- Parallel tracks of research.

2.1.1 Schemata

"All animals are equal, but some are more equal than others" (Orwell, 1945)

Identified by Holland (1975), schemata provide the key to the mechanism by which GAs work. An individual *chromosome* string may contain segments embodying qualities that make it 'fit' according to the fitness function being applied. These representational qualities are *schemata*. Amongst a population of chromosomes expressed in the same representation, some will contain 'good' segments that make them fitter ("more equal") than their peers.

But what actually is a schema? Goldberg (1989, p.29) describes it as a *similarity template*. A schema corresponds to a pattern of genetic information within an individual which contributes to its particular fitness value. The effect of the genetic operations is to manipulate schemata as they alter the genetic material in a chromosome. Over time, good schemata propagate through members of the population – hence the algorithm evolves 'fitter members' (that is, members containing good schemata) over time.

For example, in a traditional binary coded GA, individuals will contain allele values of 1 or 0. For a particular problem, a good combination (high fitness) might be a chromosome consisting of 1001100. Another chromosome of equally good (or better) fitness may be 1100101. In this example, the pattern producing the good fitness might be 1#0#1##, where # is a "don't care" operator – this pattern is defined as a schema. A schema represents a certain combination of genetic material, but all schemata have two important properties critically relevant to the effects of genetic operators. The *defining length* of a schema is the (inclusive) distance between the first and last bit represented by non-"don't care" symbols. The *order* of the schema is the number of fixed (non-"don't care") positions represented.

e.g. A 1#1#1 is of order 3 with a defining length of 5.
B ###01 is of order 2 with a defining length of 2.

Holland (1975) proposed a theory for the analysis of schema propagation over time called (not surprisingly) the *Schema Theorem*, and it is this theorem that has been used as a basis for much of the theoretical and mathematical analysis in the GA field.

The Schema Theorem is given by Goldberg (1989) as:

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

where:

$m(H, t)$	=	number of schema H at time t in mating pool m
f	=	average fitness of strings representing H.
F	=	average fitness of the entire population.
p_c	=	probability of crossover.
$\delta(H)$	=	defining length of H.
l	=	length of chromosome string containing H.
$o(H)$	=	order of schema H.
p_m	=	probability of mutation.

Figure 2.2: The Schema Theorem.

A detailed explanation of the theorem, illustrating the application to a simple problem and analysing the propagation of schemata is given by Goldberg (1989, pp. 30 - 35) and interested readers are referred to that text for a full discussion of schema theory.

GA operators make changes to the material on which they are working, with the consequence that schemata may be disrupted. Low order schemata of a short defining length are less likely to be disrupted and therefore more likely to propagate through to subsequent generations. The schema theorem assists in determining the likelihood of a particular schema being passed on to the next generation, following the selection of the individual containing it.

This information is of use in a number of areas of GA research, not least the design of new genetic processes or representations. This particular relationship was highlighted by De Jong (1985) in his 'ten year perspective' paper, who pointed out that the design of any new genetic process or representation must take into account the effect of genetic operations on the chromosome string. A schema of short defining length is more likely to propagate through time than a schema of longer defining length. Hence, new genetic encodings and operators should take into account, during the design process, the length of schemata defined by their representation.

As the description of the genetic operators unfolds, the importance of schemata and the effects of the evolutionary cycle upon them will become increasingly clear.

2.1.2 Selection

The underlying principle of genetic selection is that individuals are picked from the population in proportion to their fitness. Over the entire population, a greater selection pressure is applied to the fitter members, embodying the Darwinian principle of survival of the fittest. Selection of less fit members is not precluded, but is in fact an essential part of the process, responsible for maintaining diversity in the population. This simple description evidently shows the importance of selection schemes and their potential effects on the evolutionary cycle. Naturally, selection has been a major focus of research, with a number of different selection mechanisms being developed, particularly in order to address the problem of maintaining selection pressure. Popular selection schemes include:

- Fitness proportionate selection.
- Rank based selection.
- Tournament selection.

2.1.2.1 Fitness proportionate selection

Goldberg (1989, p. 11) described the method of operation of simple roulette wheel selection – that is, the proportionate selection of an individual related directly to its fitness. This is perhaps the simplest selection scheme and commonly adopted by researchers embarking into the GA field. However, analysis of the behaviour of this selection algorithm and its effect on the population soon led to studies which pointed out a number of failings. Baker (1987) performed a comprehensive study illustrating the problems of bias in such schemes. Baker's paper proposed a number of alternative schemes, with results showing the positive and negative effects of bias on the sampling performed by these schemes.

The result of this analysis (Baker, 1987) was the proposal of a new algorithm with improved, unbiased performance, namely Stochastic Universal Sampling (SUS). SUS represented a modification of the single pointer roulette wheel selecting a member n times to fill the population, by moving to an n pointered roulette wheel spun just once. Baker's (1987) study is indicative of the type of research carried out in the area of selection scheme analysis, providing valuable information to assist researchers in designing selection schemes that improve GA operation.

2.1.2.2 The scaling problem and rank based selection

Baker's (1985, 1987) studies illustrated some of the problems with selection methods and these concerns have been identified by other authors. De Jong (1985) mentioned the *"scaling problem"* of selection pressure, along with some of the attempts to address the issue. The scaling problem refers to the maintenance of an effective selection pressure throughout the generational cycle. Purely relying upon fitness proportionate selection can lead to a weakening of selection pressure as convergence occurs in the population. In an attempt to resolve these issues, a number of different approaches to selection have been taken – primarily rank based and tournament schemes.

2.1.2.3 Rank based selection

Ranking schemes have been proposed (Baker, 1985; Grefenstette & Baker, 1989; Whitley, 1989) as one method of addressing the roulette wheel problem of a weakening selection pressure over time. Relying purely on fitness criteria, one finds that the selection pressure reduces as the fitness difference between best and worst becomes smaller. This in turn leads to a failure of the GA to move forwards in any great direction (premature convergence) and a stagnation of the GA search to a level of almost random searching (Whitley, 1989). Rank based selection addresses that problem through sorting of individuals, assigning a ranking value to each individual and then performing proportionate selection on the basis of the *ranked* value rather than raw fitness. This allows the GA to maintain effective selection pressure as time progresses and provides a weapon with which to help balance the searching of the problem space against the efficient use of the genetic material discovered so far – the so-called 'exploration/exploitation' dilemma.

2.1.2.4 Tournament selection

Tournament selection is a newer approach, departing from the themes of proportionate selection by selecting n individuals randomly and then picking the fittest from within this group. A number of authors have studied tournament selection (e.g. Goldberg, 1990; Goldberg & Deb, 1991; Mahfoud, 1991; Blickle & Thiele, 1995) and examined its effects in detail. Tournament selection does have benefits over proportionate selection methods and, according to an interesting comparative study (Goldberg & Deb, 1991), a binary tournament exhibits similar performance to proportionate selection with ranking. The paper also performed complexity analysis of the various ranking schemes, providing a formal analysis of the timeliness of comparative GA selection methods.

2.1.3 Genetic operators: Crossover

Crossover is arguably the main driving force behind the search process in GAs (Goldberg, 1989). It involves the exchange of genetic material between two selected chromosomes, creating two new chromosome strings. It can be seen that any manipulation of genetic material will inherently manipulate schemata, producing new similarity templates within newly created chromosomes. A great deal of research has been carried out into different ways of performing crossover, with the aim of the process being to evolve children that describe improved solutions to the given problem.

With this in mind, a number of different crossover methods have been researched, expanding Holland's (1975) proposals for simple one-point crossover. In addition, mainly driven by application to real world problems, there has been considerable development and analysis of alternative, problem specific crossover operators, studied in more detail later on. The remainder of this section introduces the more traditional crossover operators and their variants.

2.1.3.1 One point and N-point crossover

The primary form of crossover, described in many introductory GA texts (e.g. Goldberg, 1989; Holland, 1975, p.98), has been one point crossover – the interchange of genetic material after a single point on the chromosome string between two selected individuals in order to produce new (and hopefully better) child individuals, as illustrated in figure 2.3.

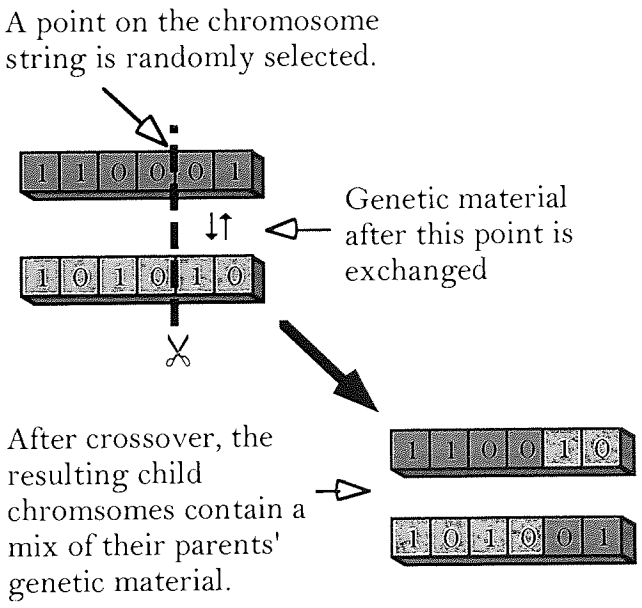


Figure 2.3: One point crossover.

However, research into crossover mechanisms has continued, ranging from simple extensions to the one point scheme through to detailed theoretical analysis of the operators and their effects. The essential principles of crossover remain unchanged, but experiments designed to improve performance, have led to extensions to the simple operators like one–point crossover. The most obvious extension is the advance to n-point crossover, illustrated in figure 2.4 (showing $n = 2$).

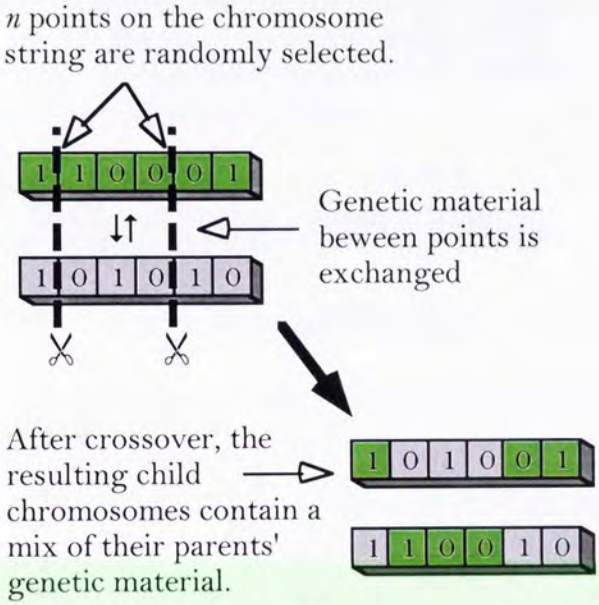


Figure 2.4: *N*-point crossover.

2.1.3.2 *Uniform crossover*

More radical extensions to the one point proposals laid out by Holland (1975) have been suggested, one of the most notable being Syswerda's (1989) Uniform crossover operator. Approaches such as n-point ($n > 1$) and uniform crossover involve a greater manipulation of the chromosome string, producing a more even-handed distribution of material exchange, as illustrated in figure 2.5.

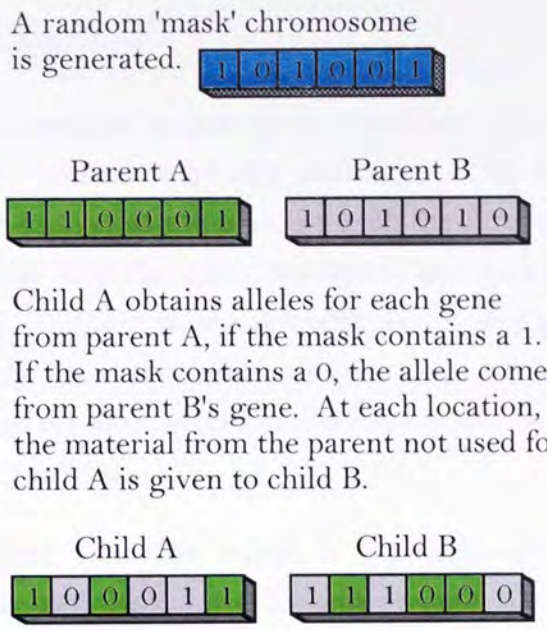


Figure 2.5: Uniform crossover.

It can clearly be seen from these illustrations that the differing forms of crossover have a distinct effect on the resulting chromosome strings. The theoretical studies carried out confirm this observation and provide future researchers with information to estimate which form of operator will be most effective for their particular problem. Syswerda (1989, p.2) highlighted the main difference between his uniform operator and traditional crossover methods, with *"1 bits uniformly distributed throughout the chromosome"* by uniform crossover.

The effect of this more even distribution is to produce an operator which has a more disruptive effect on schemata, but in a totally different way to traditional crossover methods. *N* point crossover, whether *n* has the value 1 or greater, operates on a contiguous section of the chromosome string, the length of which varies as *n* is increased. Uniform crossover approaches information exchange from a completely different direction, providing an interesting alternative operator that, empirically, produces some useful results (Syswerda, 1989). Syswerda also analysed the effect on schemata, comparing Uniform to *n* point crossover. The *"proof in the empirical pudding"* as Syswerda neatly puts it is that when in doubt, Uniform crossover is a safe bet to use. In most of the cases he analysed, uniform crossover proved to be more effective than 1 and 2 point at combining schemata, with 2 point also outperforming 1 point in most cases. Its usage as a generic crossover operator was validated in his paper, although it was not identified as a universal panacea and problems in which it exhibits worse performance were given.

2.1.4 Genetic operators: Mutation

Following the application of crossover to candidate chromosome strings, the next operator to be applied is usually mutation. This is a simple operator, functioning by the selection of a random gene on the chromosome according to a probability chance (as with the other operators) and then perturbing the value by a small amount. Without it, the introduction of new genetic material (and hence schemata) later on in the evolutionary process would prove impossible – something which would have severe consequences on the ability of the GA to produce an adequate solution.

In a binary encoded GA, the value is typically inverted, whilst other representations like real encoded GAs add or subtract a small amount. A number of mutation schemes can be seen to have developed for problem specific situations (e.g. Williams *et al.*, 1994). The paper by Williams *et al.* illustrates both sides of the mutation coin particularly well, utilising a problem specific mutation operator designed around a chromosome based on a non-binary encoding. Illustrations of the other main approach, utilising mutation as a primary search operator, can be seen in the development of Evolution Strategies (ES) (Bäck *et al*, 1991) [outlined in section 2.1.7.1] and are summarised by De Jong (1985, p.176) who points out the *"frequently tried but rarely successful strategy of increasing the mutation rate to improve GA performance."*

Mutation has been shown in the above studies to play a significant part (despite the prominence of crossover as the main driving force) and has not escaped theoretical analysis. As discussed in section 2.1.7.1, ES provide significant evidence of the power of mutation based search and papers such as Bäck *et al.* (1991) and Hansen *et al.* (1995) provide much formal analysis of the mechanisms of ES search.

Further studies in the GA community have been performed, in an attempt to understand exactly what the effects of mutation on the evolutionary process are. A good example is that of Tate & Smith (1993), which challenged the traditional role of mutation as a method of recovering lost information (Goldberg, 1989, p. 14). They related mutation to the encoding scheme used, introducing the notion of allele coverage. Their conclusions illustrated a method for studying why increasing mutation rate may be good for search, using the allele coverage metric as a tool. As we can see, the traditional view of mutation has been that of a background operator useful specifically for climbing to an optimum, or for reintroduction of genetic material lost through the domination of crossover or

by genetic drift. As a result, the majority of research has concentrated on the crossover operators discussed earlier. However, mutation has been shown to be highly dependent upon the representation scheme used by a chromosome and research into the effects of mutation is an area experiencing much analysis.

2.1.5 *Genetic operators: Additional mechanisms*

However, the genetic process does not stop with selection, crossover and mutation and there are a number of other mechanisms and operators that have developed in the course of research. Holland's original proposals (1975) included the inversion operator, designed to tighten the linkage between short order schemata in the chromosome string, assisting the genetic processes in creating schemata of short defining lengths, as illustrated in figure 2.6.

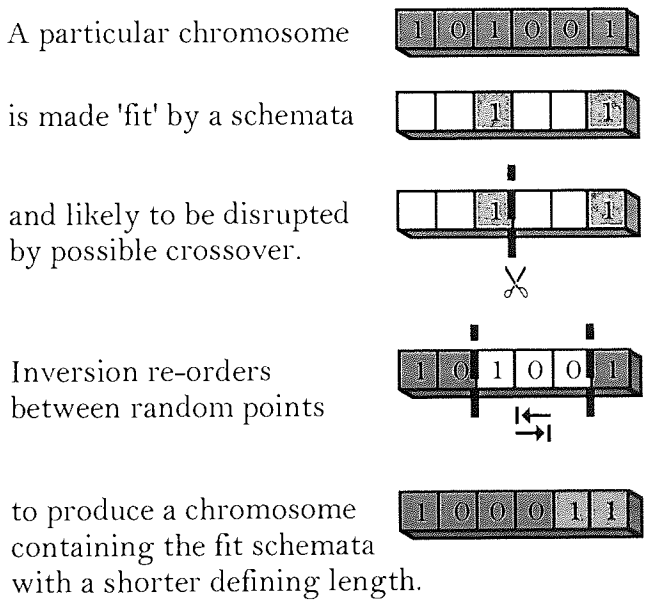


Figure 2.6: Holland's inversion operator.

Inversion effectively performs a crossover within a chromosome, producing schemata of shorter defining lengths that are equally as fit as longer order schemata by creating a tightly linked, reordered positional dependence of the component genes. The new, shorter schemata are then less likely to undergo disruption during the traditional crossover operation. A number of studies into the effects of inversion and similar style ordering operators have been carried out. It is perhaps due to these studies that such operators are not used, principally because of the time taken for them to produce beneficial effects (Goldberg, Deb & Korb, 1991a). However, other authors (Whitley, 1987) have shown that for certain encoding schemes or problems, inversion and similar operators can actually play a useful part to improve performance. This sort of result is an excellent illustration of the important information provided by

continued formal analysis, assisting future researchers in highlighting the areas in which it is most practical and useful to apply a particular operator, or combination of parameter settings. In addition to the negative results of a number of studies, inversion has other difficulties associated with it. Firstly, the traditional binary coded GA explained by Goldberg (1989) utilises a position dependent representation, whereas inversion requires position independent representation of the genes. Secondly, the demonstrated success of operators such as uniform crossover (Syswerda, 1991), which by their very operation make inversion ineffective, means that applications using this popular crossover operator will find that inversion is entirely redundant.

Arguments such as these have led to a decline in the use of inversion to an almost non-existant level, although the rise of application areas of GAs have led to much development of other reordering and problem specific genetic operators. More discussion on these other problem-specific operators, and particularly their importance in the relationship to alternative structural representations, can be found later on in section 2.2.2.1.

2.1.6 *Application of the evolutionary cycle*

The principles and mechanisms of the evolutionary cycle discussed so far highlight a number of different areas of potential for research and application of this searching mechanism. Not only have GAs resulted in a number of different research themes (discussed briefly in the next section), but they have also found application in the real world. These real world applications have had particular significance on the genetic processes themselves, and their specific relevance to the work laid out in later chapters is discussed in more detail in the section 2.2.5.

It has already been mentioned that search and optimisation problems are the central areas addressed by the GA approach and that can be seen throughout the literature. The result is that studies into the evolutionary processes and their applications have been conducted in a wide variety of areas.

Examples of the diversity of applications include formal analysis of the mechanisms of the genetic process (Radcliffe, 1991, 1992; Suzuki, 1993); automatic generation of programs – a separate strand of research, referred to as Genetic Programming (Koza, 1992); development and application of rule-based approaches known as Classifier Systems (Wilson, 1985; Sedbrook *et al.*, 1991) and a plethora of real world problems such as telephone network design (Davis *et al.*, 1993; Carse *et al*, 1995), generation of identikit style images of criminal suspects (Caldwell & Johnson, 1991) and timetable optimisation (Abramson *et*

al., 1993; Burke *et al.*, 1995). There has also been considerable research into developing new approaches and methods to improve the applicability and performance of the evolutionary metaphor and it is this line of investigation, into the representational qualities of GAs, that has been followed during the work presented in this thesis.

2.1.7 *Parallel tracks of evolutionary research*

In addition to the research undertaken specifically on the pure GA, it has already been mentioned that there are a number of other major directions that research into Evolutionary Computation (EC) has followed. These topics are very much fields in their own right, with more and more intensive research carried out in each area. Each of the fields has the same guiding principle – evolution – as its foundation and there still exists a great deal of similarity and potential for cross co-operation between each strand of research, as highlighted by the next chapter.

Some, such as Evolution Strategies (ES), have developed alongside GAs almost as an alternative school of thought; whilst areas like Genetic Programming and Classifier Systems have developed out of Holland's work (1975) into emerging fields in their own right. Whilst GAs continue to provide fruitful areas of research in search, optimisation, applications and the algorithms themselves, the same is true of each of the alternative directions taken by EC. Irrespective of the genealogy of each topic, all now provide opportunities for research and application. The three key strands of research are:

- Evolution Strategies
- Genetic Programming
- Classifier Systems.

2.1.7.1 *Evolution Strategies*

Evolution Strategies developed primarily in Germany, whilst the majority of GA research has taken place in the United States. ES rely on parameterised mutation as the primary operator for change, rather than crossover which is the driving force in GAs. Described in a comprehensive introductory paper by Bäck *et al.* (1991), ES were developed in the 1960s by Rechenberg (1973) and further researched by Schwefel (1975). Bäck's (1991) paper described the development of ES, from the initial two membered ES using a simple mutation-selection scheme, a real-valued vector individual and a descendant created from random numbers, modified by mutation.

Areas of specific interest in the ES field relate to ES parameterisation and use of real-valued representations. As discussed later in section 2.3 alternatives to the binary alphabet traditionally employed in the GA community have been taking hold and ES provide much information on the use of non-binary representations, which are central to this particular genetic strategy.

In the area of parameterisation, ES researchers have again provided studies into the effects of operator rate application, coming up with metrics such as Rechenberg's $^1/_5$ success rule (1975), defining the ratio of successful mutations to the mutation variance. Further similarities can be seen with approaches in the pure GA community to dynamic parameterisation, mirrored in the ES field by mechanisms providing for the self learning of the controlling strategy parameters through genetically inherited variables and additional strategy parameters for self-learning of the topological environment (summarised by Bäck *et al.*, 1991).

2.1.7.2 Genetic Programming

One of the newer developments in the EC field, Genetic Programming (GP) is now finding its way from the initially proposed ideas of evolving computer programs using the evolutionary paradigm (Koza, 1991; Koza, 1992) to applications of this programming strategy to solve other problems, such as automated learning of protein sequences (Handley, 1993) and image feature classification (Tackett, 1993).

Koza (1991) asked the question *"How can computers learn to solve problems without being explicitly programmed?"* and it is this central AI question that GP seeks to address, through application of the evolutionary concept. GP differs from the main thrust of GA work in the representation of the chromosome string used. Rather than working with a chromosome string consisting of a single, decodable gene representation, the GP chromosome consists of sub-trees of a program - typically in an order based language such as LISP. The genetic process randomly generates and manipulates entire sub-trees to produce a program, the success of which dictates its fitness. GP evolution is guided by two basic genetic operators – fitness proportionate selection and recombination (crossover). By working with entire sub-trees of LISP generated programs, the genetic operations always result in syntactically valid, if potentially less fit, programs after selection and recombination (Koza, 1991). Selection operates in the same way as the traditional GA approach,whilst mutation is less important, replacing an entire selected sub-tree with a new, randomly generated sub-tree.

This description of genetic programming highlights a number of similarities to the pure GA research field and, in particular, identifies several approaches to the representational structure and style of GP which have great similarity to approaches currently being examined in the pure GA field. As with the parameterisation studies carried out in ES, the structural approach taken by many of the studies in GP carries important ideas and results for those working in the pure GA field of representation – an area discussed in more detail in section 2.4.

2.1.7.3 *Classifier Systems*

Classifier Systems (CS) represent EC's approach to tackling the central A.I. question of machine learning. Maintaining the now familiar genetic operators of selection, crossover and mutation, classifier systems work with a population of *production rules.* Utilising a rule based problem solving approach, the classifier system attempts to evolve better solutions by applying a genetic algorithm to evolve new rules, combined with a credit assignment method of rewarding beneficial rule systems.

As with the other fields of GA based research, CS development has followed a number of paths, described in some detail by Wilson & Goldberg (1989) in their critical review of the field. The main topic of debate in the CS field has been concerned with the method of credit assignment, where two schools of thought have arisen. The Michigan approach (Holland & Reitman, 1978) proposed use of credit assignment at infrequent intervals, maintaining an activation history and paying classifiers activated since the last payoff, as embodied by the *bucket brigade.* The alternative Pitt approach (Smith, 1980) involved genetic evaluation of entire classifier rule sets, thereby side-stepping many of the issues concerned with credit assignment efficiency (Wilson & Goldberg, 1989).

Credit assignment is not the only area of research in the CS field and other investigations have been conducted into the mechanisms of CS operation. This has resulted in new algorithms (Shu & Schaeffer, 1991), hybrid systems (Parodi & Bonelli, 1993) and analytical studies to increase the understanding of the problems faced by genetic rule based systems (Miller & Forrest, 1989). In addition, an ever increasing number of applications to real world problems exist, with CS have been applied in a range of diverse areas from patient classification in triage (Sedbrook *et al.*, 1991), computer memory processors (Kitano *et al.*, 1991) through to routing design in telecommunications networks (Carse *et al.*, 1995).

As with Genetic Programming, there are a number of issues in classifier systems of particular relevance to recent representational research in the pure GA field and much can be learned by studying the mechanisms used by Classifier Systems. Whilst they may initially seem to attack different problem areas, as with GP the representational strategies adopted by classifier systems (e.g. Smith, 1980; Smith 1983; Greene & Smith, 1987) bear a striking similarity to parallel ideas currently being pursued in the field of pure GA research. This close correlation not only reinforces the emergence of Classifier Systems from the same roots as pure GA research, but provides a number of lessons and ideas for researchers attempting to apply parallel ideas in the field of pure GA research. This relationship, particularly in terms of structural and representational similarities that are most relevant here, is examined later on in section 2.4.5.3.

2.2 Pure GA Research

The primary area of research activity to date has been in studying the GA process itself and in determining the scope of application for this novel search methodology. The focus of this research has included not only the technical processes, but more holistic and strategic views of the research effort. Notable papers by Goldberg (1989a) and De Jong (1985) attempt to provide some guidance in the approach that researchers should undertake in their work.

Goldberg's (1989a) paper provides an insight, and a strategic approach, to the design philosophies that one should seek to undertake in the development of GAs. It illustrates the level to which the GA community have analysed both their algorithms and methods of research and provides an analysis of the potential pitfalls in the GA research process. In the same way that GAs and their evolutionary paradigm depart from traditional problem solving methods, papers such as Goldberg's indicate that a similar change in researchers' attitudes may also be required in this new and emerging field.

De Jong's (1985) 'ten year perspective' paper relates the ideas subsequently embodied by Goldberg's guidance to specific issues, providing a good indication of the areas that, in his view at the time, GA research needed to address. His analysis of the field is an excellent starting point from which to examine the area of pure GA research.

De Jong (1985) identified the principal areas into which analysis and research can be divided, namely:

- Theoretical study.
- Operators.
- Population behaviour.
- Performance (including parameterisation).
- Applications.
- Representation.

Areas requiring study identified in 1985 have not changed, but advanced to open up new topics within those same areas. Each of these areas contain issues of particular relevance to the structural and representational work presented later, particularly in the area of representation which is explored in some detail in the remaining sections of this chapter.

2.2.1 *Theoretical study and alternatives to the Schema Theorem*

Titles such as this immediately lead one to think of mathematical dissection and study, for which the GA field is no exception. In his book, Holland (1975) gave the beginnings of a formal, mathematical explanation of the mechanisms by which the GA process operates, outlined in the earlier discussion of schemata and schema theory. However, the topic of "theoretical analysis" is more wide ranging than just mathematical study and includes the effects of the genetic operators, the dynamics of the GA population over time and performance related issues. These themes were identified by De Jong (1985) in his ten year perspective.

Although very useful, the schema theorem is not the last word in theoretical analysis and other authors (e.g Radcliffe, 1991, 1992) have identified additional requirements for theoretical modelling of GA behaviour not adequately addressed by the schema theorem. Indeed, a number of areas in GA research – in particular the development of new genetic operators and processes discussed in more detail later on – are currently in need of additional theoretical analysis, so alternative approaches merit some consideration.

Radcliffe has taken analysis of schema theory a step further, introducing the concept of more general objects and manipulative operators known as *forma* (Radcliffe, 1991), along with introduction of specific genes related to these forma (Radcliffe, 1992). Other authors have taken a different approach to modelling the behaviour of GAs, from specific studies of the original schema concept

(Hulin, 1991) to different modelling approaches such as finite markov chains (Goldberg & Segrest, 1987). What is clear is that a degree of formal analysis to understand GA behaviour has been undertaken by the GA community, in an attempt to better explain how the genetic process actually achieves its results. This formal analysis continues, with each new advance or technique opening up new ground for formal explanation and relation back to the concept of schemata or alternative concepts such as forma. Interesting examples of such analysis can be seen in the development of specific problems that highlight schema propagation and identify new difficulties such as parasitic behaviour (Mitchell, Holland & Forrest, 1994) and deception (Goldberg *et al.*, 1992).

By making oneself aware of alternative methods in analysing of GA behaviour, researchers in non-theoretical fields of GA research may find useful new tools to help provide a formal basis and explanation for their work.

2.2.2 *Genetic operators*

Genetic operators are the key to the effectiveness of the evolutionary process, as mentioned earlier. Section 2.1.3 presented a discussion concerned with the studies involving direct analysis and development of genetic operators themselves (e.g. Syswerda, 1991). In addition, work has also been carried out to investigate the effects of operators on the population and schemata, as well as suggesting appropriate applications for the wide variety of general and problem specific operators in existence.

Based around the arguments raised by theoretical analysis, the main approach to the development of genetic operators has been a desire to understand and improve GA performance. Holland (1975) initially identified the genetic operations of crossover, mutation and extended re-ordering operators such as inversion, leading to much subsequent research being carried out as discussed earlier in this chapter. Of particular interest in the context of structural and representational research has been the rise of problem specific crossover mechanisms and the close relationship between the work in these two areas.

2.2.2.1 *Problem specific crossover*

When analysing real world problems, it is not uncommon to find that a binary encoded GA representation may be inappropriate. This issue has been identified by several prominent authors such as Davis (1991), who advocates the use of hybridised representations where appropriate. Alternative representations (discussed later) have also led to the development of specialised crossover operators in order to maintain legality of representation. The use of alternative

representations was viewed by De Jong (1985) as an *"alternative to finding a representation which fits the standard versions of crossover and mutation."* That is to say that problem specific operators may be required if the generic crossover and mutation operators are incapable of maintaining legal representations. Recognised as such, the development of problem specific operators has proven to be an area of intensive research, extending the scope of GA applications markedly. Hybrid operators designed for more generic use, such as Sirag and Weisser's (1987) simulated annealing based unified thermodynamic operator, have been proposed. Other authors such as Schaffer and Morishima (1987) introduced modifications incorporating a notion of dynamic control, encoding the crossover points as part of the chromosome string with their punctuated crossover mechanism. This line of research has continued, with a recent paper by Levenick (1995) proposing an extension to Schaffer and Morishima's proposals in the form of inserted 'metabits' to dictate the legality of a crossover point at each bit.

These types of approach highlight particularly well the interaction across areas of research in EC and the high correlation between the sub-processes that make up the genetic cycle. For example, it is widely recognised that genetic operators are particularly sensitive to control parameters (Schaffer *et al.*, 1989a) – a feature exploited in the insertion of crossover points and the use of metabits.

The development of such a variety of operators requires, as noted by De Jong (1985), that an understanding be gained of how they effect the genetic process. Starkweather *et al.* (1991) contributed to this understanding with their comparison of six sequencing operators. Their study provides a useful analysis of the method of operation of non-traditional styles of operator, giving an insight into the application and design of such operators in a problem oriented context. Their conclusions also backed those of a study by Schaffer and Eshelman (1991), amongst others, in observing that some operators perform better in conjunction with mutation, although analysis of the exact rates was beyond the scope of the Starkweather paper. The authors also highlighted the important theoretical point that performance of problem specific operators is directly related to the nature of the problem (Starkweather *et al.*, 1991, p. 73).

As mentioned earlier, retaining legality of representation has been one of the main reasons for the development of alternative operators. Examples vary between ordering operators for Travelling Salesman style problems (e.g. Whitley *et al.*, 1989), wrap around block operators for source apportionment problems (Cartwright & Harris, 1993), GP style structural manipulation operators (Williams *et al.*, 1994) and several alternative methods applied to the

job shop scheduling problem (e.g. Bagchi *et al.*, 1991). These applications clearly illustrate that, whilst the essence of crossover remains the same, the method by which genetic exchange takes place exhibits a wide variety according to the area in which the GA is being applied. It can also be seen that the specific area or method of application, or representation, may have a direct effect on the alternative operator being applied and vice versa. This issue is also closely related to the discussion of the real-world application of GAs, covered in section 2.2.5.

2.2.2.2 *Effects of operator application*

The genetic operators have a significant effect on the behaviour of the population in producing improved fitness over time. It is evidently an important part of the research effort to understand why and how this occurs – both in relation to the classical operators and in predicting the behaviour of newly developed operators. The schema theorem acts as a starting point and includes references to the effects of crossover and mutation in its formulation. The need for operator research was recognised by De Jong (1985, p.175) who observed that it is *"important to verify that they [new operators] aren't overly disruptive of the process of distribution of trials ... and that they encourage the formation of building blocks."* This statement reconfirms the importance of the schema theorem and schemata processing in the GA process and sets an important requirement for the understanding of operator behaviour.

Developments in GA research over the ten years since De Jong's (1985) analysis have followed this guidance and a number of authors (e.g. Schaffer & Eshelman, 1991; Starkweather *et al.*, 1991; Tate & Smith, 1993) have produced studies of the existing operators and their effects on GA behaviour. Comparative studies of operator performance have been carried out, addressing not only the classical GA operators of simple crossover and mutation, but variations on crossover along with new or less frequently used operators such as inversion (e.g. Whitley, 1987).

Crossover performance was subjected to a detailed analysis by Schaffer and Eshelman (1991) who mixed two subpopulations with differing degrees of crossover. Study of the resulting rate of take-over provided an interesting analysis of how serious the effects of schema disruption by crossover (recognised in the schema theorem as a factor affecting propagation) actually could be. Their conclusions are interesting, confirming that crossover introduction can be a mixed blessing. They went some way to identifying the circumstances in which crossover and mutation interact to produce positive results highlighting,

with illustrative reasons, the circumstances in which the more disruptive Uniform crossover operator outperforms 2 point crossover and vice-versa. They also compared and contrasted their results to those of Fogel and Atmar (1990), whose results analysed the power of selection and mutation for search – seemingly in contrast to Shaffer and Eshelman's. This discrepancy was explained by the latter authors, who highlighted the differences in representation and inversion style operators used in the two studies. This explanation in itself raises interesting issues – namely the effect of alternative representations on GA performance (covered later). Other effects of a variety of crossover operators can also be identified, as illustrated by Syswerda's (1989) observation that uniform crossover's method of operation negates any effect of inversion. The type of study undertaken by these authors is essential to a thorough understanding of how the introduction of an operator affects the performance of a GA. Crossover has traditionally been seen as the main driving force, with mutation *"appropriately considered as a secondary mechanism"* (Goldberg, 1989, p.14). However, the nature of the interaction between operators, and their effects on the population, makes it critical that a thorough analysis of their behaviour is carried out.

2.2.3 *Population behaviour*

The effects of a particular rate, algorithm or operator applied by a GA can be widespread. Just as the understanding of a particular operator is important, the wider effects of the GA process on the entire population must be understood. Population behaviour covers a large part of the evolutionary cycle, encompassing selection algorithms, convergence issues, population manipulating algorithms, operators and parameter rate control.

There have been some novel and interesting attempts by GA researchers to address the problems of premature convergence and population behaviour, those of particular interest involving structural and representational manipulation. In addition, a growing body of researchers are now investigating co-evolution and the use of subpopulation schemes, carrying important consequences both for GA development and co-operation between the different areas of EC research.

2.2.3.1 *Premature convergence*

Premature Convergence is an issue that has governed a significant part of population behaviour research. Related to the selection problem, GA search is hampered by the dominance and overtaking of single, particularly fit, individuals. The rapid spread of these individuals, which can occur in some

situations, then leads the GA to convergence at a solution that is not optimal. Having had a large proportion of the gene pool over-run by these super performers, mutation alone can find it impossible to break out of the rut into which the population has been pushed. Ranking and tournament selection go some way to addressing this issue, by reducing the reliance of genetic selection upon particularly high fitness values, producing a more even selection distribution. However, this alone is not enough and there continues to be a great deal of interest in alternative methods of solving the 'exploration/exploitation dilemma'.

A variety of different approaches have been suggested such as incest prevention (Eshelman & Schaffer, 1991; Craighurst & Martin, 1995), design of new parameters based on approaches like simulated annealing (Sirag & Weisser, 1987) and strategies that attempt to interpret and adapt to the problem surface (Shaefer, 1987). Papers such as Eshelman & Schaffer's tackled the question by controlling the population itself, whilst retaining the unmodified basic principles of the GA. Their paper presented a method of diversity maintenance through population analysis and similarity restriction by 'incest prevention'. The issue of incest prevention was further explored in a paper by Craighurst & Martin (1995), who approached the issue through maintenance of a family tree, applying varying generational levels of incest prevention. The results of their study showed some promising effects of this intriguing approach to population control. These approaches to incest prevention in particular highlight an interesting representational use of the GA process, incorporating attempts to retain some level of information from one generation to the next. An immediate similarity to work in the field of Classifier Systems and the widely debated issue of credit assignment (e.g. Smith & Goldberg, 1990; Grefenstette, 1988; Westerdale, 1989) can be implied, with incest prevention researchers mirroring the principles of bucket brigade credit assignment, through control of the genetic process based upon the history of individual's performances over time.

Other approaches with aims similar to those of alternative selection schemes include crowding (De Jong, 1975), "genetic censorship" (Mauldin, 1984) and preselection (Mahfoud, 1992). The issue of premature convergence is one that continues to occupy a significant amount of research time in attempting to find the optimum parameter set or algorithm to effectively balance the dilemma.

2.2.3.2 *Subpopulations and niching*

Subpopulation schemes are an interesting offshoot of the population dynamics area and yet another facet of behaviour for which GAs hold much potential. They have been applied in a number of areas, including real world problems, but also as a method of dealing with premature convergence. GAs are quick to find the general area of a solution, but lack what De Jong (1985) referred to as the *"killer instinct"* in seeking out a single optimum solution. This observation, which has also led to research in hybridisation, can be interpreted to show that GAs will quickly identify a single area at the expense of other areas.

Subpopulation schemes attempt to preserve genetic diversity by performing a more local search, restricting the ability of members to be dominated by other individuals from a different subpopulation (Spears, 1994). In addition, they allow effective identification of multiple areas through sharing out the available population amongst multiple peaks of interest (Goldberg, 1989; Deb & Goldberg, 1989).

In addition to preservation of genetic diversity, subpopulation schemes have been applied in a number of other areas of research where they have shown significant potential for benefit, particularly in relationship to parallelisation of GAs (outlined in section 2.3.4.5 on performance issues). Just as parallelisation has been utilised to take advantage of the implicit parallelism of GAs, there also exists enormous potential for parallel evolution of subpopulations or niches. The issue of niching and the use of subpopulations has been identified and successfully applied by several authors (e.g. Harvey, 1992a, 1992b, 1992c; Deb & Goldberg, 1989) and presents great potential for application to parallelisation.

In addition, the use of structural modifications and representational alternatives to the traditional GA, such as Harvey's SAGA system, Grefenstette's SAMUEL system (Grefenstette *et al.*, 1990) and the structurally related field of Classifier Systems all provide opportunities for the expansion of subpopulation schemes. Indeed, a recent paper by Potter, De Jong and Grefenstette (1995) explores the subpopulationary application of the SAMUEL system. The area of co-evolutionary application holds significant potential for future research, both across the different fields of EC and especially given the structural advances being currently undertaken in the pure GA field (examined in section 2.4). As discussed in Chapter 7, there is a direct potential for the work presented here to be applied to subpopulations, with promising indications for the future.

What is clear throughout these approaches to population dynamics, and the variety of selection schemes and operator mechanisms discussed elsewhere in this thesis, is the huge potential for improvement of GAs by adaptation. A wide variety of different approaches, all with merits and opening new questions, are being researched by authors in an attempt to address the core issue of exploration vs. exploitation.

2.2.4 *Performance issues*

Having undertaken theoretical analysis of the population and operators, there has naturally been research into the performance of the GA itself – that is, how well it actually solves the task in hand. Very much related to the studies of the effects of operators and the exploration/exploitation dilemma, research into GA performance has concentrated to a large extent upon suitable problem design. In order to complete a thorough analysis, one must naturally have a benchmark upon which to base a study.

2.2.4.1 *Test Suite studies*

De Jong (1975, 1985) has again been a central figure in this area of research, designing a widely used 5 problem test suite. He also identified the need for careful thought and research into fitness functions looking at a combination of other factors, rather than single scalar values (De Jong, 1985, p. 175).

Research into suitable testing functions has continued, with alternative functions designed to highlight specific performance problems, having been developed. A recent paper by Whitley *et al.* (1995) presented a comprehensive study of the failures of existing test problems, suggesting that GA researchers should avoid problems that can easily be solved by other methods. Whitley's paper also identified a number of specific failings in commonly used test functions and suggested combinations of problems that provide a more rigorous test suite. The results of Whitley *et al.* (1995) confirm the study by Davis (1991a) investigating test functions F1 - F7 with traditional and steady state GAs, showing that the existing test suite is not unbiased to alternative GA representations, is significantly parameter and representation coding dependent and may be more amenable to solution by non-GA methods.

2.2.4.2 *Deception and other techniques*

Goldberg *et al.* have provided a great deal of work in designing so-called 'deceptive' functions (e.g. Deb & Goldberg, 1992; Kargupta *et al.*, 1992; Goldberg *et al.*, 1992), which are difficult problems specifically designed to

deceive the GA search process. The analysis of these problems, and methods used to overcome deception, has led to a number of notable structural changes to the basic GA (discussed later in more detail).

The identification of such problems has opened the field to further work, with theoretical analysis being extended through specific methods such as subpopulations and other premature convergence avoidance techniques (discussed in the previous section). There have also been attempts by authors (e.g. Forrest *et al.*, 1993; Mitchell *et al.*, 1994) to take a higher level view and analyse what exactly makes problems hard for GAs, along with developments like the 'Royal Road' problem identifying new behaviour such as 'parasitic schemata'.

There is considerable analysis being undertaken to understand not only the 'how' of GA performance, but also the more strategic 'why?' – exactly the point raised by De Jong (1985, p. 170), who stated *"GAs have properties of their own ... the key to a successful application is to understand and exploit these properties."* The theoretical analysis of GA mechanisms, behaviour and performance gives us the opportunity to do exactly that.

2.2.4.3 *Effects of parameter selection*

Any complex process involving a number of different variables naturally requires parameterisation and the GA is no exception. This chapter has so far illustrated a variety of features exhibited by GAs, each of which has associated parameters, the different values of which will have differing effects on the entire process including the parameters and features themselves. As with the other facets of GAs, parameterisation has been a major focus of the research effort, in an attempt to find a robust and general approach to optimising parameterisation settings.

The scope of features that require parameterisation in a GA is large. Firstly, there are issues related to population dynamics, such as the size of the population itself, the frequency of communication between subpopulations or niches, the rate of replacement into the new population etc. Issues of elitism – whether to carry across the best member(s) – arise, as does the choice of selection strategy. For each new selection algorithm comes a potential new set of parameters, as is the case with the genetic operators. Operators have to be applied, but with what frequency? Having decided to apply the operator, one then faces questions as to the type of operator – posing the further question of

treating differing types of operators themselves as a form of parameter, given the different effects they have on the population and its rate of convergence.

This graphically illustrates the immense scope of the number of decisions and their potential effects. This is without adding the complexity that may be required by applying a GA to, say, a problem with a changing fitness function modelling a dynamic system! The difficulty of this parameterisation task has been a focus of research right from the outset of the GA field.

Perhaps the most renowned study is De Jong's (1975) analysis of parameterisation, producing a set of benchmarking functions and parameter settings he found to provide good general performance. Indeed, despite the initial impression created by the large number of parameters, De Jong puts forward the argument that *"within reasonable ranges, the values of such parameters are not all that critical"* (1985, p.176). This analysis lends support to the acceptance in the GA community of De Jong's (1975) work and the results obtained in that study, which showed that a population size of 50 - 100, crossover rate of 0.6 and mutation rate of 0.001 gave good results over a wide variety of problem domains.

Since De Jong's (1975) thesis, studies have been performed on a wider range of issues than simply varying parameter rates for the crossover operators. For example, analysing issues related to the chromosome encoding schemes and type of operators applied have also been studied. One such paper is that by Schaffer *et al.* (1989a) who, in addition to contrasting De Jong's (1975) results with those of authors like Grefenstette (1986), also tackle issues like the usage of gray coding as opposed to simple binary coding. Schaffer's (1989a) study performed a detailed analysis of crossover and mutation rates for a variety of population sizes, suggesting some alternative parameter settings to those put forward by De Jong and Grefenstette. This area of research is still active and widespread, as illustrated by authors such as Reeves (1993) studying the issue of small population size and Whitley *et al.* (1995) confirming the benefits of gray coded binary over simple binary encoding.

2.2.4.4 *Dynamic parameterisation*

Dynamic parameterisation attempts to find the optimal parameter settings for a GA during the GA process itself. One of the most referenced papers in this field, mentioned already, is Grefenstette (1986). He provided a novel approach to finding the optimum set of GA parameters from the range of possible solutions, by assigning this optimisation task to a GA! This 'meta-GA' approach

was used to evolve a set of parameters utilised by a lower level GA applied to a particular problem and gave promising results, with rates differing from De Jong's (1975) suggested parameters. Grefenstette's recommended values were population size 30, crossover rate of 0.95 and mutation rate of 0.01 and he went on to identify a number of reasons explaining these results.

This style of parameterisation has continued, with a number of approaches being taken in attempts to provide the most flexible GA possible, capable of adapting to dynamically changing problem surfaces. The idea has obvious merits – if the problem surface the GA is applied to alters, it is logical that the GA should evolve new parameters most suited to its new environment. An algorithm that professes to follow an evolutionary paradigm should not exclude itself from self-adaptation! Examples of recent research into self-adaptation include Lee & Takagi's (1993) system utilising fuzzy logic techniques to control parameterisation; examination of the problem space before deciding upon parameterisation (Cartwright & Mott, 1991) and Davis' (1989) dynamic adaptation of operator probabilities.

The idea of dynamic parameterisation obviously shows promise – if one accepts the concept of evolutionary algorithms being able to adapt to form acceptable solutions, the concept of dynamic adaptation of the multitude of parameters by which those algorithms operate then follows logically. This theme is discussed more directly in chapter 5.

2.2.4.5 *Parallelisation*

An alternative understanding of 'performance' is perhaps defined in terms of speed, or machine efficiency. One of the criticisms levelled at GAs by researchers in other fields, such as neural networks, is that they take too long to find solutions. The CPU intensive nature of GAs is well known and attempts to improve the speed performance of GAs have led to research in areas concerning parallelisation.

Another, more important, factor in the drive to research parallelisation is that GAs have been shown (Holland, 1975) to utilise *implicit parallelism* in their manipulation of schemata. The notion of implicit parallelism – the processing of something like n^3 schemata for n structures (Goldberg, 1989, p.40) – has provided a natural channel for the development of GAs on parallel hardware, and has been investigated by many researchers (e.g. Spiessens & Manderick, 1991; Chen *et al.*, 1993; Kitano *et al.*, 1991). Other authors (e.g. Mühlenbein *et al.*, 1991) have proposed structures that attempt to incorporate parallelism into

their operation, to take advantage of diversification onto this form of hardware system. This general move to take advantage of hardware to exploit implicit parallelism has been specifically identified by prominent authors such as Davis (1990), who forecast that *"Genetic Algorithms will be widely used on parallel computers, since they are intrinsically parallel algorithms."*

Parallelisation is not limited to genetic operators alone, with studies having also been suggested for other areas of the GA field. Population dynamics are a prime example, with the subpopulation and niching concepts also being obvious candidates, investigated by several studies (e.g. Cohoon *et al.*, 1991; Davidor, 1991). Tournament selection was identified by Goldberg & Deb (1991, p.81) as *"particularly easy to implement in parallel."* Some theoretical analysis of the mechanisms by which parallel GAs operate has also been carried out (Pettey & Leuze, 1989).

It is this two track development – modification of the GA structures to exploit implicit parallelism and the application to specific parallel hardware – that has provided the main path for theoretical analysis and practical implementation of the essentially parallel nature of GAs.

2.2.5 *Applications*

There has been much discussion in this chapter so far about the widespread application of GAs and the role this has played in the theoretical development of new types of GA. It is therefore pertinent that some of these applications be mentioned, to highlight the variety of uses that GAs have been successfully applied to both inside and outside the academic community.

2.2.5.1 *Common optimisation test problems*

Within the academic environment, a great deal of effort has been put into applying GAs effectively to the problem of function optimisation. Indeed, GAs are typically described as search and optimisation procedures based on Darwinian Evolution. Widely used problems such as Travelling Salesman (Homaifar *et al*, 1993; Whitley *et al.*, 1989), Job shop scheduling (Nakano & Yamada, 1991) and Prisoner's Dilemma (Fujiko & Dickinson, 1987) have all been subjected to GA analysis. The success of the GA at solving these kind of problems has played a two-fold role in GA development. It has undoubtedly led to research into the theoretical and structural advancement of the GA, with a number of papers studying new GA approaches to function optimisation (Mühlenbein *et al*, 1991).

2.2.5.2 *Real world problems*

Secondly, real-world applications with an optimisation requirement have been targeted and in recent years, GAs shown to have widespread beneficial applicability in many areas. Right from the early stages of GA research following Holland's (1975) work, prominent authors began to apply GAs in the real world – notably Goldberg's (1983) application to gas pipeline operation. Areas of application spread from the obvious optimisation tasks such as timetabling (Abramson, 1993; Burke *et al.*, 1995), routing and scheduling (Gabbert *et al.*, 1991; Thangiah, 1995) through to the less obvious tasks such as criminal identikit development (Caldwell & Johnson, 1991) and combat target detection (Bala & Wechsler, 1993).

2.2.5.3 *The role of applications in GA development*

GA applications have played their part in theoretical development, and through application to problems such as air quality and the source apportionment problem (Cartwright & Harris, 1993) alternative operators required to maintain legality of representation or improved search potential have arisen. More definite departures from the traditional GA, which have led to significant structural changes, can also be seen in applications to telephone network optimisation (Davis *et al.*, 1993) and brewery delivery scheduling (Juliff, 1993), covered in more detail in section 2.4.

Examples such as these serve to highlight the symbiosis between applications development and research into better GA operation and techniques, brought about partially as a consequence of their application. The widespread areas of application also serve to show that GAs are a serious methodology, despite their relative youth in comparison to other more established artificial intelligence techniques. This is given further weight by the seriousness with which GAs have been taken by major companies, in a number of cases investing millions of dollars in GA based solutions (Davis, 1993a).

2.3 Representation Issues

Given the method by which GAs manipulate an encoded chromosome string to search problem spaces, finding an acceptable representation to map the problem space into chromosome string form is critical. Account has to be taken of both the accurate description of the problem space and the ability of the genetic operators to manipulate the schemata in an efficient manner. These issues neatly encapsulate the continuing debate in the GA community as to the most appropriate form of alphabetic representation.

2.3.1 Chromosome encoding

The importance of chromosome encoding cannot be underestimated, with prominent authors such as De Jong (1985, p. 170) identifying the problem of selecting an appropriate mapping as ranging from *"a trivial activity to a highly creative one"*. Binary encoded chromosome strings have been suggested as preferable (Goldberg, 1989; Holland, 1975) because of the low cardinality of the alphabet. This relates closely to the schema theorem and the ability of the GA to manipulate building blocks, which some authors believe to be an easier thing to undertake in a low cardinality alphabet. Other authors however (Davis, 1991) propose that non-binary representations, or hybrid representations, are in fact essential for an accurate representation of many problems. They propose that GA research should not presume binary encoding to be best in all cases, despite the schema based theoretical arguments. The debate over alphabetic representation is defined by these two summary points of view, with the discussion of cardinality and 'minimal alphabet' properties of a given representation making a highly relevant contribution.

2.3.2 Minimal alphabets

De Jong (1985, p. 171) pointed out that *"it is easy to demonstrate a dramatic improvement in the behaviour of GAs in switching from a short length, high cardinality representation ... to a longer, lower cardinality representation."* As mentioned by Antonisse (1989) in his proposals for a novel schema encoding, the number of schemata processed for an alphabet of cardinality v and string length k is about $(v + 1)^k$. Schema processing, it is argued, is simpler and more effective through use of a simpler cardinality alphabet due to the maximisation of hyperplanes in the GA search space (Whitley, 1993). Other representational considerations, such as the implementation of encoding schemes like gray coding in order to avoid hamming cliffs in the search space (Grefenstette, 1986; Whitley *et al.*, 1995), add to the attraction of binary encoding through its simplicity of implementation. The need for specialised genetic operators is eliminated, with the GA implementations being true to Holland's (1975) initial proposals and schema analysis work.

However, there is significant debate as to the merits and usage of non-binary encoded GAs. The argument of greater complexity in cardinality of alphabet is interesting, but studies such as those by Antonisse (1989) and Goldberg (1991) significantly contribute to the theoretical understanding of schemata and minimal alphabet behaviour in non binary encoding schemes. Antonisse undertook an analysis of Holland's (1975) schema theory proposals, suggesting

that the number of possible strings indicated by use of the # (don't care) symbol is in fact greater than originally indicated. He proposed consideration of remaining *subsets* of strings, thereby increasing the power of the # operation within the schema theory and, as suggested by the title of his paper, overturning the binary encoding constraint. As is commonly found, all is not as it seems and the paper analysed a number of other necessary changes, but nevertheless provides an interesting introduction to a discussion of interpretation of the schema notion.

Goldberg (1991) took the accepted concept of minimal alphabets into the domain of real-coded GAs, providing theoretical analysis for non-binary coded GAs and the proposal of a *virtual alphabet*. The paper illustrates the virtual alphabet, identifying the searching carried out by a higher cardinality alphabet in terms similar to those of binary minimal alphabets. The study provided by Goldberg in analysing virtual alphabets provides a great deal of insight into the method by which non-binary representations actually perform searches. It goes some way to identifying the reasons for the success of real encoded GAs and also (as expected once an understanding has been gained) providing examples of situations at which they are likely to fail, presuming that the GA reflects the simple nature assumed by Goldberg (1991).

2.3.3 *Binary vs. Real encoding*

As illustrated by the above discussion, the conclusion adopted by GA researchers so far seems to be "whichever representation is most appropriate for the problem in hand" and this has to be seen as the most sensible solution.

Despite this, a number of studies have attempted to resolve the issue of real or binary encoding. For example, papers such as Goldberg's (1991) analysis provide other authors with a good foundation on which to explore the issues raised in the real/binary debate in more detail. Janikow and Michalewicz (1991) have done just this, taking suggestions proposed by Goldberg (1991) and proposing modifications to genetic operators with floating point representations to give enhanced performance. Once again, it can be seen how the identification of psychopathic blocking problems only leads to further research aimed at overcoming these problems with a consequent improvement in GA performance.

In addition to such theoretical analysis, the applications of non-binary GAs are widespread, advocated by such prominent authors as Davis (1991). His book contains a number of different applications of GAs to problem environments, specifically requiring the use of alternative representations. Indeed, Davis

concentrates on the issue, stating that *"genetic algorithms, although robust, are generally not the most successful optimisation algorithm on any particular domain"* (1991, p. 59). Davis' argument is exactly that summarised earlier, that one should use whichever representation seems most appropriate to the problem in hand.

This argument holds some considerable weight, both in the theoretical analysis provided by Davis (1991) and in the number of practical applications of non-binary encoded GAs from areas such as Travelling Salesmen problems (Whitley *et al.*, 1989) and chemical structure database search (Jones *et al.*, 1993), to musical composition (Horner & Goldberg, 1991). In addition to these, there is strong evidence provided by the other strands of GA research as to the benefits and applicability of a non-binary representation. Particularly in Evolution Strategies – which are founded upon the ideas of a real valued vector representation – the use of non-binary alphabets has been fundamental throughout an area of research that shows no sign of lacking in either potential or application.

What there seems to be therefore, is no definitive and universal answer to the question - real or binary?

2.4 Alternative Structural Representations

Whilst the immediate thought at the mention of 'representation' is one of encoding schemes, representation may also refer to structural representation. Considerable research has been undertaken recently to look at alternatives to the single binary chromosome string and a number of different approaches have been developed. Arising in many cases out of GA applications, alternative alphabetic encodings have led to alternative genetic operators in order to maintain legality of representation. This in turn has led to the development of structural modifications in a number of areas, manipulating the fabric of the GA chromosome string itself in order to better attack a particular problem (e.g. Davis *et al*, 1993).

However, application areas are not the sole source of chromosome diversification and theoretical research – particularly into deceptive problems – has led a number of authors (e.g. Goldberg *et al.*, 1989, 1991a) to produce differing structural modifications.

These structural alternatives can be summarised by work in a number of areas, which are dealt with in turn:

- Messy GAs.
- Hybrid solutions.
- Segmented chromosomes.
- Hierarchical representations.
- Multiple chromosomes.

2.4.1 *Messy GAs*

Theoretical analysis of deceptive problems, which by definition are difficult for traditional GAs to solve, highlighted the problem of linkage to be an important issue. Approaches such as inversion (Holland, 1975) have not been altogether successful in tackling this issue in practice and this analysis of difficult problems led to the development of the messy GA (Goldberg, Korb & Deb, 1989).

The messy GA (mGA) retained the underlying Darwinian principle of survival of the fittest, but departed from traditional GA approaches quite radically, as explained by Goldberg, Deb & Korb (1991a). The traditional GA's fixed length chromosome string was removed, with variable length strings being used in a method involving *overspecification* or *underspecification* (modification of conflicting genes caused by varying the length in a way appropriate to the problem in hand). Traditional fixed crossover operators were abandoned in favour of *cut* and *splice* operators that manipulated the variable length strings.

The evolutionary process was further split into a *primordial* phase in which potential building blocks are generated and a *juxtapositional* phase, during which the genetic operators are applied to manipulate the genetic material. *Tournament* selection was utilised, inherently addressing the requirements of fitness scaling. The result was a novel GA having split the evolutionary process, abandoned fixed representations and designed new genetic operators.

The results were reported by the authors to be promising, with the mGA having *"always found globally optimal strings"* in the deceptive problems tested in the 1991 paper. The initial research into the mGA has begun to inspire further investigation and application of the structure. Goldberg *et al.* (1989) discussed the relationship of the mGA operators to the schema theorem. Authors have also undertaken further theoretical analysis (Deb, 1990; 1991; Goldberg *et al.*, 1993) exploring particular facets of the behaviour of the mGA. Without a doubt, the messy GA is an area of huge potential, illustrating the benefits in

performance that can be derived from taking a different structural representation to that of the traditional approach.

2.4.2 *Hybrid solutions*

Davis (1991) argues that a successful application of a GA to a particular problem should not avoid exploiting other methods containing useful problem information. GAs are known to identify an area of interest reasonably quickly, but lack the *"killer instinct"* (De Jong, 1985). Hybridisation is another way of attacking this question, with hybridised GA/hill climbing algorithms being a particularly good example of a combined system with the homicidal behaviour needed to finish the job.

Both applications specific and general-purpose GAs have been developed using hybridisation, which has progressed in a number of areas. Two specific areas are discussed here, namely general hybridisation with alternative methods such as simulated annealing and the approaches of Mühlenbein *et al.* (1991, 1993) using the Parallel and Breeder GAs. Both these areas illustrate the use of representational modifications to the standard GA, in an attempt to improve performance.

2.4.2.1 *General hybridisation*

Hybridisation with other systems has been carried out by a number of authors, using the GA to begin a search process and then handing over to more traditionally accepted methods. Examples include Powell *et al.* (1989), whose EnGENEous expert system seeded a GA to find improved domain knowledge and Kido *et al.* (1993) who handed over a GA initiated search to methods like TABU search. Other authors have pursued a hybridisation approach more integrated with the genetic process, rather than handing over a partially complete search.

Simulated Annealing (SA) (Kirkpatrick *et al.*, 1983) has provided the basis for most of these combination methods, illustrated particularly well by Shaefer's (1987) ARGOT algorithm. Shaefer absorbed the SA concept of "temperature" (controlling the detail of the search) to produce a GA with search bounds adapting according to convergence properties of the chromosomes. He also utilised a representational diversion, with ARGOT 'learning' strategies through the use of an intermediate mapping between the chromosome string and the problem space – a distinctly different representational strategy to the accepted GA methodologies.

Other SA based representations include Sirag and Weisser's (1987) proposals for a Unified Thermodynamic Operator (UTO). Again borrowing the temperature concept, their genetic process was modified to replace crossover with the UTO, the temperature (adjusted by a measure of population convergence) controlling the degree of change taking place during crossover. Their paper presented a good example of true hybridisation, substituting a chunk of the GA cycle with a concept derived from another methodology altogether. The ideas behind the UTO have been continually developed, with a recent paper by Varanelli & Cohoon (1995) extending the linkage of the UTO to GAs through the introduction of a population concept. They presented encouraging results, illustrating how the closer ties to the evolutionary cycle allow greater control over the convergence process.

The variety of hybridisation schemes also encompasses the move towards improved computational performance through parallelisation (discussed earlier). The most notable examples are the Parallel and Breeder GAs developed by Mühlenbein *et al.* (1991, 1993).

2.4.2.2 *The Parallel GA*

The Parallel GA (PGA) (Mühlenbein *et al.*, 1991) incorporated into the overall system a number of alternative genetic and non-genetic methods, including the co-evolutionary metaphor provided by subpopulations, modified crossover operators, non-genetic hill climbing and parallel computation.

The PGA operated by application of the genetic process to a number of subpopulations, periodically exchanging members of the subpopulations. If no significant improvement was seen within a certain time period, a hill climbing algorithm was engaged in an attempt to improve the solutions found. The representation of the PGA differed from that of a traditional GA through the introduction of a set of chromosomes, rather than a single chromosome string. Recombination was provided as an additional operator, periodically exchanging homologous chromosomes between sets.

This illustrated a number of distinct differences between the PGA and traditional methods, both in hybridisation and representational terms, similar to the split and multiple chromosome structures to be discussed shortly. However, the large amount of hybridisation with co-evolutionary, parallel and non-genetic schemes puts the PGA into the class of a hybrid GA, as opposed to a purely representationally diverse GA.

Additional work on a PGA has also been carried out by von Laszewski (1991), who extended a parallel GA implementation to include a 'structural crossover' operator. Von Laszewski's application to a k-way graph partitioning problem utilised a string based representation and introduced structured crossover and mutation operators. Structured crossover operated in a similar manner to GP style methods discussed shortly, exchanging a block of graph information. The extension of the work by Mühlenbein *et al.* (1991) by von Lazewski (1991) illustrates the potential given by hybrid schemes such as the PGA.

2.4.2.3 *The Breeder GA*

The Breeder GA (BGA) (Mühlenbein *et al.*, 1993) performed a further hybridisation on the PGA, combining the advances made by the PGA with the ideas of population control put forward in ES. Utilising truncation selection (taking the $T\%$ best individuals and randomly mating them with the population) to imitate the actions performed by human breeders, the BGA came a step closer to the ES (μ, λ) selection strategy. The BGA took further steps in the ES direction by controlling the population convergence through a measure of fitness distribution, utilising the 'response to selection' metric in its selection strategy.

The results presented in the 1993 paper showed promise, illustrating an improvement over the PGA strategy (Mühlenbein *et al.*, 1991). The concepts and theory of the BGA were further explored by Mühlenbein *et al.* (1994) in their analysis of the interaction between crossover and mutation. This paper also identified the way forward for the BGA, specifically mentioning the merits of application to operators analogous to GP subtree exchange and the use of variable length chromosome representations. The BGA has continued to provide a vehicle for research into the beneficial hybridisation of GA research with other areas in EC.

2.4.3 *Segmenting the chromosome*

In the traditional GA a single chromosome string decodes, via a fitness function, to a particular solution on a problem surface. As problem applications and developments in GA techniques have become more complex, this representational method has proven too restrictive. This issue was identified by De Jong (1985) throughout the representation sections of his ten year perspective. He identified a need for research into a number of different areas, relating to adaptive representation, type and length of string that may be restricted by the use of a fixed length, binary chromosome.

2.4.3.1 *Incorporating additional genetic information*

One chromosome segmentation idea has been proposed by a number of authors in their incorporation of genetic operators into the chromosome structure itself. Schaffer & Morishima (1987) explored ideas of self-adaptation through the incorporation of crossover points. Self-adaptation has already been mentioned as an area of interest, specifically highlighted by De Jong (1985) in the ten year perspective. The Schaffer & Morishima approach involved the evolution of crossover points by incorporation of a punctuated crossover operator into the chromosome string.

The result of this was a segmented chromosome, containing genetic information and encoded crossover points, both of which were evolved during the genetic process. Their results indicated that the adoption of an evolved crossover mechanism suggested better performance, with productive crossover continuing, even in the face of a converging gene pool. Other investigations into adaptive operator rates (e.g. Grefenstette, 1986; Davis, 1989; Julstrom, 1995) indicate that dynamic adaptation has benefits. However, the approach taken by Schaffer & Morishima showed, at a relatively early stage in GA research, a forward-looking chromosomally manipulative way of achieving this goal.

A more recent study by Levenick (1995) has taken a similar, but extended, approach to the problem. Levenick leant towards the biological metaphor of introns – inserted genetic material – already investigated in some areas of the GP community. His paper proposed the introduction of a 'metabit' alongside each gene in the chromosome string, dictating whether or not crossover was permissible at that location. This method incorporated both a novel chromosome structure modification, an extension to the Schaffer & Morishima (1987) study and a biological metaphor. The results, indicating benefits for population control, illustrated the methods in which a variation in the chromosome structure has allowed a more flexible GA operation.

Successful results from this kind of insertion methodology are important, indicating a direction for GA research which incorporates both a more flexible chromosome representation and ideas of dynamic configuration (De Jong, 1985; Grefenstette, 1986).

2.4.3.2 *Multi-part chromosomes*

Continuing the line suggested by the approaches detailed so far, one arrives at the idea of splitting a chromosome into distinct representational parts. As an extension of the insertion schemes, several studies have been undertaken that utilise this more flexible approach in applying GAs to particular problem areas.

Kelly & Davis (1991) applied a hybrid GA technique to the development of classifiers, incorporating genetic and *k*-nearest-neighbour techniques. In doing so, they used a real-valued chromosome string, split into 3 sections. Each section dealt with rotation in space, classifier weight evaluation and *k* neighbours' weights respectively. The paper also used an interesting hybrid approach, basing creation of new chromosomes on the ranking of individuals according to classification, rather than traditional genetic selection. As such, it is a good example of a study illustrating the flexibility of a split representation and the benefits of a hybridised approach, producing promising results for *k*-nearest neighbour classification.

Another hybridised approach that incorporated split chromosomes was presented by Bowen & Dozier (1995) in their application to constraint satisfaction problems. Their algorithm used a four field chromosome data structure, recording a variety of information in a manner similar to some ordering problems (e.g. Bruns, 1993). They also used an interesting approach to population diversity, maintaining a measure of 'family' through gene similarity comparisons and restricting the number of members of a 'family' in the population at any one time. Their approach then combined these genetic notions with an 'arc revision' space search, using the results of this space-reduction technique to decide whether to halt the search if a solution was not possible.

Davis *et al.* (1993) gave one of the best illustrations of representational flexibility, having applied a GA with a tripartite chromosome to the design of a robust telecommunications network. Their paper split the chromosome into three distinct parts dealing with network link capacities, routing and survivability respectively. Davis *et al.* also took a further step, designing independent operators that acted only on specific parts of the chromosome string – an approach not taken by the other split-chromosome studies. The scope of previous research into problem-specific operators, applied at a single level to the entire chromosome string solely to maintain a legal representation, was consequently challenged.

2.4.4 *Hierarchical representations*

A natural extension of the approaches discussed above is to increase the independence of functional parts of the chromosome in a hierarchical manner. This bears a striking similarity to the concepts embodied in the GP and CS fields of research. The principle of structured information and manipulation of entire sub-structures is well founded in those areas and can also be seen to have been applied successfully in the pure GA area.

The difference between the simple segmentation of a chromosome and more structured representations primarily exists in the amount of manipulated information. The previous section illustrated typical examples, whereby different parts of a split chromosome relate to different parts of a problem. The hierarchical representation provides a more definitive segmentation, giving greater independence to the chromosome string in its manipulation, as well as coding.

2.4.4.1 *GP style hierarchical encoding*

GP development of programs, especially utilising manipulation of sub-trees and ideas like introns, has provided a great deal of inspiration for alternative representational structures in the pure GA field. Williams *et al.* (1994) clearly illustrated the flexibility of this approach, by applying a GA to the development of a novel neural network structure – the bumptree. Consisting of a number of hierarchically organised gaussian functions, the bumptree structure was translated into a real-encoded GA, with sections of the chromosome string representing (in a similar way to GP) entire subsections of the bumptree.

In evolving different bumptrees, subordinate gaussians were exchanged through the genetic manipulation of the chromosome sections representing that subordinate. This action could be directly compared to the common GP operation of crossover of sub-trees, although GP is still ahead of this type of research by working with sub-trees of potentially different sizes.

The representational approach taken by Williams *et al.* (1994) demonstrates not only the advantage of a hierarchically structured chromosome, but further reinforces Davis' (1991) argument of using the most appropriate alphabetic representation – in this case real valued gaussian parameters. It also indicates how Davis' suggestion of the most appropriate representation can be interpreted in a structural manner, in this case applying an appropriately ordered hierarchical chromosome to a naturally hierarchical task.

2.4.4.2 *The Structured GA*

The structured GA (sGA) takes the hierarchical representation demonstrated by Williams *et al.* (1994) and formalises it into a more general genetic algorithm approach, rather than a specific problem application. The initial sGA model (Dasgupta & McGregor, 1991), and subsequent applications (Dasgupta & McGregor, 1992, 1992a; Dasgupta, 1994), utilised a structured representation designed to inherently represent a number of the issues that have been tackled by other authors (e.g. crowding (Mahfoud, 1992), sharing (Goldberg & Richardson, 1987) and messy GAs (Goldberg *et al.*, 1989)).

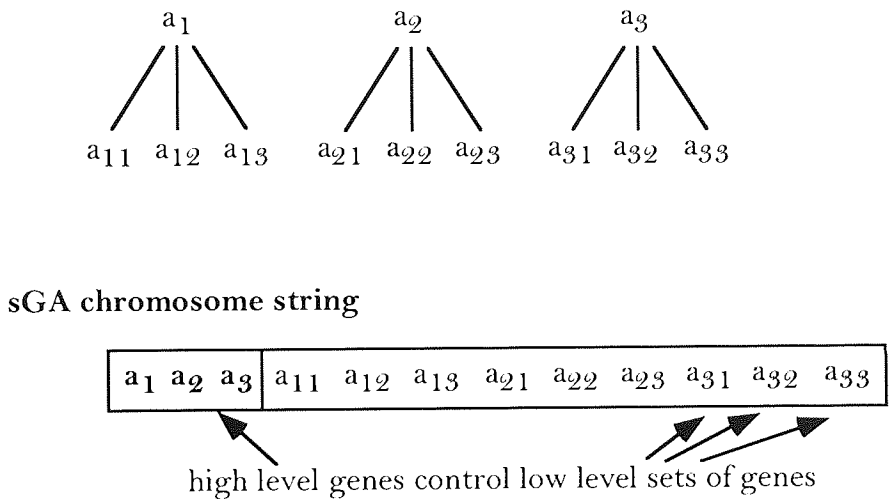


Figure 2.7: The Structured GA representation.

The sGA differed from the traditional GA in its use of a directed graph style hierarchical chromosome representation. Incorporating ideas discussed earlier, namely the use of redundant genetic material and split chromosome functionality, the sGA chromosome decodes into high and low level genes. The state of the high level genes activated sections of the string that referred to the corresponding low level genes, illustrated in figure 2.7.

Utilising this structure, Dasgupta & McGregor reported some success in application to the evolution of neural networks (1992), evolving both network connectivity and weight matrices simultaneously. They also indicated positive optimisation results, and directly compared their method to the messy GA (Goldberg *et al.*, 1992) as a possible alternative, in the field of non-stationary function optimisation (Dasgupta & McGregor, 1992a).

They indicated that the use of the high-level genes to control the activation of lower level genes produced an inherent genetic diversity, a form of distributed

memory and biological plausability (1992a, 1994). The sGA is an interesting extension of Williams' (1994) approach, forming a more general GA using a similar hierarchical approach. Using ideas of integral redundant material, dominance operators and structured representation, whilst maintaining the traditional GA operations of classical crossover and fitness based selection, the sGA is an interesting combination of approaches advocated by other authors (e.g. Schaffer & Morisihima, 1987; Holland, 1975; Williams *et al.*, 1994).

2.4.4.4 SAGA

Harvey's Species Adaptation Genetic Algorithm (SAGA) (Harvey, 1992a) presents a similar approach to that of the sGA, but takes a slightly different line of attack, using the evolution of variable length structures as opposed to the incorporation of redundant genetic material proposed in the sGA. The SAGA structure also includes a number of other ideas, such as the independent contribution of each gene to the final chromosome fitness, the relevance of the subpopulation concept and a more detailed analysis of the role of schema in a variable length methodology. Some of these ideas are similar to those seen elsewhere, in particular to Smith's (1980) LS-1 classifier system (discussed in more detail later on), whilst others seek to address areas Harvey (1992a) claims have lacked sufficient detail when implemented by CS and GP approaches.

The SAGA system operates by allowing the chromosome string to lengthen under the control of an 'increase-length' operator whilst retaining some of the genetic information from the parents. Harvey identifies the effect of this to be the creation of different 'species' of chromosome string within the overall population, introducing the evident scope for extension to the subpopulationary concept. Harvey goes on in further papers (Harvey, 1992b) to extend the discussion of the mechanisms by which the crossover operators and the notion of schema theory to the recombination operator applied in the variable length context of SAGA, noting the need for the exchange wherever possible of homologous segments of chromosome string – an issue identified by Holland (1975) in his proposals for translocation and segregation operators in position independent GA applications.

Harvey identifies the main potential for the use of variable length GA mechanisms (and SAGA in particular) as being in areas of environments of unknown complexity, such as the evolution of an artificial life Animat. In pre-defined domains such as traditional GA function optimisation, Harvey points out that *"one would do best to stick to fixed lengths"* (Harvey, 1992a). The SAGA mechanism has been quite widely explored and developed by Harvey in his

papers (1992a, b, c) and its context in relationship to the research being undertaken in co-evolution and Artificial Life explored in some detail.

He also makes a number of comparisons to Goldberg's Messy GA, Classifier Systems and GP, setting out the SAGA mechanism as distinct in a number of areas - mainly true variable length chromosomes (unlike the messy GA manipulation of fixed length genotypes (Harvey, 1992a)) and by extending the study of schemata and their relationship to the operators he uses (Harvey, 1992b, c). A similarity to the other hierarchically structured representations discussed previously in this section, to the development of cubic and wrap around crossover mechanisms and their associated theoretical analysis is also evident (described in the next section). Harvey's SAGA presents an interesting area of great potential for future research, holding much relevance to the work presented here (discussed in more detail in the next chapter).

2.4.5 *Cubic, wrap-around and multiple chromosome encoding*

The work described so far indicates the trend towards more diverse structural representations. However, the majority of the representations mentioned still work with some form of linear chromosome system, albeit well structured. A number of authors have taken diversification of the chromosome structure to a greater degree, particularly along the lines of cubic multi-dimensional representations and distinct, independent chromosomes.

2.4.5.1 *N-cube representation*

In a recent drive to attack problems of an inherently multi-dimensional nature, a number of authors have proposed alternative operators and chromosome structures that attempt to reflect this multi-dimensionality. A paper by Beasley *et al.* (1993) suggested a method of encoding referred to as 'expansive coding,' splitting a problem into sub-problems concatenated together on the chromosome string. Whilst the representation maintained a concatenated string, their genetic operators treated chromosome sub-problems as a distinct 2D structure, utilising this metaphor in the exchange of 2D blocks through crossover. Their operators strictly limited themselves to the sub-regional 2D blocks – an approach also adopted by others. Their results demonstrated an increased complexity and size of chromosome string, but a reduced complexity of problem for the GA to solve.

The multi-dimensional representation idea was also taken up by Watabe & Okino (1993) in their study of genetic shape design. Their approach differed from that of Beasley *et al.* (1993) in the direct translation of an *n*-d problem into

an n-d chromosome of real numbers. They implemented this using an array structure of floating point values representing the genes. This representation was then hybridised with the Free Form Deformation (FFD) technique to evaluate the shape being defined, through genetic manipulation by operators that exchanged randomly sized subgrids in the array structure. Their paper illustrates a combination of multi-dimensional representation and operators, distinct from that given by the Beasley *et al.* (1993) approach.

That combination has recently been further explored, with studies beginning to explore the effects of such multi-dimensional exchanges. A study by Bui & Moon (1995) investigated the formal generalisation of multi-point crossover to n-dimensional encoding. Their paper also took the first step towards detailed formal relationship of this multi-dimensional structure back to the schema theorem, relating examples of schema formation and preservation in n-dimensions and the effects of n-dimensional crossover.

The study of n-dimensional representation proposed in the Bui & Moon (1995) paper was continued by Kahng & Moon (1995) in their analysis of recombination mechanisms in n-dimensional encoding schemes. Their study identified the weakness in research of schemata understanding in multi-dimensional representations and provided an analysis of potential schema effects in an n-dimensional crossover mechanism. Their results suggested applicability of the proposed multi-dimensional encoding scheme within traditional linearly represented chromosome structures, as well as relating the proposed geographical crossover in n-dimensions to current crossover methods. In an analysis of future research required, they identified the need for greater research into multi-dimensional structures and generalisation into n-d structures utilising different lengths of chromosomes across dimensions.

2.4.5.2 *Distinct, multiple chromosomes*

Whilst the research identified thus far contains radical departures from the traditional linear chromosome representation, none of the studies mentioned represent a wholesale departure from the traditional method. All retain some measure of type similarity in representation or linear chromosome representation (albeit hierarchically organised or segmented).

Holland (1975) identified an extension of GA potential through an increase in independence of chromosomes, moving to a set of homologous n-tuples controlled by mechanisms of dominance and translocation. This indication of a multi-chromosome structure was still a step further than studies discussed

above, although the trend described is moving in that direction. A distinctly multi-chromosome structure has been adopted by Juliff (1993) in her application of a multi-chromosome GA to pallet loading.

Juliff used a hybrid genetic algorithm combined with an intelligent load builder in the optimisation problem of beer delivery. The paper described a GA structure with three independent chromosomes within an individual applied to different facets of a multi-dimensional problem. Chromosomes were independently responsible for packing beer onto a pallet, pallets onto a truck and ordering the pallets for packing. Her proposals also indicated independence of operator application and representation, ensuring that legality of representation is retained - an issue already discussed and mentioned by Davis (1991) frequently in his discussions of problem specific applications. Juliff's multi-chromosome approach takes the structural independence and flexibility implied by the linearly represented studies mentioned so far to its more natural conclusion, separating out the functional parts into distinct chromosome representations altogether.

However, the system retains a significant measure of hybridisation to non-genetic processes by passing resulting chromosome information into an intelligent load builder that makes the decisions. This places a hybridised distinction on the Juliff approach compared to that of the purely genetic approaches suggested by Harvey and other areas of GA research such as Classifier Systems (discussed in the next section). Nonetheless, the results presented by Juliff (1993) indicated positive performance compared to other GA methods and implemented a number of the suggestions outlined by other authors investigating structural diversification.

Juliff's study also took a further step in the line indicated by the continuing trend in the GA community for diversification away from linear representations. As GAs are applied to more complex, multi-dimensional problem areas, greater structural and representational flexibility would seem to be a sensible way forward. Indeed, this requirement was identified by both De Jong (1985) and Holland (1975), with the mechanisms for achieving this explicitly suggested by Holland.

2.4.5.3 Pure GA structural alternatives and other areas of EC

Whilst the discussion so far has related to structural developments in the area of pure GA research, it has already been mentioned that there is considerable similarity amongst these approaches to the other areas of research in

Evolutionary Computation. The similarity between the use of some structural alternatives and the GP style use of subtrees was identified in section 2.4.4.1. However, the recent trend towards independence of elements within the chromosome string and distinct separation of chromosomes has great structural relevance to the work that has been undertaken elsewhere, particularly in the field of Classifier Systems.

Whilst using the GA mechanisms to evolve new production rules within the classifier systems, the classifier system itself has developed along different lines to research in pure GAs. However, examination of the structures used within many classifier systems reveals striking similarities, particularly relevant to the direction being taken by recent pure GA structural developments outlined in this chapter so far. The similarities can be demonstrated very graphically by examination of an early Classifier System - the LS-1 system proposed by Smith (1980).

Smith's Learning System 1 consists of a population of knowledge structures, as opposed to individual functional units, with a performance measure evaluating the knowledge structure entities as a whole and the GA evolving classifier rules by manipulation of the knowledge structures. Each individual (fixed length) rule in the production system makes a contribution to the knowledge structure, representing the set of rules on which the GA makes its evaluation. The learning component of LS-1 uses a production system memory consisting of an unordered list of fixed length rules (not of fixed size). In addition to the traditional genetic operators, additional crossover mechanisms are applied to exchange rule components within the knowledge structure (identified in the pure GA context by Holland's (1975) discussion of translocation and segregation).

The 'LS-1 critic' responsible for credit assignment calculates the payoff by application of a weighted relationship of the components performance to the knowledge structure's evaluation. Smith goes on to discuss the relationship of the various weighting factors and growth of the knowledge structures on performance, identifying a number of issues and areas for future development of the system.

From this description, the similarity of Smith's system to a number of the current approaches being developed in the pure GA research field - particularly the use of knowledge structures as compared to Juliff's (1993) independent chromosomes and Harvey's (1992a) independent gene contributions to the variable length SAGA genotype - becomes evident. In addition, the

independence of the knowledge structures opens up immediate observation of the potential of such a system for application to subpopulationary evolution.

The additional underlying theme of co-operation of component classifier items has been further developed in a recent paper by Bull, Fogarty and Snaith (1995). Their paper, evolving a quadrupedal robot with communicating, co-operating Classifier Systems builds on the ideas of contribution of knowledge structure to overall evaluation outlined by Smith (1980). It also presents a similar parallel to the pure GA work of Harvey's (1992a) SAGA mechanism and the gene to genotype fitness relationship, along with overtones similar to the future direction implied by a number of the other structurally novel GA mechanisms recently proposed and discussed here.

In summary, Smith's (1980) LS-1, its development and application (e.g. Smith, 1983; Greene & Smith, 1987) and other related work in the Classifier Systems field highlights the level of similarity and cross co-operation that exists within the different strands of GA research. In addition, it demonstrates that the recent trend by authors in the field of pure GA research to move towards more structured chromosomal representations, away from the traditional linear chromosome, shows promise and may well have distinct advantages in taking lessons from the other areas of EC research that have been going on around these developments. It is this theme that is pursued further in the remainder of this thesis.

Chapter 3: The Multi-Chromosome Approach

3.1 Inspiration and conceptual design

The previous chapter illustrated the movement of the GA community towards architectures and structures allowing a greater operational and representational flexibility. It is not uncommon in GA studies to see the authors refer to a biological or natural precedent for their method (e.g. Dasgupta, 1991; Holland, 1975). Following this guidance, high-level considerations of natural processes led to the identification of what would seem to be an anomaly between current pure GA methods and the natural world.

3.1.1 The qualities of natural organisms

If one considers life in the natural world, a variety of highly complex organisms can be seen, from single-celled amoeba through to the higher forms of life. The associated increases in complexity of organisms up the evolutionary chain is reflected in their genetic encoding and DNA, correspondingly showing an increase in complexity. Bacteria and viruses have a simple structure efficiently designed for self-replication, whereas the DNA defining a human being is a complex double-helix structure.

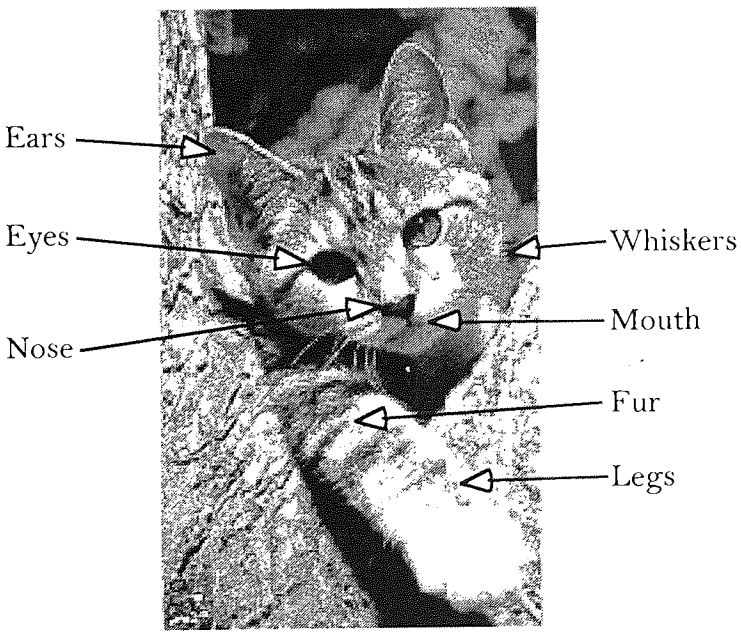


Figure 3.1: Higher life forms are an amalgamation of several component features

Simple observation of the higher forms of life shows that, whilst they are an integrated unit, they consist of a number of component features – independent in nature – that make up the whole. For example, as shown in figure 3.1, a

complex creature such as a cat is made up of a number of component parts. Each of the legs, eyes, ears, fur, tail, organs etc. all combine and act together to form the complete creature, competing in the world for survival.

Whilst all the component parts are still 'bits of cat' and inseparable from the cat, it is conceivable that they may make an independent contribution to the fitness of that animal. The nature of the contribution will be highly dependent upon the environment in which the animal is trying to survive, but each component may be considered to have an independent 'fitness', contributing to the ability of the animal to survive.

Consider a population of cats. Each cat has eyes, ears, a nose, fur etc. Some cats will have better eyesight, sense of smell and different fur patterns from others. Good eyesight can be interpreted as the eyes having a high fitness. For good hearing, the ears have high fitness, but this would be independent of the fitness of the eyes. Only when the two are brought together within a particular animal in its specific environment does the combination become important. If an animal has to hunt its own lunch, then eyesight and hearing are important. If it gets fed by people, cute fur colour and fluffiness might be more important to its survival!

Bringing the discussion back to the GA metaphor, it can be seen from this example that an individual in a complex environment might consist of a number of component features. Whilst each makes a contribution to the overall fitness of the individual, the component parts of the individual may have a high degree of independence from one another and, as such, evolve in different ways within the individual. It is the concept outlined by this example that is embodied within the ideas for the multi-chromosome structure.

3.1.2 *Over-simplification in current representations*

When considering this discussion in the light of current GA methodologies, it could be surmised that the traditional GA has an architectural simplicity approaching that of the amoeba! Whilst the representational complexity of the GA makes this a rather broad generalisation, the metaphor is appropriate when one considers the traditional GA with singly applied rates and a single fitness function. Only recently, with the more structured GA representations discussed in sections 2.3 and 2.4, has any attempt been made to create an appropriate independence within the traditional GA mechanism.

It is conceivable that, despite the flexibility of encoding schemes utilising binary (or non-binary) representations, problem applications to which a GA might be applied are inappropriate to encode in such a linear fashion. Whilst representation may be possible through the use of highly sophisticated encoding, operators and fitness functions, such a methodology may introduce a higher degree of complexity into the traditional GA.

Hybridisation and modifications to Holland's (1975) GA structures are now widely accepted and advocated as necessary in many problem domains (Davis, 1991). Looking back to Holland's (1975) proposals, his explanation of the more advanced behaviour of the GA included references to positional independent coding schemes and suggestions for independence of operators, using segregation and translocation within sets of genes to evolve independent parts for greater representational flexibility. Rather than producing more complex traditional GA encodings and fitness functions, the multi-chromosome approach utilised here brings together a number of different concepts. In particular, these include the trend for structural flexibility illustrated by recent pure GA papers (e.g. Harvey, 1992a, Dasgupta, 1991; Mühlenbein, 1991; Juliff, 1993; Williams, 1994), the concepts seen in nature, Holland's (1975) proposals and ideas explored in the other areas of EC research (e.g. Classifier Systems, Smith, 1980; Smith, 1983; Greene & Smith, 1987).

3.2 Defining a multi-chromosome GA

The concept outlined above has been translated directly across into a GA structure, to produce the multi-chromosome GA (Multi-GA). From this structural base, the exchange of genetic material intended to evolve both feature chromosomes and to group together feature chromosomes within an individual, was developed. The potential for future developments made possible by the new structure was also identified and a number of steps taken to realise this potential, resulting in the development of genetic processes typically unseen in the traditional GA approach.

3.2.1 Structure

The core feature of the Multi-GA structure is a number of independent feature chromosomes grouped together to define a single entity, in a similar manner (in fact devised in parallel) to the approach taken by Juliff (1993). This has resulted in a minor change of expression from that used within the traditional GA. In the traditional GA context, a chromosome string is frequently referred to as a member of the population. In the Multi-GA structure, an additional level of

structure is effectively being introduced, requiring clarification of the terminology.

From this point on, the definitions of items within the population are as follows:

Chromosome = feature chromosome, an entity within a multi-chromosome individual.

Individual = a member of the population, consisting of a number of feature chromosomes.

Population = a collection of Multi-GA individuals (no real change from the traditional GA interpretation).

References may also be made during operator discussions to *Individual level* and *Chromosome level* (or *Feature level*) operations. Individual level refers to those operations manipulating entire component chromosomes within the individual. Chromosome or feature level refers to operations taking place within component feature chromosomes of an individual. Figure 3.2 illustrates the full potential of the Multi-GA, including feature chromosome fitnesses – not always realised in our studies.

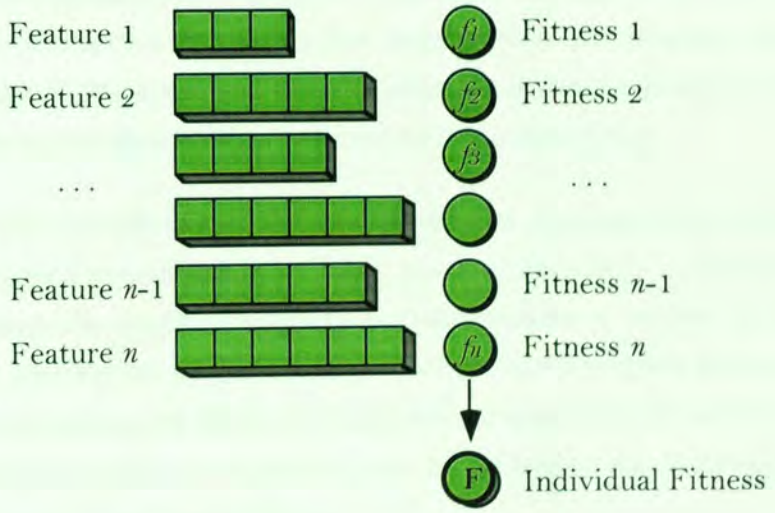


Figure 3.2: The Multi-GA structural representation.

This figure illustrates a single Multi-GA individual, showing the component chromosomes, their independent fitness functions and the overall, individual fitness function on which the entire organism is evaluated. Each feature chromosome has the capacity to operate in the same way as a traditional GA chromosome string, potentially with its own representation, operator rates, genetic operators, string length and fitness function behaving in a manner

unaffected by the other chromosomes in the individual. The relationship between them is governed by the individual level fitness function.

As can be seen from this illustration, the structure is inherently more flexible than the traditional GA approach and allows for a considerable independence of representation and manipulation, although utilisation of these facilities is not compulsory. The structure provides a method for adaptation to the degree required by the problem in hand and is able to act in exactly the same way as a traditional GA if so required. However, there are a number of additional representational and genetic features available.

3.2.2 *Basic genetic operations*

Structural differences are only a part of the distinction between the Multi-GA and traditional GA. Modification to produce a number of independent chromosome strings also required a reorganisation of the genetic operators and processes, in order to realise the conceptual goal. As outlined earlier, the independent features of an organism may well be capable of developing independently of one another. Their interdependence is governed by the individual in which both sets of features are contained. This argument was also advanced by Holland (1975) in his discussion of segregation and translocation operators. Holland's proposals for segregation (to identify homologous chromosomes within a set) and translocation (to ensure exchange of information between those chromosomes) are embodied in the Multi-GA.

The Multi-GA makes use of the traditional GA operators, not requiring any custom designed operations in its basic mode of execution. The application of these operators is modified slightly but the operators remain fundamentally unchanged, leaving the Multi-GA as a non-hybridised method utilising the pure GA concepts outlined by Holland (1975) and subsequently developed by others. Basic genetic operations take place at two levels within the Multi-GA structure – Individual and Chromosome level.

3.2.2.1 *Individual level operations*

Selection is performed exactly according to traditional GA methods, with individuals selected from the population on the basis of their *individual level fitness*. This embodies the principle that the fitness of an individual, and hence its survival in the environment, is governed by the overall combination of the contributions made by the features making up that individual. Standard GA

selection algorithms, with or without varying degrees of elitism, can be applied to the Multi-GA.

Crossover is also utilised in the same way as traditional GA methods, with the exception that it may be applied at *both* levels in the Multi-GA. *Individual* level crossover involves the application of the crossover operator to manipulate entire feature chromosomes across two selected individuals. Returning to the biological example, the motivation for this operator is to exchange an entire set of ears in one cat with those of another. Because the individual is assessed on the basis of the combination of its feature chromosomes, the evolutionary process must include some method of exchanging features across individuals in order to evolve an individual with the optimum *combination* of features. Individual level crossover fulfils this requirement. Again, the popular crossover operators can be implemented at the individual level - namely *n*-point or uniform crossover.

3.2.2.2 *Chromosome level operators*

Having selected individuals and undertaken exchange of feature chromosomes by crossover, genetic operators are then applied at the chromosome level. Both crossover and mutation are applied here, in order to improve the quality of feature chromosomes. Operations at the individual level result in the optimum combination of feature chromosomes currently present in the population, but make no attempt to actually improve the quality of those feature chromosomes. In the biological example, it would be akin to swapping eyes, ears and noses in cats but at no point exchanging *between* sets of eyes in order to produce better eyes. This action is carried out by chromosome level genetic operators.

Chromosome level crossover and mutation are applied in exactly the same way as the traditional GA operators, taking material from two identical feature chromosomes in the selected individuals and exchanging. The rate at which chromosome level operations are applied may be different from the individual level crossover, allowing flexibility in the rate of genetic manipulation at both levels in the structure. Once again, any of the popular crossover and mutation algorithms may be utilised, as appropriate to the representation of the feature chromosome in question. The important point to remember is that genetic exchange only takes place between peer feature chromosomes – that is chromosome 1 exchanges only with another chromosome 1. No inter-chromosomal exchange mechanism is currently employed by the Multi-GA, although it is not inconceivable that this could take place if required.

3.2.2.3 *Fitness function calculations*

Fitness functions may be calculated at both individual and chromosome levels. At the chromosome level, the fitness function is applied in exactly the same way as it would be with a simple, traditional GA. The feature chromosome is evaluated and a fitness value assigned, appropriate to the problem in hand. This is repeated for each feature chromosome in the individual.

The individual level fitness function is a more difficult proposition, as it is this function which has to relate the component features together. A specific method for doing this is inappropriate to define here, as it will depend entirely on the problem to which the Multi-GA is being applied. However, it is not acceptable to entirely dodge the issue and the problem of how to gain a suitable combination of potentially different feature level fitness function representations was given some thought.

Proposals for tackling this method in the Multi-GA used during this study revolved around a summing of relative contributions. Should the problem representation of each feature chromosome be identical, then the individual level fitness function is not difficult to calculate. The combination of the features (of the same type) would be defined in some way by the task being tackled and reflected in the fitness function applied.

However, when tackling a combination of different representations, the question becomes more complex. In such cases, comparison of differing chromosome representations would be achieved by a process of normalisation. The relative contribution of each chromosome to the individual would be obtained by normalisation amongst its peers, repeated for each feature chromosome in the individual. The resulting series of normalised (if necessary, weighted) values can then be combined by the individual fitness function.

In this manner, the combination of fitnesses in an individual composed of diverse representations can be achieved, whilst retaining the chromosome level requirement of maintaining fitness relative to one another, required for accurate evolution of each feature chromosome.

3.2.2.4 *A summary of Multi-GA operations*

Figure 3.3 illustrates the main points of Multi-GA operation, described in detail in sections 3.2.2.1 - 3.2.2.3.

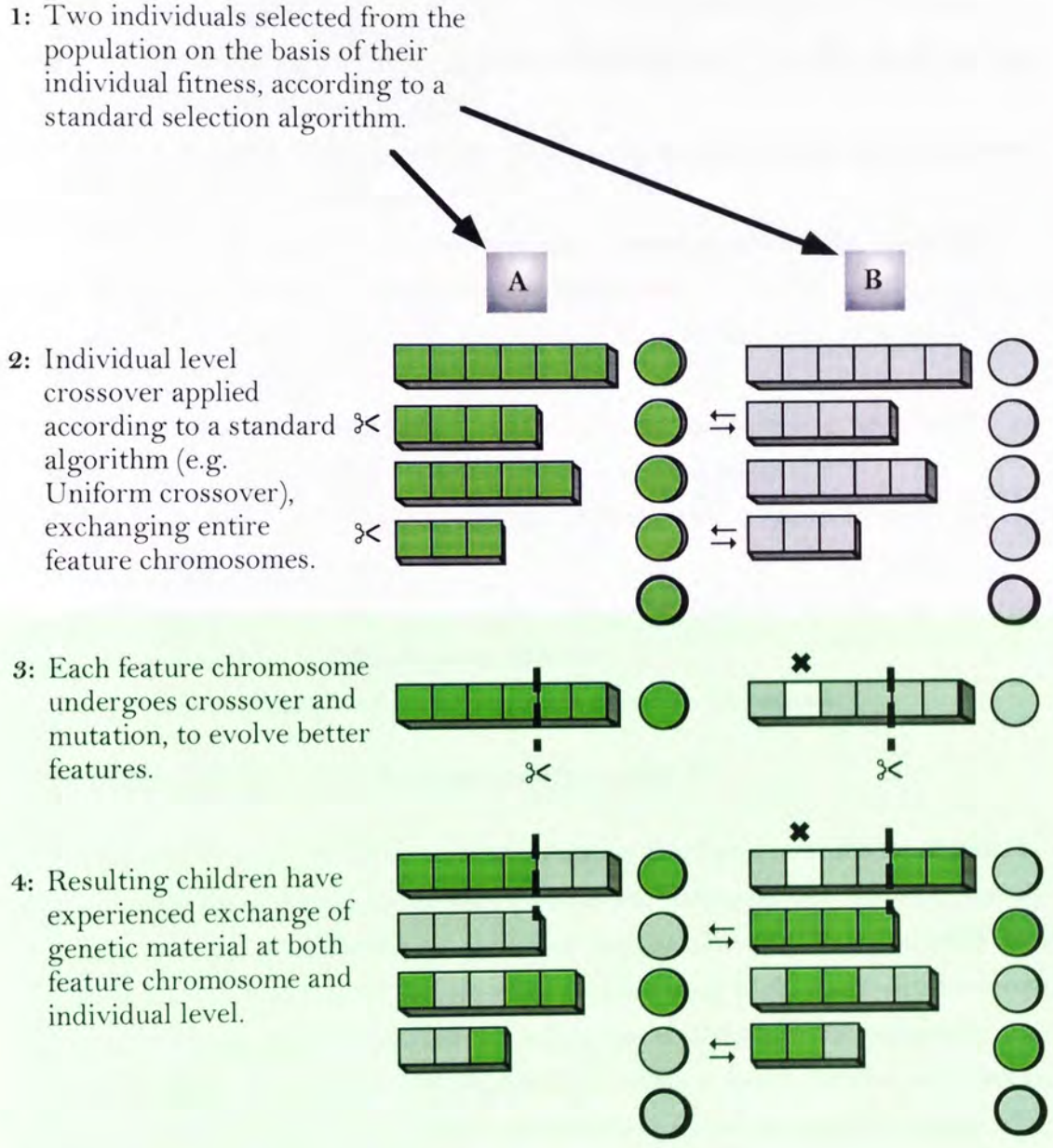


Figure 3.3: Multi-GA operations.

In summary, the main features of the Multi-GA operation are:

- Individuals consist of (potentially) independent feature chromosomes.
- Feature chromosomes may have different lengths, representation, independent operators and rates of application and their own distinct fitness function.
- Selection takes place at the individual level, on the basis of the individual fitness function.
- Individual level crossover exchanges entire feature chromosomes between selected parents.
- Chromosome level crossover and mutation allow the evolution of better quality features within an individual.
- Chromosome operations may be applied at a different rate to individual level operations.
- Chromosome operations occur only on homologous features – no inter-chromosomal exchange currently takes place.
- Chromosome fitness functions indicate the relative fitness of each feature in the individual.
- Individual fitness functions assign a fitness according to the combination of chromosome fitnesses.
- Fitness functions at both levels are problem dependent.

3.2.3 *Increased potential and advanced genetic operators*

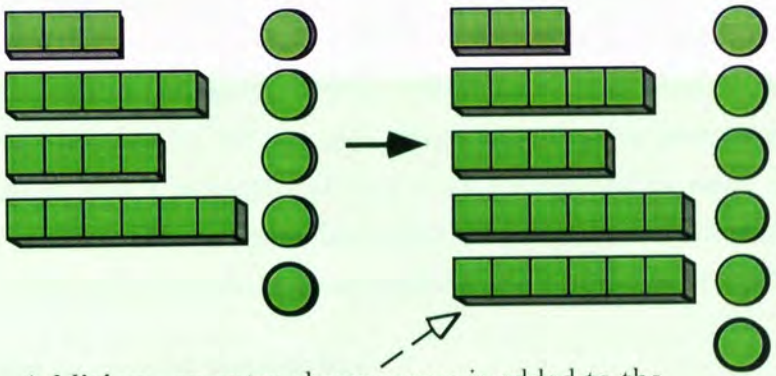
As illustrated, the Multi-GA structure contains great potential for flexibility, in areas including chromosome representation, independent application of appropriate genetic operators, rates of application and the potential for application of the genetic metaphor to problems of high multi-dimensional capacity. As opposed to a more complex linear chromosome encoding and fitness function, the Multi-GA approach provides a more natural, structural representation in which to express a multi-dimensional problem by means of a combination of simpler, but related, sub-representations.

The Multi-GA structure also releases additional potential for new approaches, techniques and applications that make use of the extra flexibility. In some cases, the structural similarity across areas of EC research embodied by the Multi-GA approach may indicate previously untried approaches to existing problems. In other cases, such as those explored later and discussed in chapter 7, recent approaches (e.g. dynamic parameterisation, population geneology, co-evolution) have the potential for further exploration by making use of the structural

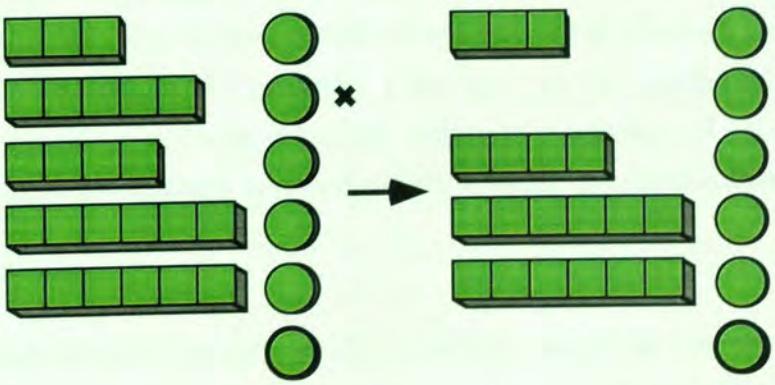
benefits given by the Multi-GA. This may prove to highlight new real-world problem application areas for successful GA application, in the same way as those explored in chapter 6 of this thesis have been developed. For now, the discussion is limited to a description of two new Multi-GA structural genetic operators that have been investigated during the course of this thesis – addition and deletion.

3.2.3.1 *Addition & Deletion operators*

Chapter 6 will describe how the Multi-GA has been applied to a spatial analysis application, leading to the development of two new operators – addition and deletion. The purpose of these operations was to dynamically adapt the chromosome structure of an individual, under the control of the individual itself. The specifics of the problem application and motivation behind the development of the new operators are discussed in chapter 6. The effect of applying the addition and deletion operators is shown below in figure 3.4 and can be seen to confirm the structural potential of the Multi-GA in this context.



Addition: an extra chromosome is added to the individual, producing an individual with $n+1$ feature chromosomes.



Deletion: a selected chromosome is removed, producing an individual with $n-1$ feature chromosomes.

Figure 3.4: Addition and Deletion operators.

Triggered by problem specific circumstances, the addition and deletion operators create and destroy new chromosomes within an individual. When called, the creation operator augments an individual with one new chromosome, typically randomly initialised, of a type and parameterisation specified by the user when configuring the Multi-GA. The new chromosome then becomes a functional part of the individual, as with the pre-existing feature chromosomes, producing a population of individuals containing a varying number of chromosomes. The scope of creation and deletion operators investigated has been restricted to a single representation, but the extension to mixed type creation would not be difficult.

When conditions specified by the problem are met, the deletion operator is called and removes an identified chromosome from the individual. In the current application, the candidate chromosome is identified by a combination of the problem requirements and the creation mechanism. Problem specific applications aside, the potential for self-modification of the genetic structure through the addition and deletion of genetic material is demonstrated by these operators. Addition and deletion incorporates the principles of a number of current research areas, specifically insertion of genetic material (Levenick, 1995) and self-adaptation (Julstrom, 1995). Similarities may also be drawn to the principles approached by the variable length modification concepts of Harvey's (1992a) SAGA and Lindgren's (1991) work (compared in more detail in the following section). The purpose for which the additional chromosomes may be utilised is at the behest of the user in applying the Multi-GA in a problem area.

3.3 Comparison to current approaches

One of the key questions with any new technique is "how does it differ from approaches taken by other people?" Chapter 2 illustrated recent pure GA trends towards more structured representations and operators, of which this work is an extension. The Multi-GA provides a new perspective on many of the ideas proposed by other authors, bringing together a number of ideas currently scattered amongst different approaches into a single genetic mechanism.

3.3.1 Pure GA methods

As discussed briefly in section 3.1.2 earlier, many of the traditional GA approaches contain a linear chromosome representation, giving a marked distinction between the Multi-GA approach. However, the more recent trends discussed in section 2.4 reveal the moves of pure GA researchers to a more complex representation.

For the majority of these cases, there is an increase in complexity, but still a linear feel to the chromosome representation used. Whilst approaches like the structured GA (Dasgupta, 1991) and segmented chromosomes (e.g. Davis *et al.* (1993)) introduce a great deal of ordering, they still maintain a single linear representation and do not break up the fitness or operator structure in quite the way undertaken by the Multi-GA. The approach set out here formalises the structural organisations proposed by authors such as Dasgupta (1991) and Williams *et al.* (1994) into a more distinct, multi-dimensional representation. Consequently, the distinction between chromosome segments and the ability to apply different operators to segment parts becomes easier to implement, with the Multi-GA representing an advance on the work carried out in these areas so far.

Some authors have already taken further steps towards this approach, with systems like SAGA (Harvey, 1992a) treating genes on the variable length linear chromosome string as having a distinct contribution to the overall fitness. The similarity between this approach and the Multi-GA combination of independent chromosome fitnesses is evident and a number of the ideas touched upon by the SAGA system are incorporated into the concept and implementation of the Multi-GA structure. However, the degree of problem representation flexibility is expanded by the Multi-GA structure. The independent feature chromosomes provide the potential for alternative type representation within the individual, allowing the ability to represent sections of a multi-dimensional problem that would be difficult to envisage with a single (albeit independently contributing) gene on a chromosome string. In addition, Harvey's (1992a) investigations into the use of operators governing the length of his SAGA chromosomes are extended to the Multi-GA concept, with the study of the addition and deletion operators establishing their role in the governance of feature chromosome propogation within Multi-GA individuals.

The use of independently contributing fitnesses is not restricted to the SAGA system. An earlier paper by Lindgren (1991) demonstrated a pure GA system tackling the Prisoner's Dilemma problem and explored both dynamic increase in genome length and (in a similar manner to SAGA) calculated a final fitness value by the combination of a number of component fitness solutions. In addition, Lindgren's discussions on the mechanisms for control of the population and the evolution of species provide ideas explored by both Harvey's SAGA and the Multi-GA. Of particular relevance is the investigation (outlined in chapter 6) of the effects of the Multi-GA's addition and deletion operators on population behaviour. This analysis of the mechanisms of dynamic population control

using addition and deletion explores behaviour similar to that of Lindgren's (1991) 'extinctions'. In doing so, the Multi-GA provides a vehicle for research, identified by Lindgren as necessary, into the effects on population behaviour of dynamically controlled self-adapting individuals.

Despite the evident similarities, both the Lindgren and Harvey approaches maintain the use of an essentially linear chromosome representation, albeit with some interesting and highly useful ideas for species evolution. The work investigated here using the Multi-GA structure provides a more structured, open representation than that proposed so far, allowing the independent feature chromosomes a greater freedom of representation. Whilst using suggestions and proposals similar to those implemented by Harvey, it has been successfully applied to a fixed-domain problem (chapter 6) – an approach which Harvey considered to provide little benefit for SAGA and other systems with species evolution capability. This serves to demonstrate the usefulness of applying ideas explored by some authors in other application areas, allowing a similar, but slightly diverse, approach to provide new problem applications for concepts that may not intially appear to be easy to implement.

However, chapter 2 also pointed out that the pure GA field has already moved on from the single, linear representation into more structurally diverse areas. Work undertaken to investigate the effects of n-cubic representation and multi-dimensional operators (e.g. Bui & Moon, 1995) is highly relevant and may have direct application to the multi-chromosome structure. Juliff (1993) has already implemented a distinctly multi-chromosome approach to a problem application and the structure implemented here – developed in parallel to Juliff's beer stacking application – can be seen to have similarities. However, Juliff's (1993) multi-chromosome approach is extended here by the reliance on a purely genetic process, rather than a hybridised, external intelligent load-builder. Whilst hybridisation is an approach carrying high recommendation, the Multi-GA is restricted to hybridisation of representation and structure within the GA metaphor. It does not attempt at this stage to hybridise itself with other AI techniques, in the way proposed by Mühlenbein *et al.* (1991). In addition, these studies provide a more specific structural investigation of the multi-chromosome concept, with additional operators and the general-purpose characteristics of the Multi-GA highlighting a distinct move forwards from the Juliff approach.

3.3.2 *Related research elsewhere in EC*

Chapter 2 also mentioned work in the other fields of Evolutionary Computation of great relevance. This manifests itself in a number of areas, from structural similarities and the borrowing of concepts, through to the potential to expand pure GA research into other current research areas. It is the high degree of cross co-operation between the fields that becomes evident as one examines the work undertaken here in more detail.

Perhaps the most evident similarity, is the structural likeness to work undertaken in the field of Classifier Systems. In particular, Smith's (1980) LS-1 system and its developments (e.g. Smith, 1983; Greene & Smith, 1987) share a number of common themes with both the Multi-GA structure and some of the related pure GA research already discussed. Whilst set in a different context, it is obvious that Smith's use of independent knowledge structures contributing to the success of a classifier's evaluation is similar in concept to the independent chromosome fitness evaluation within an individual. In consequence, it can also therefore be related to the independent single gene fitness contributions (discussed in the previous section) of Harvey's (1992a) and Lindgren's (1991) work, along with the multi-chromosome aspects of Juliff's (1993) study. Smith's LS-1 also incorporated a mechanism for the exchange of components within the knowledge structure - a function undertaken by the individual level crossover of the Multi-GA.

Ideas such as this, backed up by an established use of such concepts within the CS field (demonstrated by work such as the communication between independent classifier systems within Bull, Fogarty and Snaith's (1995) quadruped robot), illustrate the merits of exploration of alternative structures shown to have success in other areas. This is true not just of Classifier Systems, but can be seen throughout the EC field in both future potential (discussed in Chapter 7) and current research.

Other notable examples already discussed include the co-evolutionary metaphor, with independent subpopulations evolving to produce localised solutions to a part of a problem. This would seem to be similar to the application of a feature chromosome to the particular problem - indeed, the Multi-GA structure could be applied in such a way as to provide a distinct mechanism for subpopulationary evolution in the manner carried out by many researchers today. However, there is a potential for future expansion within the Multi-GA structure, allowing definition of the inter-relationship between the

subpopulations in a way not easily exhibited in most current GA representations of co-evolutionary strategies. A good example of this is that of timetabling, discussed in chapters 6 and 7.

In addition to structural mechanisms, the Multi-GA discussion has also outlined the potential for the use of dynamic parameterisation. Dynamic parameterisation, explored in more detail in chapter 5, has been an area showing promise in both the GA and ES fields. The Multi-GA structure used here provides, not a particularly novel mechanism for dynamic parameterisation, but an interesting vehicle for the application of established dynamic parameterisation techniques. In addition, the two dimensional nature of the structure does open a debate as to the most effective method of dynamic configuration between the two levels of the structure – an issue which has not necessarily arisen in GAs using a linear chromosome string. The discussion of dynamic parameterisation undertaken in chapter 5 introduces it as a method which may have benefits within the Multi-GA context, and addresses some of the issues of existing dynamic parameterisation research that become appropriate when dealing with a structure of higher dimensionality.

3.3.3 *What does the Multi-GA achieve?*

Having identified a number of similarities with existing approaches, both inside and out of the pure GA field, the natural conclusion is to ask what is actually achieved by the Multi-GA? The answer can be gained from consideration of a number of different perspectives.

The similarity between LS-1 and other classifier systems brings an indication of the level of cross co-operation between the fields of EC research that remains, as yet, fully unexplored. In the pure GA field it can be seen that a number of authors are skirting around structural ideas already implemented quite successfully in a different application context. It may well prove to be useful to take these ideas and look once more at their application to areas of research in fields such as function optimisation and pure GA specialisation to see if the currently evasive multi-dimensional problems can't be better tackled. The Multi-GA work goes part way to addressing this question, providing a direct example of a co-ordination of similar ideas from within and without the pure GA field into a distinct structure, identified as applicable in a pure GA context. The potential for dynamic parameterisation (e.g. Grefenstette, 1986) and GP style variable length crossover manipulation (Koza, 1992) is also incorporated into the Multi-GA structure, through the provision for independent parameter rates

to be associated with each chromosome. Self-generation of genetic material through addition and deletion may also provide an avenue for fruitful investigation, as well as incorporating proposals put forward by authors such as Levenick (1995), who advocated modifying the genetic material by insertion.

Finally, the Multi-GA embodies the proposals put forward by Holland (1975) for a structure incorporating sets of homologous chromosomes, controlled by segregation and translocation operators. It also incorporates a number of De Jong's (1985) 'ten year perspective' visions relating to adaptive representations (not necessarily of fixed length) and the investigation of strategies like sub-populations and self-adaptation.

In summary, the Multi-GA brings together a number of suggestions and directions that current pure GA research is moving towards into a distinct, flexible and purely evolutionary mechanism. In doing so, it provides a useful platform for the further development of these ideas, and others of a highly relevant nature from the associated fields of Evolutionary Computation. Given that some areas of pure GA research appear to be moving towards greater usage of multi-dimensional representations, an opportunity to refocus on the achievements and issues studied by a variety of authors - embodied into a single, distinct structure - is a valuable and worthwhile exercise.

3.4 Design and Implementation in Software

Implementation of the Multi-GA structure required careful thought, in order to best overcome the software engineering difficulties associated with the use of different types within a single structure. Software development and experiments were carried out on SUN UNIX systems running initially under SUNOS 4, subsequently upgraded to Solaris 2.3. In making the decision as to the best method of implementing the ideas encompassed by the Multi-GA structure, a number of factors were considered including ease of maintenance, language simplicity and flexibility, machine performance, software support and future development potential.

Perhaps the deciding factor was the need to implement flexibility of representation within an individual, presenting a requirement for ease of type definition within the chosen implementation language. In addition to its well engineered modular design, the recently introduced facility of 'templates', allowing the parameterisation of types within C++, made that the natural choice. As a result, the Multi-GA developed as an object oriented program, functionally decomposed for ease of operation. The process by which the

software system was designed, implemented and validated is discussed here, along with details as to the methods by which experiments were carried out.

3.4.1 *Modular Decomposition*

The evolutionary cycle, outlined in chapter 2, consists of a number of distinct operations which present the software designer with naturally separate component parts. The inherent division within GAs lends itself particularly well to an object oriented design process, with items such as chromosomes, individuals and populations immediately obvious as potential objects. In addition, the inherent parallelism of GAs assists in defining the boundaries of these objects.

The initial task in undertaking the work was to design a simple, traditional GA. This served to familiarise the author with the C and C++ languages, as well as providing a practical understanding of the GA process. During this activity, ideas for the Multi-GA began to emerge and extensions of the simple GA to a multi-chromosome implementation were undertaken. As with any software system, a number of changes took place during development, for both implementation and research oriented reasons, but the fundamental design concepts changed little. A great deal of research effort was invested in designing a robust, easily modifiable suite of software for use as a research tool.

3.4.2 *Multi-GA object structure*

In this section, a basic familiarity with object-oriented programming is assumed. Interested readers are referred to Holub (1992) and Murray (1993) for an introduction to object oriented design and programming with C++. The basic concept is the definition of software structures as distinct objects, containing member procedures and functions which operate on the objects.

The structural concept outlined in section 3.1 provided a natural break into three core objects – chromosome, individual and population. The genetic operations, applied to objects, were implemented as member functions of the appropriate object class.

Certain additional classes were required for implementation reasons and simplicity of design. The most notable of these is the class 'Individual_pair' containing the member functions for the core genetic operators, crossover and mutation. The genetic process dictates selection of two members from a population (assuming a roulette wheel based method) and the application of

genetic operations to that pair to create children. As such, the genetic process spends a large proportion of its time dealing with just two individuals, hence the creation of a class containing the required genetic function procedures and two individuals.

For ease of development, sections of code requiring frequent modification were grouped together, namely genetic operators and fitness functions. Addition of experimental testing procedures was then simply achieved through the insertion of an appropriately identified section of code in the respective function module, adhering to the software engineering principles of modularity and information hiding. The other major distinct module developed was a data storage module, holding arrays of chromosomes. This was a change forced by C++ language restrictions.

The complete modular structure of the Multi-GA software suite is illustrated in figure 3.5. This shows the major classes utilised in the software implementation, in addition to the location of the main genetic procedures and the newly developed add/delete mechanism. It is clearly visible from the diagram how the natural distinction of the GA processes, along with ease of modification through easy insertion of new operators or functions, has produced a simple modular C++ object structure. The main module acts to oversee the genetic process. The individual consists of a number of chromosome objects, accessed for implementation reasons via the data storage class.

The functions module, containing both chromosome and individual level fitness functions, is accessed from both of those modules at the time of the fitness evaluation. Additional fitness functions are added directly into this module, with minimal effect on other objects in the structure. The same is true of the operators module, called by the pair of selected individuals to perform the required genetic manipulations. Again, for implementation reasons, use of the addition/deletion operators required interaction with the data store class but standard genetic procedures interact only between population and selected individuals.

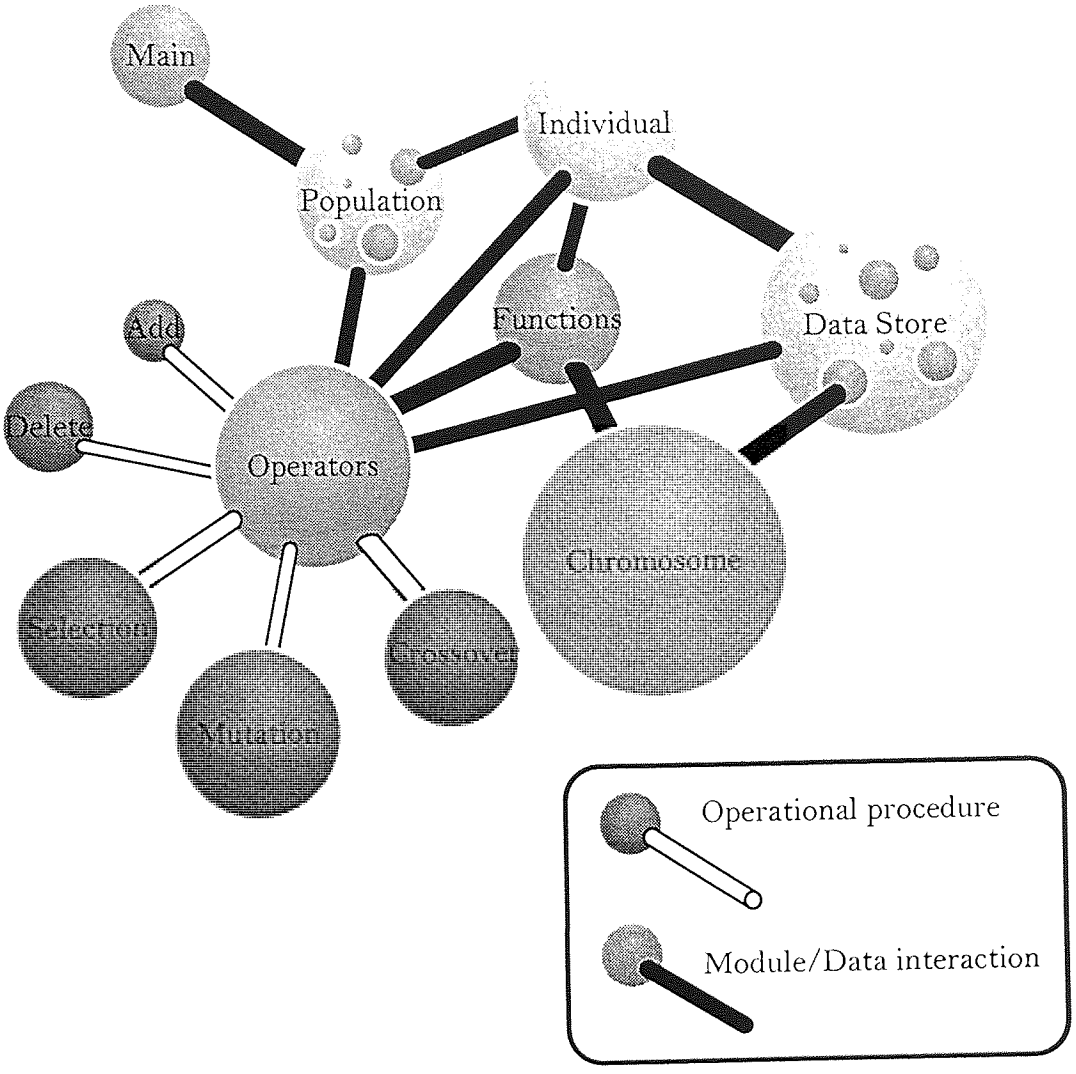


Figure 3.5: Multi-GA software object interactions.

3.4.3 *Templated parameterisation*

One of the key features of the language selected was the availability of templated parameterisation. The templating feature of C++ allows the programmer to define the type of a procedure, function or object at execution time. The ability to pass a type (predefined or user defined) as a parameter into a procedure has proven to be immensely useful, reducing the complexity of the resulting software considerably.

It is not an exaggeration to say that a great deal of the progress made in the implementation and testing of the Multi-GA was made possible by the flexibility of parameterisation given by the use of templated software. For other researchers implementing multi-representational structures, the availability of compilers capable of handling this style of templated parameterisation is an invaluable aid.

The relationship of the Multi-GA structure and the software implementation made possible by the use of templating is illustrated in figure 3.6.

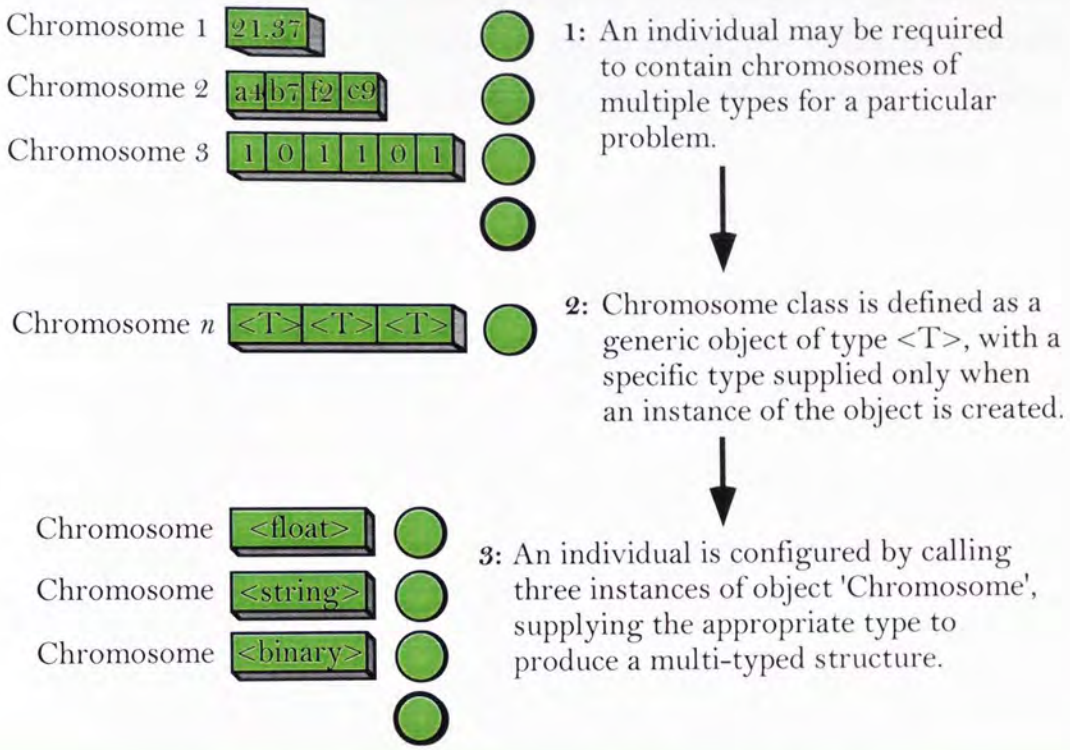


Figure 3.6: Templating applied to the Multi-GA in software.

3.4.4 *Validation and testing*

Having designed a detailed set of class structures, testing was obviously required to ensure the validity and correct execution of the code and the GA processes. The initial programs for a simple GA, not utilising templates, were tested by running the GA on a simple $f(x) = x^2$ problem as outlined in Goldberg (1989). A small population size (5) was utilised and the output of the program examined by hand to ensure the correctness of the genetic operators, and the expected behaviour of the GA cycle. Once satisfied of the correctness of code by manual examination, this simple GA was used as a control study during the preliminary stages of testing the more complex software.

Development of the templated, object oriented Multi-GA took place over a considerable period of time. Once the initial code was completed, tests were run using the same parameter settings as for the simple GA program and the outputs compared to ensure that the results on simple test problems were correct. Again, examination of the code during execution, through use of a graphical, interactive debugger and manual analysis of the program output verified the correctness of the GA's functionality.

This procedure was then repeated as each new routine, operator or function was added to the code. Tracing of the program via the interactive debugger allowed exact interrogation of variables during execution and the trickier bugs were located by this process. By tackling software development in this structured, incremental manner, the problem of major debugging effort was avoided through verification of smaller procedures at the time of their implementation.

Chapter 4: Comparative Testing

4.1 Introduction

A series of tests were performed on the Multi-GA in order to investigate its performance in relation to the traditional GA. A number of different studies were carried out, exploring the features and flexibility given by the Multi-GA. Chapter 6 deals with the application to more complex real-world problems, but a direct comparison to the traditional GA structure on accepted problem surfaces was also necessary.

This chapter describes the test problems applied, the rationale behind the testing, the implementation of the chromosome encoding within the GAs and the results of the fixed rate experiment series carried out. It also describes the move to dynamic parameterisation that arose out of the fixed rate experiments, laying the foundations for both the discussion of dynamic parameterisation and application oriented results presented in chapters 5 and 6 respectively.

4.2 Test problems used

4.2.1 Rationale behind surface selection

In selecting the type of problem to apply, recognised optimisation surfaces of a complex, scalable nature were sought in order to allow experimentation with a number of chromosomes in the Multi-GA structure. Whilst problem surfaces such as the De Jong test suite (1975) are well recognised as benchmark tests, it was felt that these problems lacked the requirement of a multi-chromosome implementation since they could be described by a simple chromosomal representation.

Initial investigations into optimisation surfaces containing multiple local minima led to the identification of surfaces defined in a paper by Styblinski & Tang (1990). Following experimental results and further encouraged by recommendations from Mühlenbein, more complex, widely accepted surfaces were selected – namely Schwefel's F7 and Griewank's F8 function. The choice of these two functions from the set listed in Mühlenbein's (1991) parallel GA study was made because of their contrasting surface shape and Mühlenbein's identification of them as difficult, scalable problems. In particular, Mühlenbein (1991) described F8 as *"one of the most difficult to optimise because it is non-separable ... [and] has to climb a hill to get to the next valley."* Selection of these problems

allowed higher dimensionality tests to be performed, making full use of the multi-chromosome potential which the tests were intended to investigate.

4.2.2 *The problem surfaces*

The relative simplicity of the representational qualities of problems such as the deceptive problems, or the De Jong test suite, led to the decision to steer towards scalable problems, fully utilising multiple chromosomes. The test problems were also selected in order to provide an upwards gradation in difficulty, from the Styblinski & Tang (1990) surfaces through to the more difficult F7 and F8 problems identified by Mühlenbein (1991).

4.2.2.1 *Styblinski & Tang surfaces*

Styblinski & Tang (1990) carried out a study of function optimisation and identified three equations defining problem surfaces of varying difficulty. All three were multimodal, providing complexity at least similar to some functions in the De Jong test suite, but with the added benefit of scalability in the later two functions. Consequently, these problems were the first set tackled, initially in 2 variable form and then scaled to the more complex 10 variable form.

$$f(x_1, x_2) = \frac{1}{2}\sum_{i=1}^{2}\left(x_i^4 - 16x_i^2 + 5x_i\right) \qquad -4 \le x_i \le 4 \qquad [4.1]$$

The first problem presented, defined by equation [4.1] above, provided a scalable, multimodal surface with four local minima, the global minimum at i = -2.903534 (i = 1, 2) and a maximum close to the origin, illustrated in figure 4.1.

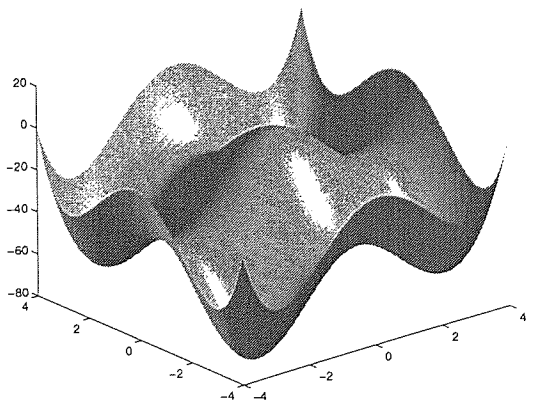


Figure 4.1: Optimisation surface 4.1

In the 2 variable form, this surface was extended by Styblinski & Tang to increase the number of local minima, providing a surface (illustrated in figure

4.2) with 13 small local minima through addition of cosine terms (equation [4.2]).

$$f(x_1, x_2) = \frac{1}{2}\sum_{i=1}^{2}\left(x_i^4 - 16x_i^2 + 5x_i\right) \qquad [4.2]$$

$$- 10\cos\left(4\left(x_1+2.903534\right)\right)\cos\left(2\left(x_2+2.903534\right)\right) \quad -4 \le x_i \le 4.$$

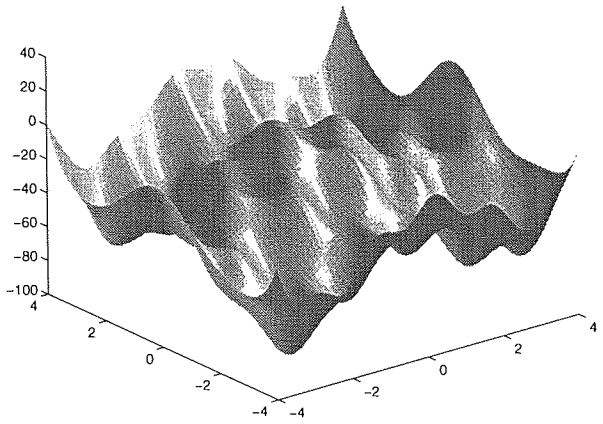


Figure 4.2: Optimisation surface 4.2

A further increase in complexity was obtained by scaling equation [4.1] to 10 variables, increasing the number of local minima to 1024. In addition, Styblinski & Tang presented another surface with the global minimum at an off-centre location, defined by equation [4.3] and illustrated in figure 4.3. Whilst illustrated in 2 variable form, the experiments were performed only on the more complex 10 variable form of this problem.

$$f(x_1, \ldots, x_n) = \left(\frac{1}{2n}\sum_{i=1}^{n} x_i^2 - 4n\prod_{i=1}^{n}\cos(x_i)\right) + 40 \qquad -4 \le x_i \le 4 \qquad [4.3]$$

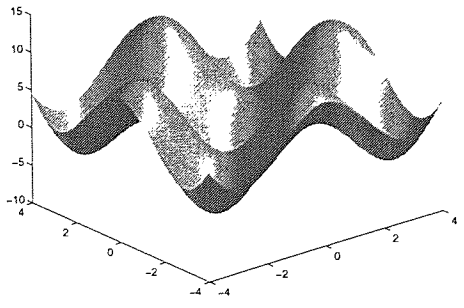


Figure 4.3: Optimisation surface 4.3

These surfaces provided a number of scalable optimisation alternatives, with increasing numbers of local minima and different global minimum locations.

4.2.2.2 Schwefel's F7 function

$$f_7(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{|x_i|}\right) \qquad -500 \le x_i \le 500 \qquad [4.4]$$

The F7 function provides a difficult surface for an optimisation task and is scalable in nature, allowing expansion to an arbitrary number of variables. As illustrated in figure 4.4, the problem surface contains a large number of local minima, with the global minimum located off centre, at $x_i = 420.9687$ for $i = (1, ..., n)$. In addition, the next best optimum value is located far away from this point, potentially leading the GA to convergence at a point from which it may be difficult to recover. Figure 4.4 shows the F7 function for a 2 variable problem, clearly showing the large number of local minima present.

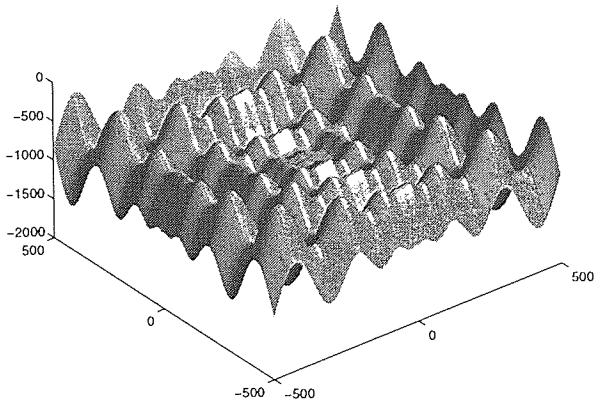


Figure 4.4: Schwefel's F7 function (2 variable)

4.2.2.3 Griewank's F8 function

$$f_8(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \qquad -500 \le x_i \le 500 \qquad [4.5]$$

The F8 function is another example of a scalable problem, applicable in arbitrary dimensions. The function provides a different type of surface from that presented by F7, having an overall basin shape, but on closer examination consisting of a number of hills and valleys moving out from the central global optimum. The behaviour of this surface can clearly be seen in figure 4.5,

illustrating the F8 surface's general shape (left) and close detail near the origin (right).

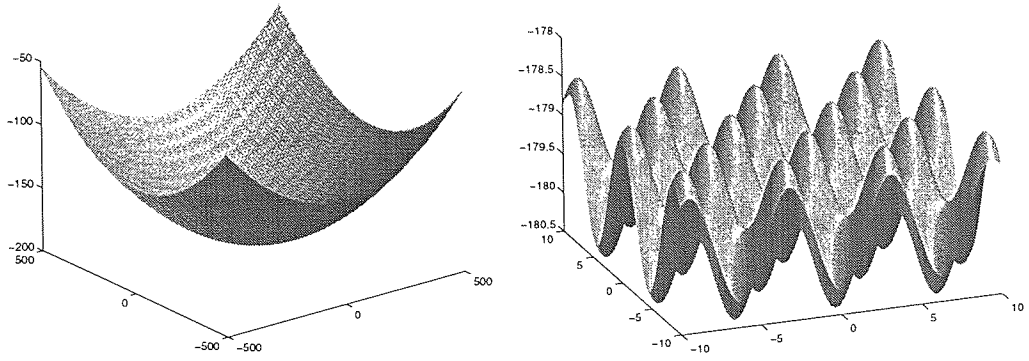


Figure 4.5: Griewank's F8 function (2 variable) general shape (left) and close detail

F8 apparently has the qualities of providing a highly testing surface, specifically identified by Mühlenbein (1991) as extremely difficult to optimise. As with the F7 function, the tests were applied in both 2 variable and 10 variable combinations, giving the GAs increasingly complex surfaces to optimise. Whilst this problem surface looks highly complex, recent research (Whitley *et al.*, 1995) has presented a powerful argument as to the lack of suitability of F8 for GA optimisation. The details of this research, published after the experiments described here were carried out, is discussed in section 5.5.

4.2.3 *Representational handicaps*

Whilst it was necessary to apply the Multi-GA and traditional GA on accepted optimisation problems, it is important to note that the application to these problems did not fully utilise the structural benefits proposed by the Multi-GA. In particular, the concept of independent chromosome fitness functions was not permitted by the representation required to implement benchmarking functions. This feature of the Multi-GA required a more complex application problem, of the type described in chapter 6. This will become clear in the following section, where the encoding method more clearly illustrates the lack of an independent chromosome fitness function. However, significant differences were still made possible by the structure and the comparative testing exploited these to produce a number of interesting results.

4.2.4 *Encoding the test problems*

As with any GA application, a method of encoding the variables into chromosome string form and designing the fitness function had to be found. An acceptable balance between ensuring an accurate comparison and exploiting the

representational differences in the two GAs was required. Following the lead of many other GA applications, each variable was encoded onto the chromosome string in turn, producing a traditional GA consisting of a concatenation of the i variables. The range of the variables, and hence the positional limits along the chromosome string, were incorporated into the appropriate fitness functions in order to ensure accurate decoding of the representation.

In the multi-chromosome structure, the representational flexibility was utilised by encoding each of the i variables as a feature chromosome. However, the representational handicapping discussed earlier introduced difficulties, with the chromosome level fitness function effectively becoming redundant in this type of problem application. The chromosome fitness function represented the decoded value of the feature chromosome string, with the individual level fitness function behaving in the same way as with the traditional GA. The result of this problem-oriented restriction was to produce a Multi-GA structure that implemented separate chromosomes, with the potential for independent rates and operations, but without the full utilisation of chromosome fitness functions or alternative representations in other feature chromosomes.

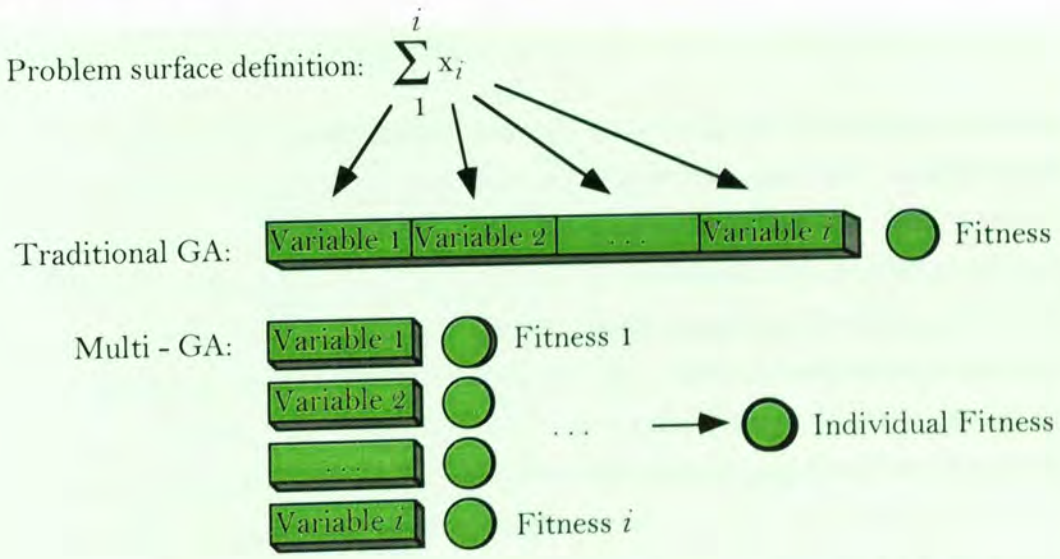


Figure 4.6: Encoding problem surface variables into traditional and Multi-GA structures

As the problems increased in dimensionality, a corresponding increase in the number of concatenated variables or feature chromosomes occurred. This is illustrated in figure 4.6, showing the traditional and Multi-GA methods for representing i variables. The method of representation was the same for each of the optimisation surfaces tested, all being examples of the same style of problem.

An additional change from the problem surface equations given was introduced for operational simplicity. Where the surface implemented contained both positive and negative values, an offset was subtracted from the final fitness evaluation to ensure that the surface was (in the case of minimisation) below the zero axis. For Goldberg (1989) style implementation of fitness proportionate selection, values both above and below the zero axis can mislead the selection process from minimisation to maximisation, leading to location of the global maximum rather than global minimum. This problem is overcome when specifying ranked selection with minimisation or maximisation identified as a specific parameter, as encoded in the Multi-GA software. The offset produced no change in the behaviour of the fitness functions, but simply lowered the final result by an appropriate amount.

4.2.5 *Designing a suite of experiments*

Having identified a series of optimisation surfaces of varying complexity and scalability – and bearing in mind the representational handicaps in traditional optimisation contexts outlined earlier – a series of illustrative, rather than complete and exhaustive tests, was constructed.

4.2.5.1 *Objective of the optimisation tests*

The intention of the optimisation tests was to investigate the effects of multiple chromosomes and analyse how the multi-chromosome GA, conceptually suggesting increased flexibility, performed on a series of problems that the traditional GA handles reasonably effectively. A number of potential areas, such as co-evolution, dynamic parameterisation, representational flexibility etc. have been identified as holding potential for the multi-chromosome structure. However, before advancing to these and multi-dimensional applications problems that may be more directly suited, the obvious question "how does it do against the traditional GA?" had to be addressed.

The test suite was intended to provide this basic comparison of traditional to Multi-GA performance on an increasingly complex series of problem surfaces. Having performed these initial experiments, a number of the more advanced areas offering greater potential benefit to the Multi-GA were explored. However, the time available regrettably did not allow for a full and exhaustive series of tests of all aspects of comparative testing and the test series was not designed as such. The intention was to investigate the performance with a view to moving forwards to practical, multi-dimensional problems with greater feature independence following an initial behavioural comparison.

4.2.5.2 *The experiment series implemented*

The test series moved from simple to more complex surfaces, providing discrimination between the traditional and Multi-GA by exploiting the areas of the Multi-GA expected to prove beneficial. In doing so, the design of the experiment suite was a dynamic process, with the results of early experiments indicating the way forward for the later tests carried out.

Initial tests were conducted on the Styblinski & Tang (1990) surfaces described in section 4.2.2.1. Starting with 2 variable versions, then progressing to 10 variables, the intention was to analyse the effects of increasing the representation to 10 chromosomes within the Multi-GA structure and demonstrate the effects on performance (if any) of distinct individual and chromosome level operations.

The first experiments were performed with fixed parameters on Styblinski & Tang problem surfaces [4.1] and [4.2] in 2 variable form. The move to higher dimensionality was carried out with surface [4.1] and surface [4.3] in 10 variable form. Typical parameter settings were used initially and the results of these experiments dictated the move on to the more complex surfaces of F7 and F8, recommended by Mühlenbein (1991) in both 2 and 10 variable forms.

The general aim of the experiments on F7 and F8 was the same as above – to illustrate performance on a simple surface, then repeat for a higher dimensionality to increase the effect of the multi-chromosome structure. Following the results of early experiments using fixed rates, later experiments considered dynamic parameterisation and alternative alphabetic representations. Experiments in these areas were designed to explore the effects of dynamic application of both crossover and mutation rates and ranking alternatives to increase maintenance of genetic diversity. The move to alphabetic alternatives then explored the effects (if any) of utilising a higher cardinality alphabetic representation, in conjunction with the increased structural flexibility, and the development of a new real-encoded crossover mechanism – Quotient crossover.

As the discussion shows, the experiments performed were by no means an exhaustive comparison, but provided a base for indicative analysis of the structural potential in both binary and non-binary alphabetic representations for a variety of increasingly complex problem surfaces. The remainder of this chapter presents the results of the fixed parameter rate experiments.

4.3 Experimental results

The first series of comparative tests were performed using fixed parameter settings, initially on the surfaces defined by equations 4.1 - 4.3, then on the more complex F7 and F8 problems. These tests provided the preliminary indications of comparative performance of the traditional and Multi-GAs, over increasingly complex surfaces.

4.3.1 GA configuration

The initial series of experiments on surfaces 4.1 - 4.3 were performed with typical GA parameter settings – population size 100, 75% crossover, 4% mutation and ranked roulette wheel selection using a generational GA, with elitism carrying across the single best member at each generation. The Multi-GA used the same parameter settings, with the addition of Individual level crossover operating also at 75%. Each GA trial was run for 500 generations, initial studies indicating that this gave adequate opportunity for population convergence, whilst not demanding excessive CPU time. Every experiment set consisted of 20 GA trials, run with a different random seed to provide a measure of statistical significance.

The chromosomes were encoded in binary form, with varying ranges for the simpler and more complex surfaces. Surfaces 4.1 - 4.3 were encoded from -4 to +4, using 12 binary bits after the decimal point to provide a significant search space. Offsets applied in order to ensure negative fitness evaluations were -12 for surfaces 4.1 & 4.2 and -80 for surface 4.3. F7 and F8 were encoded from -500 to +500, with 4 bits after the decimal point and offsets of -840 and -180.5 respectively.

Tables presented all take a similar format, showing a number of statistics about GA performance. The tables detail the type of problem tested (including parameterisation information where pertinent) and show results, under the column headings shown overleaf.

Average Best/Worst	Best/worst results of each GA trial, averaged over the 20 trials performed.
Std. Dev.	Standard deviation of the average listed in the preceding column.
Best/Worst Ever	Best/worst single result obtained by any of the 20 runs.
Average Best Gen.	Number of generations taken to find the best result of each trial, averaged over the 20 trials performed.
Degree of Elitism	Number of individuals carried forward without modification to the subsequent generation by elitism.

4.3.2 2 variables, surface 4.1

The table below presents the results of the 20 GA trials for both traditional (TGA) and Multi-GA (MGA) on surface 4.1.

GA Tested	*Average Best*	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*Std. Dev.*
TGA	-78.29	0.05	2.18	3.73	9.37	-78.33	57.50	65.92
MGA	-78.32	0.05	5.27	4.30	14.13	-78.33	40.95	23.98

Table 4.1: Fixed rate Traditional and Multi-GAs applied to 2 variable surface 4.1

The results indicated interesting trends, showing comparable performance from both the traditional and Multi-GA, as shown by the average best results achieved. Both GAs found the global optimum (-78.33) during the 20 runs, indicated by the best ever result. However, a number of differences are noticeable between the two GA methods, relating to worst performance and generations to converge.

Whilst indicating similar average best performance between the two GAs, the Multi-GA method converged, on average, more quickly than the traditional GA. Whilst the result is not statistically significant, being within the standard deviation of the traditional GA, it does seem to indicate a quicker convergence from the Multi-GA with a comparably smaller standard deviation.

Examination of the average worst result also shows a difference. Analysis of the worst results can provide information about the population behaviour of the GA. One of the primary objectives of GA researchers has been to maintain genetic diversity, thereby restraining premature convergence. A measure of population diversity can be gained by analysis of the range of fitnesses present within a

population, indicated by the upper and lower fitness bounds. As such, the average worst result in a population, taken in conjunction with the average best result, assists with indicating the level of fitness diversity (and hence convergence) within the population.

The results shown by the average worst figures indicated a slightly higher value from the Multi-GA, although again just within the traditional GA standard deviation range. This greater range suggests (as described) improved fitness diversity from the Multi-GA.

However, the results from both GAs are within the standard deviation limits of one another, so it would be unwise to conclude anything other than on average, comparable but slightly improved convergence and population diversity is seen from the Multi-GA approach, with no significant disadvantages seen by utilising the more structured Multi-GA approach.

4.3.3 *2 variables, surface 4.2*

The indications given by the results obtained from surface 4.1 led to identical experiments being carried out on a more complex surface – the extension to 13 local minima provided by surface 4.2.

GA Tested	*Average Best*	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*Std. Dev.*
TGA	-100.21	0.27	-4.82	4.15	5.69	-100.33	79.75	81.07
MGA	-100.31	0.05	-3.76	4.59	5.35	-100.33	54.30	64.02

Table 4.2: Fixed rate Traditional and Multi-GAs applied to 2 variable surface 4.2

The results shown in table 4.2 are similar in trend to those presented in table 4.1. Both GAs find the global optimum (now -100.33) on at least one of the 20 trials carried out. On average, the Multi-GA performs marginally better than the traditional GA, but again well within the bounds of standard deviation, indicating comparable performance between the two. The range of standard deviation on both average best and generations to converge is smaller with the Multi-GA, indicating a more consistent performance. In general, the results indicated by the test on the simpler problem surface were repeated here, indicating comparable performance, converging in slightly less time with a smaller standard deviation in the Multi-GA.

4.3.4 10 variables, surface 4.1

The Multi-GA concept outlined in chapter 3 suggested the use of multiple chromosomes, with individual level crossover allowing exchange of entire feature chromosomes. In the 2 variable problem surfaces, there is little scope for this operation to take place. In addition, both GAs found the global optimum, indicating a relatively simple problem surface. As a result, the extension of surface 4.1 to 10 variables, utilising its scalable nature, was implemented to provide a more demanding test surface containing 1024 local minima. The results are presented in table 4.3.

GA Tested	*Average Best*	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*Std. Dev.*
TGA	-77.92	0.72	-13.23	2.99	-7.88	-78.29	396.45	54.95
MGA	-78.25	0.00	-13.83	2.03	-8.45	-78.33	298.95	65.83

Table 4.3: Fixed rate Traditional and Multi-GAs applied to 10 variable surface 4.1

The results again indicate similar trends, with the Multi-GA performing on average slightly better than the traditional GA. Whilst the Multi-GA average best was still within the bounds of traditional GA standard deviation, it was noted that the traditional GA did not actually find the exact global minimum in this example, falling fractionally short of the -78.33 required. The apparently zero standard deviation reflects a smaller standard deviation than can be expressed to 2 decimal places.

Again, convergence was achieved more quickly with the Multi-GA, with a larger difference than that shown in previous experiments. These results showed an increase in the trends indicated by the 2 variable results, with the Multi-GA exhibiting slightly better performance than the traditional GA on this surface, measured in terms of fitness convergence to the global optimum, with comparable average performance and comparable, if slightly improved, average convergence time. In general, the trends hinted at with the 2 variable version were confirmed by the more complex 10 variable surface.

4.3.5 10 variables, surface 4.3

This surface presents a more complex challenge still, with Styblinksi & Tang (1990) describing it as having *"a large (unknown) number of local minima, whose 'deepness' and 'frequency' depend on a specific choice of the coefficients of the sinusoidal part."* The results of experiments on this surface are presented in table 4.4.

GA Tested	*Average Best*	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*Std. Dev.*
TGA	-119.24	0.51	-45.29	4.25	-40.02	-120.00	374.00	52.85
MGA	-119.16	0.50	-41.19	1.33	-40.01	-119.97	298.60	52.75

Table 4.4: Fixed rate Traditional and Multi-GAs applied to 10 variable surface 4.3

The results for this surface again indicated comparable performance between the two GAs. The Multi-GA converged a little faster on average, but fractionally missed the global minimum, found by the traditional GA at -120.00. No indications were given by this set of results of a continuation in the trend indicated by the previous surfaces, although the results provided no conclusive evidence of a loss of performance either. The conclusions on this more complex surface were comparable performance, well within the bounds of standard deviation for both GAs.

The trend indicated by the first three experimental sets and the inconclusive results of the third led to further experiments on more complex surfaces. Performance was comparable on surfaces 4.1 - 4.3 and in all cases both GAs came close to, or found, the global minimum. Consequently a similar set of tests, initially in 2 variable form then expanding to 10 variable form, were carried out on the more difficult F7 and F8 surfaces.

4.3.6 *2 variables, F7*

As with the experiments carried out so far, 20 GA trials were performed under the same parameter settings as previously used. In addition, experiments were performed to investigate the effects of increased selection pressure on the search process. Population convergence, controlled by the selection algorithm and the use of elitism, encapsulates the issues defined by the exploration/exploitation dilemma. Whilst detailed studies on a variety of selection schemes have been carried out elsewhere (e.g. Goldberg, 1990; Whitley, 1989), it was considered instructive to investigate the effects of selection pressure through elitism on the traditional and Multi-GA architectures.

Investigations of varying degrees of elitism were carried out with the best and best 10 members of a 100 member population being transferred to the next generation. Table 4.5 presents results obtained for the 2 variable F7 surface.

GA Tested	*Degree of Elitism*	*Average Best*	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*Std. Dev.*
TGA	1	-1677.79	0.25	-123.96	82.29	-2.04	-1677.97	61.60	38.46
MGA	1	-1677.58	0.93	-87.98	79.56	-2.04	-1677.97	30.95	5.72
TGA	10	-1676.98	1.70	-130.68	77.58	-21.70	-1677.97	347.40	60.35
MGA	10	-1677.46	0.77	-76.38	73.58	-2.04	-1677.97	49.75	103.57

Table 4.5: Fixed rate Traditional and Multi-GAs applied to 2 variable surface F7

The results showed similar trends to the experiments performed so far, with performance of traditional and Multi-GAs being comparable. Although the traditional GA showed marginally improved performance on average, the results were well within the Multi-GA standard deviations. Performance measured in terms of the best ever result was identical, with both GAs finding the global optimum of -1677.97 in all cases. The differences between the two GAs occurred in the areas of population diversity and convergence time. The Multi-GA average worst values indicated a greater spread of fitness ranges (and hence genetic diversity) than that given by the traditional GA, although again within standard deviation limits. However, convergence times did present a more substantial difference, indicating Multi-GA convergence to be, on average, 50% that of the traditional GA with a much smaller standard deviation.

The differences indicated by the use of a higher degree of elitism were minimal, showing a slight degradation of average performance and an increase in convergence times. The traditional GA was most seriously affected by the move to 10 member elitism, with a huge increase in convergence time from 61.6 generations (±38.46) to 347.4 generations (±60.35).

Overall, the results indicated that the problem surface itself is not sufficiently complex to discriminate between the performance of the traditional and Multi-GA. The effect of 10 member elitism did not overly influence performance, but appeared to play a part in the time taken for the GA to reach its final result.

4.3.7 *2 variables, F8*

In addition to F7, Mühlenbein (1991) identified F8 as a particularly testing problem surface, illustrated by the complex shape of this surface as shown in figure 4.5 earlier. Results for the experiments performed on F8 are given in table 4.6 overleaf.

GA Tested	*Degree of Elitism*	*Average Best*	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*Std. Dev.*
TGA	1	-180.46	0.10	-68.48	5.70	-58.38	-180.49	58.00	100.57
MGA	1	-180.48	0.12	-68.03	6.02	-58.72	-180.50	56.65	90.16
TGA	10	-180.40	0.09	-71.19	7.94	-57.22	-180.50	31.35	20.38
MGA	10	-180.46	0.00	-67.60	5.82	-56.76	-180.50	30.40	35.19

Table 4.6: Fixed rate Traditional and Multi-GAs applied to 2 variable surface F8

The results were strikingly similar to those presented previously, showing comparable performance between the two GAs. In terms of both best ever performance and average best, there was little difference between the two – the Multi-GA performed slightly better and found the global optimum of -180.50. Interestingly, convergence times were very similar for both 1 and 10 member elitism results, with 10 member elitism having little effect other than a marginal lowering of the average best performance.

4.3.8 *10 variables, F7*

Despite the more complex nature of the F7 and F8 surfaces, 2 variable experiments still resulted in the location of the global optimum by both GAs. For the same reasons as outlined earlier – namely greater surface discrimination and greater utilisation of the multi-chromosome structure – the more complex surface experiments were repeated at higher dimensionality. The results of the scaling of F7 to 10 variables are given in table 4.7 with the global optimum now located at -5029.83.

GA Tested	*Degree of Elitism*	*Average Best*	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*Std. Dev.*
TGA	1	-4640.83	139.57	689.84	195.24	1046.63	-4901.57	347.40	60.35
MGA	1	-4718.59	172.31	710.56	152.24	1146.35	-5020.01	275.55	57.41
TGA	10	-4710.48	142.74	645.95	138.47	849.83	-4932.30	310.90	54.31
MGA	10	-4685.29	170.79	725.34	185.31	1213.99	-5016.82	237.65	81.90

Table 4.7: Fixed rate Traditional and Multi-GAs applied to 10 variable surface F7

The results reflected the more discriminating nature of the surface, with both GAs failing to find the global optimum at -5029.83. The Multi-GA showed comparable performance in terms of best ever fitness and average best performance, again coming within standard deviation limits. The increased discrimination of the problem surface was reflected by the larger standard deviations in both average best and worst values. The Multi-GA also exhibited a higher value for average worst performance, indicating a higher diversity

range within the population, although again these values were in range of the traditional GA's standard deviation.

10 member elitism showed a degradation in the performance of the Multi-GA on average and marginally in best ever performance. Convergence time was lower with the Multi-GA, although not statistically significant.

As with previous experiments, the results indicated comparable performance between the two GAs, with a tendency for the Multi-GA to take the edge slightly over the traditional GA, shown by a higher average best, best ever and convergence time in most cases.

4.3.9 *10 variables, F8*

The experiments were repeated with F8 in 10 variable form, completing the series of fixed rate tests as shown in table 4.8.

GA Tested	*Degree of Elitism*	*Average Best*	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*Std. Dev.*
TGA	1	-180.45	0.00	-117.98	1.00	-116.99	-180.50	202.25	48.22
MGA	1	-180.47	0.10	-117.70	0.77	-116.98	-180.50	360.15	81.14
TGA	10	-180.45	0.08	-118.03	0.93	-117.02	-180.49	186.20	37.66
MGA	10	-180.47	0.05	-118.14	1.18	-116.98	-180.50	202.45	65.56

Table 4.8: Fixed rate Traditional and Multi-GAs applied to 10 variable surface F8

Comparable performance was again indicated from the two GAs. Both GAs located the global optimum of -180.50 with similar population diversity and average best performance within standard deviation values of one another. Convergence times presented interesting results, in contrast to previous experiments, showing the Multi-GA taking longer to converge than the traditional GA. The discrimination indicated by the escalation to 10 variables with F7 did not appear to be mirrored by the F8 surface.

4.4 A summary of fixed parameter testing

A number of tests were performed on a variety of increasingly complex problem surfaces, comparing the performance of traditional and Multi-GA structures. Experiments were performed initially with the simpler surfaces 4.1 - 4.3, increasingly the complexity from 2 to 10 variables in the scalable surfaces 4.1 and 4.3. The results showed good performance from both GAs, in most cases locating the global minimum. Indications of a trend towards marginally better performance, convergence in fewer generations and a wider population fitness

diversity were suggested, although the range of standard deviation results over the 20 trials left no statistically significant conclusions.

Experiments were then carried out on the more complex F7 and F8 surfaces in both 2 variable and 10 variable form, with similar results to surfaces 4.1 - 4.3. Investigation of population convergence through increased degrees of elitism resulted in surface dependent behaviour, with no clear indication given of a general benefit when applied in the manner described here.

What the results did show was that, despite the representational handicap restricting full utilisation of the Multi-GA in the traditional GA problem environment, comparable performance was seen from the two GAs over a range of increasingly complicated problem surfaces of widely differing characteristics. Although no clear lead was shown by the Multi-GA, indications were seen of improvements in some areas of performance and the more structured representation certainly did not produce a degradation of performance in the traditional GA environment.

These results led to further investigation of the Multi-GA, still in a traditional GA context with associated representational restrictions, but in an attempt to further exploit the structural flexibility in a beneficial manner through application to dynamic parameterisation.

Chapter 5: Dynamic Parameterisation

5.1 Exploring dynamic parameterisation

The field of dynamic parameterisation, introduced in section 2.3.4.4, has been an area of ongoing research in GAs for some time. The structural flexibility of the Multi-GA, laid out in chapter 3, contains a great deal of potential for a number of current techniques to be explored, including dynamic parameterisation. The Multi-GA's independent feature chromosome parameter rates lend themselves to dynamic control, allowing independent assignment of the most appropriate parameter rate to each chromosome. It is this line of investigation that was taken up during the experiments described here.

5.1.1 Related research

As discussed briefly in chapter 2, a number of authors have undertaken research into dynamic parameterisation, exploring several different facets of this approach. Of this research, a number of papers exploring the dynamics and application of rate calculation are particularly appropriate to the Multi-GA studies carried out here. Srinivas and Patnaik (1994) proposed the use of an Adaptive Genetic Algorithm (AGA) that dynamically controls both the crossover and mutation rates of individuals, according to their relationship with the current state of the population fitness. As we shall see, the mechanism of rate calculation used by the Multi-GA is similar, making use of and confirming a number of important points made by Srinivas and Patnaik.

An investigation into adaptive operator rates was carried out by Davis (1989), who made the observation that different operators are required at different stages of the genetic process, with the GA able to adapt and select as appropriate. This theme has been followed up by a number of other authors, referenced throughout this section. In their investigation into adaptive crossover, White & Oppacher (1994) followed this theme by proposing an automaton controlling crossover rate, using a version of the Adaptive Uniform Crossover operator (AUX) and inserted into the genetic representation itself, thus being subjected to the genetic process. This theme, similar to the concepts of inserted operators or genetic material discussed by Schaffer & Morishima (1987) and Levenick (1995), has been followed up by several authors as a mechanism for self-adaptation. In his study of adaptive mutation, Bäck (1992) encodes the mutation rate directly into the genetic string in a similar way to White & Oppacher (1994). This is of particular relevance to the Multi-GA, as

the later discussion (chapter 7) into self adaptation and the use of multiple chromosomes demonstrates. The use of self inclusion of operator rates for rate evolution, mentioned by Davis (1991) and followed up in the other studies discussed, has seen interesting results in application to both dynamic crossover and mutation, which have relevance to the Multi-GA studies carried out here. In particular, a number of observations made by authors such as Bäck (1992) and White & Oppacher (1994) are observed to some degree in the Multi-GA results that follow.

Starkweather *et al.* (1990) carried out a study into dynamic mutation, similar to (and directly referenced by) that of Srinivas and Patnaik, exploring the use of an adaptive mutation rate in a distributed genetic algorithm. This work, again providing important lessons for the structural development seen in the Multi-GA, takes a significant step in the application of the dynamic parameterisation mechanism to the increasingly popular co-evolutionary concept. Other studies in dynamic mutation, such as that of Hesser and Männer (1992) confirm the results of other studies (such as Schaffer *et al.* (1989), Bäck (1992) and Fogarty (1989)) in demonstrating the relationships between parameter rates, population size and convergence. In a wider context, other authors such as Julstrom (1995) have taken the approach of calculating dynamic operator rates by referring to the past history of individuals, demonstrating a cross-fertilisation of ancestral history mechanisms of the type seen in papers by Eshelman & Schaffer (1991) and Craighurst & Martin (1995).

The study by Starkweather *et al.* (1992) is particularly interesting, mentioning a number of points which are taken up in the Multi-GA exploration of dynamic rate application. Their paper examines the use of a distributed genetic algorithm, with communicating subpopulations exchanging members with an adaptive mutation rate. Of particular significance is the expectation of similar results from GA approaches evaluating individuals in parallel, with mechanisms for maintaining 'locality' of mating. Given the discussion in chapter 7 relating the Multi-GA to a subpopulationary context, their experiments (analogous to a future development of a Multi-GA implementing subpopulations, with individual level crossover maintaining locality of mating) are of particular relevance and may provide a useful comparison for such a future study. This is also the case when examining the results and conclusions of Starkweather *et al.*'s study, in which a number of observed trends are similarly observed in the Multi-GA experiments conducted here. Further discussion of these relationships is examined in the dynamic mutation results section.

Fogarty's (1989) paper provides interesting results for Multi-GA experimentation, in exploring a number of alternative mutation rate calculation mechanisms related to generation number, bit string representation, initial population seeding and a combination of all three. Fogarty's examination of a decreasing mutation rate over the bit string of an individual presents an immediate similarity (as with Harvey's (1992) SAGA structure, discussed earlier) to the structural application of mutation rates in the Multi-GA context. The experiments carried out here did not include studies into the relationship between initial population seeding or generation number and the effects on chromosome level dynamic mutation rates. However, Fogarty's (1989) results for dynamic mutation rates varied by bit position would undoubtedly provide an indication of expected performance with the Multi-GA representing, what is in many respects, an expanded version of Fogarty's bit string to mutation rate relationship.

5.1.2 *Objectives of dynamic application to the Multi-GA*

As outlined in the brief introduction to this chapter, the structure of the Multi-GA as defined in chapter 3 opens up the possibility for independent application of genetic operator rates at the chromosome and individual levels. With the potential for customisation so evident, the success of other authors in dynamic application of parameter rates to the genetic process becomes significant. Given a Multi-GA structure with the potential for a local chromosome operator and rate, it became evident that an investigation into the possibility of dynamic configuration of those rates would be appropriate.

The results of the initial fixed parameter tests indicated that surfaces 4.1 - 4.3 were easily solved in 2 and 10 variable form, by both traditional and Multi-GAs. Given the time constraints placed upon the work, it was decided to restrict subsequent experiments to the 2 and 10 variable F7 and F8 surfaces, since these offered greater discrimination in performance. In addition, this point is mentioned by Srinivas and Patnaik (1994), who observed improved performance of their AGA on problems of a greater multimodal nature. The 2 variable versions of the surfaces provided simple examples of performance and it was recognised that, on the basis of the results presented in section 4.3, surfaces 4.1 - 4.3 would behave in a similar manner to the 2 variable F7 and F8 surfaces.

In addition, the earlier results indicated comparable performance between the traditional and Multi-GA. Investigations into dynamic parameterisation were designed to explore the structural potential of the Multi-GA, demonstrating

alternative methods of dynamic parameterisation within that structure; rather than performing a distinct comparison with a dynamically parameterised traditional GA. Whilst the structure of the Multi-GA differs from that tested in Starkweather *et al.* (1992), a number of similarities exist between the two structures and Starkweather points out that structures providing dynamic rate calculation in parallel *"could display behaviour similar to the implementation described"* if local evaluation is maintained.

Although a direct comparison to a dynamically parameterised traditional GA would be a desirable and useful study to undertake, the objective here was to illustrate that dynamic parameter rate application within the Multi-GA context is feasible and worthy of future investigation, as suggested by other researcher's studies. It is accepted that a traditional GA operating with dynamic parameterisation methods may well give comparable performance, but the requirement to undertake the application work described in chapter 6 meant that time and CPU constraints did not allow for such a study to take place here.

What the results do give is an indication of some alternative methods of applying dynamic rates within the Multi-GA structure and the behaviour of such application. They are not an exhaustive test of all aspects of dynamic parameterisation, but indicative results showing successful application and the potential for futher investigation. Importantly, having demonstrated a mechanism for their application, a foundation is laid for future investigation of their interaction with the dynamic population control explored in chapter 6 through the use of addition and deletion. Chapter 6 undertakes some study of the effects on population behaviour of this new mechanism, but time did not permit a study of the interaction with dynamic rates. This is of particular interest given the observation of Srinivas and Patnaik (1994) that *"a similar dynamic model for varying the population size in relation to the fitnesses of the population"* warrants investigation.

5.1.3 *Mechanisms for diversity maintenance*

In approaching dynamic parameterisation, it was noted that one of the main objectives of this type of work has been to find a set of rates that use suitable diversity maintenance to best address the exploration/exploitation dilemma. Methods of effective diversity maintenance are, as such, an important part of the parameterisation debate from direct control studies (e.g. Julstrom, 1995) through to alternatives like incest prevention (Eshelman & Schaffer, 1991) and ancestral history (Craighurst & Martin, 1995). The latter studies showed the

potential for diversity maintenance by reference to the surrounding population members and this style of approach was followed in the experiments performed here. However, the other studies mentioned earlier also take a less direct approach, attempting to maintain diversity as a function of the rates being dynamically applied. It is this approach that was adopted here, although more direct maintenance of an individual's history is made possible by the Multi-GA structure (discussed in chapter 7).

Methods of dynamic rate calculation were devised that took into account both the Multi-GA structure and the performance of other population members. The obvious candidates for dynamic parameterisation were crossover and mutation rates. In the Multi-GA structure, dynamic crossover was applied at the chromosome level only, with other studies being carried out to show the potential benefits of individual level crossover. In order to retain the relationship between the dynamic rate and performance of other chromosomes, a comparative rate calculation method was used. This method calculated an appropriate rate, related to the performance of the current chromosome with respect to the rest of the population, resulting in the following relationship:

$$\text{Rate} \quad \alpha \quad \text{Fitness difference between current and best chromosomes.}$$

The above method results in an increasing rate, up to a maximum specified by the user at configuration, as the difference between the two chromosomes becomes wider. Application of a rate directly proportional to the fitness distance (or ranked position in the population) produces a higher level of genetic exchange as the fitness of the current chromosome decreases. Consequently, there is a greater exchange of the less fit material contained in members of lower fitness, whilst the disruption of building blocks in the fitter members is minimised by a reduced rate of genetic manipulation.

However, it could also be surmised that a reduction in the rate of exchange may lead to less incorporation of fitter schemata in the genetic process and perhaps induce premature convergence at a local minimum. Also, the better individuals may contain some sub-optimal chromosomes (as individuals are selected on the basis of a combination of chromosome fitnesses), so a peer chromosome operation within a fit individual may not necessarily involve exchange of a fit chromosome. Consequently, an additional metric was developed with calculation of a rate *inversely* proportional to the chromosome fitness distance – that is, the fitter chromosomes are most likely to engage in crossover:

$$\text{Rate} \quad \alpha \quad \frac{1}{\text{Fitness difference between current and best chromosomes.}}$$

Inverse fitness distance calculation acts in the opposite way to direct fitness distance, applying a higher crossover rate to the fitter members, thereby encouraging propagation of their genetic material through the population. The effect of any disruption of good material is lessened through the application of elitist selection strategies, maintaining the best (n) individual(s).

Application of dynamic mutation rates presented further potential for alternative rate calculations. In addition to a calculation based upon chromosome performance, it is possible to base mutation rate calculation on the relative performance of the *individual* within the population. In crossover experiments, the presence of individual level crossover provided exchange at an individual level. Mutation is applied only at the chromosome level, so no structural relationship exists to the individual performance as a whole, hence the methods used for dynamic crossover were reformulated to include references to the individual's rank. Both direct and inverse methods were again applied to this metric.

Specifically, the rank based methods developed were:

(i) $\text{Rate} \quad \alpha \quad$ Pop. ranking of the individual containing current chromosome.

(ii) $\text{Rate} \quad \alpha \quad \dfrac{1}{\text{Pop. ranking of the individual containing current chromosome.}}$

The result was a set of dynamic calculation methods that investigated approaches linked to both exploitation and exploration, based on relative chromosome fitness and performance within the population as a whole.

A number of similarities can be seen in the methods used here when compared with those of other authors. Starkweather *et al* (1992) used the same principle of applying a dynamic rate calculated between zero and n, where n is an appropriate ceiling value selected by the user. Their approach related the application of this maximum rate to the percentage difference in the bits of the two parent strings, rather than directly to fitness values. The relationship to fitness values was taken up by Srinivas and Patnaik (1994), who used a very similar mechanism to that proposed here. The principal difference is that their mechanism related the rate to be applied back to the average fitness of the population. In addition to this, they applied rates in a more liberal manner, forcing a mutation and crossover at high rates on all members that fell below

the 50% mark in the population. In not maintaining a relationship to the average fitness, the Multi-GA relates the current rate value directly to the current best in the population and does not impose any particular rate setting on individuals within the population if they are below a certain threshold. In taking this approach, the Multi-GA more closely follows the evolutionary metaphor, allowing the fitness relationship of the individual to evolve the mutation rate rather than using an interventionist threshold. Given the more complex multi-dimensional structure of the Multi-GA, a purely evolutionary approach requiring minimal pre-configuration is desirable as it allows more flexible application of the structure to an unknown environment, evolving its own parameters to suit that environment. Studies such as White & Oppacher (1994) also demonstrate that, given time, such an evolutionary approach does produce positive results.

In addition, the Multi-GA operator rate application was designed to promote exchange of schemata in individuals relating to their performance. In many GA experiments, it can be noted that worst performing members of the population do contain good schemata. By application of a mechanism inversely relating rate of exchange to performance, the objective of achieving greater exchange in members with low fitness (to whom disruption makes little difference in performance terms), whilst maintaining the currently good solutions with minimal disruption is achieved. This mechanism should allow for greater redistribution of schemata in the lower performing members of the population - an effect not so dramatically achieved by a levelling out with relationship to average fitness values.

5.1.2 *Experiment suite tested*

In order to provide a comprehensive test set, each dynamic rate was applied in turn, producing a series of GA trials carried out on F7 and F8 in 2 and 10 variable forms. As with previous experiments, 20 GA trials were performed for each experiment, introducing statistical significance to the results. Dynamic rates were implemented by allowing the rate to float, calculated by the relevant method, up to a maximum figure supplied at the time of GA configuration. The other parameters regarding population size etc. were unchanged from earlier experiments, namely 100 members running for 500 generations, following indicative experiments that revealed no significant advantage by execution for a greater length of time.

The effects of elitism on dynamic parameterisation were investigated, with experiments applying three rates – 0, 1 and 10 member elitism. Individual level crossover was also investigated, applied in the dynamic crossover experiments at fixed rates of 0% and 75% respectively.

The dynamic mutation series was carried out in a similar manner, fixing the crossover rates at individual and chromosome level and allowing the mutation rates to float up to their maximum values of 10%, 30% and 50% according to each of the four methods devised. The conclusions of the dynamic crossover experiments also played a part, slimming down the initial wide-ranging test series. More details on the dynamic mutation tests is given in section 5.3.

5.2 Dynamic Crossover

The first series of experiments investigated the effects of dynamic parameterisation of chromosome level crossover and the effect of individual level crossover. Experiments were performed on each problem surface with a fixed mutation rate of 4% across the entire individual (that is, a 4% chance of mutation of a chromosomal bit somewhere in the individual, not a 4% chance of mutation within each feature chromosome) and individual level crossover set at 75%, then disabled (i.e. 0%). For each set of experiments, the chromosome crossover rate was calculated according to the direct and inverse chromosome fitness distance methods, up to maximum values of 60%, 75% and 90%. Elitism was applied at all three settings of 0, 1 and 10 members for each experiment.

The experiments were performed first on the 2 variable F7 and F8 surfaces, then moving on to the more discriminating 10 variable cases. The main objectives of the dynamic crossover test series were to:

- Identify the effect of individual level crossover.
- Assess the effects of dynamic chromosome crossover at different rates.
- Assess the effects of elitism at increasing dynamic crossover rates.
- Assess the usefulness of applying dynamic rates in direct and inverse proportion to relative fitness performance.

5.2.1 2 variable, F7

Experiments were performed for each of the dynamic crossover rates and elitism settings, for both 75% and 0% individual level crossover, utilising direct and inverse fitness distance calculation methods. The results are presented in that order, starting with 75% individual level crossover and direct fitness distance calculation (complete results in appendix C, table C.1).

The literature suggests that the use of elitism is a beneficial method of retaining good genetic material from one generation to another, generally leading to improved performance and faster convergence to an area of interest. The results shown in table 5.1 confirmed this, indicating that performance with single member elitism was indeed better than without elitism. In all cases, the average best showed a slight improvement through the use of elitism, although well within the bounds of standard deviation. A further increase to 10 member elitism was not conclusive though.

Chromsome Xover rate	*Degree of Elitism*	*Average Best*	*St. Dev.*
60%	0	-1676.01	1.92
	1	-1676.86	1.19
	10	-1676.94	1.29
75%	0	-1676.61	1.81
	1	-1676.96	1.38
	10	-1677.19	3.74
90%	0	-1676.81	1.94
	1	-1677.21	1.28
	10	-1677.52	0.22

Table 5.1: 2 variable F7 dynamic crossover average best results for direct fitness distance calculation with 75% individual crossover.

Comparison of the differing rates applied showed little discrimination, with a mild improvement as the dynamic rate climbed towards 90%, but well within standard deviation parameters of other results. Comparison to the fixed rate GAs (figures taken from table 4.5) showed a slight decrease in average best performance from the dynamically configured Multi-GA, although again within standard deviation bounds, as illustrated in table 5.2.

GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Average Best*	*St. Dev.*
Fixed	–	1	-1677.79	0.25
TGA	–	10	-1676.98	1.70
Fixed	75%	1	-1677.58	0.93
MGA		10	-1677.46	0.77
Dynamic	max.	1	-1676.86	1.19
MGA	60%	10	-1676.94	1.29
Dynamic	max.	1	-1676.96	1.38
MGA	75%	10	-1677.19	3.74
Dynamic	max.	1	-1677.21	1.28
MGA	90%	10	-1677.52	0.22

Table 5.2: **2 variable F7 dynamic crossover optimisation results for direct fitness distance calculation at 75% individual crossover compared to fixed rate GAs.**

Generally, dynamic crossover rates calculated by direct chromosome fitness distance produced roughly comparable performance to the fixed rate approach, with little distinction as to the level of dynamic rate application. The experiments were then repeated without individual level crossover, in an attempt to gauge the effect of this important Multi-GA operator (complete results in appendix C, table C.2).

A				B	
Chrom Xover	*Degree of Elitism*	*Average Best*	*St. Dev.*	*Average Best*	*St. Dev.*
60%	0	-1654.26	47.20	-1676.01	1.92
	1	-1633.08	59.02	-1676.86	1.19
	10	-1625.79	62.26	-1676.94	1.29
75%	0	-1618.64	73.22	-1676.61	1.81
	1	-1608.60	70.67	-1676.96	1.38
	10	-1649.75	46.60	-1677.19	3.74
90%	0	-1626.16	59.35	-1676.81	1.94
	1	-1611.88	58.99	-1677.21	1.28
	10	-1625.40	62.46	-1677.52	0.22

Table 5.3: **2 variable F7 dynamic crossover optimisation results for direct fitness distance calculation with individual crossover at 0% (A) and 75% (B).**

It was immediately obvious that experiments without individual level crossover produced noticeably worse results (table 5.3). Across the board, average performance of the GA with 75% individual level crossover enabled was superior. Whilst both GAs showed comparable best ever performance, reflecting the simpler nature of the problem surface, the use of individual level crossover produced better, more consistent results (table 5.3) and, on average, faster convergence (table 5.4).

A			
Chrom Xover	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*
60%	0	105.95	73.36
	1	100.80	97.50
	10	160.20	143.42
75%	0	98.65	59.42
	1	120.50	129.93
	10	98.10	116.26
90%	0	109.15	87.83
	1	134.55	135.25
	10	78.45	65.52

B	
Avg. Best Gen.	*St. Dev.*
82.65	62.30
80.60	93.74
84.95	108.84
99.85	77.59
60.55	28.92
51.80	28.27
88.60	79.52
100.80	123.75
65.80	53.57

Table 5.4: **2 variable F7 dynamic crossover convergence times for direct fitness distance calculation with individual crossover at 0% (A) iand 75% (B).**

Having established a picture for direct fitness distance calculation, experiments were repeated for inverse fitness distance calculation (complete results in appendix C, tables C.3 and C.4). The results showed a repetition of the trend indicated by direct fitness distance calculation concerning individual level crossover. Worse performance from the average best and a larger standard deviation was seen when individual level crossover was removed. Elitism showed no clear trend, but results were mainly comparable (table 5.5).

A			
Chrom Xover	*Degree of Elitism*	*Average Best*	*St. Dev.*
60%	0	-1671.37	14.57
	1	-1635.71	67.36
	10	-1635.07	55.93
75%	0	-1622.00	77.29
	1	-1659.19	41.92
	10	-1636.23	56.67
90%	0	-1653.52	47.41
	1	-1671.07	25.62
	10	-1665.61	35.43

B	
Average Best	*St. Dev.*
-1677.44	1.07
-1676.86	3.39
-1677.88	0.11
-1677.61	0.24
-1677.64	0.90
-1677.09	1.08
-1677.62	0.00
-1677.48	1.17
-1677.58	1.17

Table 5.5: **2 variable F7 dynamic crossover optimisation results for inverse fitness distance calculation with individual crossover at 0% (A) and 75% (B).**

The comparison between inverse and direct fitness distance calculation was more revealing, with slightly improved average best results for inverse fitness distance over direct fitness distance and overall performance comparable to both fixed rate GAs (table 5.6).

A			
Chrom Xover	*Degree of Elitism*	*Average Best*	*St. Dev.*
60%	0	-1676.01	1.92
	1	-1676.86	1.19
	10	-1676.94	1.29
75%	0	-1676.61	1.81
	1	-1676.96	1.38
	10	-1677.19	3.74
90%	0	-1676.81	1.94
	1	-1677.21	1.28
	10	-1677.52	0.22

B	
Average Best	*St. Dev.*
-1677.44	1.07
-1676.86	3.39
-1677.88	0.11
-1677.61	0.24
-1677.64	0.90
-1677.09	1.08
-1677.62	0.00
-1677.48	1.17
-1677.58	1.17

C				
GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Average Best*	*St. Dev.*
Fixed	–	1	-1677.79	0.25
TGA	–	10	-1676.98	1.70
Fixed	75%	1	-1677.58	0.93
MGA		10	-1677.46	0.77

Table 5.6: **2 variable F7 dynamic crossover optimisation results for 75% individual level crossover comparing direct fitness distance calculation (A) with inverse (B) and fixed rate results (C).**

The most noticeable effect was in convergence time, with inverse fitness distance producing around 50% fewer generations than direct fitness distance for an average covergence, in many cases with a standard deviation around 50-66% smaller (table 5.7). Convergence results were very similar to the fixed rate Multi-GA.

A			
Chrom Xover	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*
60%	0	82.65	62.30
	1	80.60	93.74
	10	84.95	108.84
75%	0	99.85	77.59
	1	60.55	28.92
	10	51.80	28.27
90%	0	88.60	79.52
	1	100.80	123.75
	10	65.80	53.57

B	
Avg. Best Gen.	*St. Dev.*
48.35	22.61
42.15	33.80
43.15	46.94
46.55	23.03
30.60	12.00
26.00	9.34
34.90	16.23
30.20	9.80
24.30	8.42

C				
GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*
Fixed	–	1	61.60	38.46
TGA	–	10	347.40	60.35
Fixed	75%	1	30.95	5.72
MGA		10	49.75	103.57

Table 5.7: **2 variable F7 dynamic crossover convergence times for 75% individual level crossover direct fitness distance calculation (A) v. inverse fitness distance (B) and fixed rate results (C).**

5.2.2 *2 variable, F8*

An identical set, performed under the same experimental conditions, was then applied to the F8 problem surface. The results of direct fitness distance calculation at 75% and 0% individual crossover are given in full in appendix C, tables C.5 and C.6, and summarised in table 5.8.

The results for the 2 variable F8 with individual crossover disabled were not as revealing as the F7 surface, with individual level crossover suggesting a slight performance improvement. Average best values were marginally improved, but well within the standard deviation range of the alternatives. The lack of discrimination between the two methods was also shown in the best ever results achieved, with both methods performing comparably on the surface. Analysis of the effects of elitism and dynamic rate application showed little discrimination, with a slight increase in average best as the dynamic rate approached 90% in both tables. Comparison to the fixed rate results given in table 4.6 showed better performance by both traditional and fixed rate Multi-GAs over the direct fitness distance dynamic rate results.

A			
Chrom Xover	*Degree of Elitism*	*Average Best*	*St. Dev.*
60%	0	-180.31	0.16
	1	-180.25	0.19
	10	-180.30	0.15
75%	0	-180.28	0.14
	1	-180.32	0.19
	10	-180.29	0.20
90%	0	-180.35	0.00
	1	-180.49	0.05
	10	-180.41	0.00

B	
Average Best	*St. Dev.*
-180.39	0.09
-180.40	0.06
-180.35	0.14
-180.37	0.12
-180.37	0.08
-180.38	0.17
-180.46	0.06
-180.42	0.10
-180.35	0.00

C				
GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Average Best*	*Std. Dev.*
Fixed	–	1	-180.46	0.10
TGA	–	10	-180.40	0.09
Fixed	75%	1	-180.48	0.12
MGA		10	-180.46	0.00

Table 5.8: **2 variable F8 dynamic crossover optimisation results for direct fitness distance calculation at 0% (A) and 75% (B) individual crossover.**

The overall lack of any discrimination in the results, combined with a good solution from most methods, indicated a particular simplicity in the problem surface itself. Indeed, the overall basin shape of F8 converges to a central global minimum, although the complexity of the surface points towards Mühlenbein's (1991) observations of its difficulty. However, most of the tests on the 2 variable F8 surface produced results close to the global optimum with little discrimination between the methods. Experiments with inverse fitness distance were the next to be performed, giving the results shown in full in appendix C, tables C.7 and C.8.

The results for these experiments showed few differences, with a marginal improvement shown by addition of individual level crossover, although within standard deviation ranges. Examination of best ever performance showed that both methods again found, or came very close, to the global minimum. The alternative rates of crossover and elitism also showed little discrimination.

Comparison to direct fitness distance calculation revealed a slight improvement in average best values, but still within standard deviation results. However, the trend illustrated by the F7 2 variable results of a reduced, more consistent number of generations to converge was repeated (table 5.9). Inverse fitness distance calculation showed average generations to converge of just under 50%

of those for direct fitness distance calculation, with a substantially lower standard deviation for the lower rates of 60% and 75%. Whilst still within the limits of the direct fitness distance standard deviation, the trend for greater consistency of performance (as indicated by smaller standard deviation values) was confirmed with the inverse fitness distance method. In addition, performance was directly comparable to the fixed rate GAs, but again with a smaller standard deviation for the convergence time.

A			
Chrom Xover	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*
60%	0	98.95	102.94
	1	77.55	121.38
	10	77.50	122.92
75%	0	96.85	104.92
	1	88.65	125.28
	10	85.50	130.42
90%	0	67.15	30.93
	1	147.75	84.00
	10	35.30	14.93

B	
Avg. Best Gen.	*St. Dev.*
40.90	21.17
41.05	28.70
26.20	15.57
50.65	27.68
47.05	32.35
53.60	80.82
65.70	88.09
83.70	123.71
81.95	133.97

C				
GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*
Fixed	–	1	58.00	100.57
TGA	–	10	31.35	20.38
Fixed	75%	1	56.65	90.16
MGA		10	30.40	35.19

Table 5.9: **2 variable F8 dynamic crossover convergence times for 75% individual level crossover direct fitness distance calculation (A) v. inverse fitness distance (B) and fixed rate results (C).**

5.2.3 *10 variable, F7*

Following the results obtained on the 2 variable surfaces, with many of the different approaches coming close to, or locating, the global minimum, the problems were scaled up to the 10 variable surfaces. In doing so, not only were experiments performed on a more complex surface, but a greater number of feature chromosomes were provided, giving individual level crossover a more significant role to play. As before, the same series of experiments were run, the results of which are presented in full in appendix C, tables C.9 and C.10 for direct fitness distance calculation at 75% and 0% individual level crossover respectively.

Results were interesting, showing recurrence of the trends indicated by 2 variable F7. The distinction between individual level crossover at 0% and 75% was clear, with the 75% rate producing statistically significant improvements in average best performance (illustrated in table 5.10). Standard deviation of average best was also consistently lower, in the region of 50% for many cases. Comparison with the fixed rate GAs (data from table 4.7) indicated average best results slightly better than the traditional GA and comparable to the Multi-GA, although within standard deviations. Elitism showed a slight improvement at 75% individual level crossover in both average best and best ever results in all but one case. Application of different rates of chromosome crossover showed little difference though, with comparable results across the board.

A			
Chrom Xover	*Degree of Elitism*	*Average Best*	*St. Dev.*
60%	0	-4080.17	357.49
	1	-4079.80	259.53
	10	-4153.58	204.96
75%	0	-4036.30	303.11
	1	-4165.12	269.31
	10	-4093.35	272.14
90%	0	-4123.00	333.71
	1	-4050.74	346.97
	10	-4049.48	296.00

B	
Average Best	*St. Dev.*
-4703.22	149.57
-4719.32	168.38
-4743.73	133.10
-4708.18	173.92
-4702.17	167.48
-4704.02	228.28
-4620.68	159.24
-4730.17	175.73
-4691.05	167.27

C				
GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Average Best*	*Std. Dev.*
Fixed	–	1	-4640.83	139.57
TGA	–	10	-4710.48	142.74
Fixed	75%	1	-4718.59	172.31
MGA		10	-4685.29	170.79

Table 5.10: **10 variable F7 dynamic crossover optimisation results for direct fitness distance calculation at 0% (A) and 75% (B) individual crossover, compared to the fixed rate results (C).**

Best ever performance also showed improved results at the 75% rate, coming closer to the global optimum of -5029.83 on several occasions (table 5.11).

A			B
Chrom Xover	*Degree of Elitism*	*Best Ever*	*Best Ever*
60%	0	-4833.53	-5014.50
	1	-4485.72	-5023.92
	10	-4501.45	-5017.69
75%	0	-4573.20	-4933.06
	1	-4641.08	-5017.82
	10	-4476.53	-5016.96
90%	0	-4840.40	-4892.87
	1	-5017.19	-5018.98
	10	-4707.77	-4929.98

Table 5.11: 10 variable F7 dynamic crossover best ever optimisation results for direct fitness distance calculation at 0% (A) and 75% (B) individual crossover.

Elitism also had an effect on the convergence time, with experiments using elitism showing fewer average generations to convergence (table 5.12), whilst maintaining comparable average best performance. The fall in convergence times and the effects of 10 member elitism confirmed earlier conclusions that elitism concentrates the search effort, with excessive elitism indicating premature convergence.

A				B	
Chrom Xover	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*	*Avg. Best Gen.*	*St. Dev.*
60%	0	475.65	41.01	407.70	94.33
	1	484.05	16.10	336.95	67.93
	10	464.75	40.00	344.50	67.93
75%	0	482.20	31.25	430.00	64.48
	1	484.40	11.97	335.35	59.15
	10	461.15	34.05	309.40	63.14
90%	0	470.90	34.24	402.95	68.85
	1	464.65	32.93	352.55	49.72
	10	439.95	55.94	316.95	45.20

Table 5.12: 10 variable F7 dynamic crossover convergence time results for direct fitness distance calculation at 0% (A) and 75% (B) individual crossover.

The experiments were then repeated using inverse fitness distance calculation, the results of which are presented in full in appendix C, tables C.11 and C.12. Results for inverse fitness distance calculation showed similar trends to those of direct fitness distance. Again, the use of individual level crossover resulted in statistically significantly improved performance. Comparison to the fixed rate traditional GA (data from table 4.7) revealed better performance from the 75% dynamic chromosome crossover rate and comparable fixed rate Multi-GA

performance. Inverse fitness distance showed a slightly improved performance in average best for single member elitism (although within standard deviations), lending weight to the concepts behind this calculation method outlined in section 5.1.1. These results are illustrated in table 5.13. As previously demonstrated, average convergence times were slightly improved with inverse fitness distance, although not with statistical significance.

A			
Chrom Xover	*Degree of Elitism*	*Average Best*	*St. Dev.*
60%	0	-4182.23	288.84
	1	-4253.49	294.08
	10	-4181.14	304.47
75%	0	-4205.79	220.92
	1	-4225.30	236.97
	10	-4083.70	261.65
90%	0	-4339.10	339.95
	1	-4173.48	299.52
	10	-4112.71	238.47

B	
Average Best	*St. Dev.*
-4737.47	180.12
-4742.36	169.84
-4669.78	188.81
-4680.43	146.40
-4754.75	186.88
-4711.92	137.17
-4794.70	133.26
-4719.03	138.00
-4771.21	120.93

C				
GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Average Best*	*Std. Dev.*
Fixed	–	1	-4640.83	139.57
TGA	–	10	-4710.48	142.74
Fixed	75%	1	-4718.59	172.31
MGA		10	-4685.29	170.79

Table 5.13: 10 variable F7 dynamic crossover optmisation results for inverse fitness distance calculation at 0% (A) and 75% (B) individual crossover, compared to the fixed rate results (C).

5.2.4 *10 variable, F8*

Completing the series of test surfaces, the experiments were repeated on the 10 variable F8 problem. Results of the direct fitness distance calculation experiments are given in full in appendix C, tables C.13 and C.14.

The results indicated a great deal about the complexity of the problem surface, with nearly all cases coming close to the global minimum. A slight increase in discrimination over the 2 variable form of the problem was seen, with the results for 75% individual crossover giving slightly higher average best results and a greatly reduced standard deviation. Indeed, in most cases, the standard deviation values were so small as to be unrepresentable by 2 decimal places. Elitism appeared to have little effect, with minimal performance discrimination

between the three varying rates. In addition, the results were comparable, but marginally less impressive, than the fixed rate GA tests (data from table 4.8).

A			
Chrom Xover	*Degree of Elitism*	*Average Best*	*St. Dev.*
60%	0	-180.34	0.18
	1	-180.41	0.05
	10	-180.35	0.08
75%	0	-180.38	0.10
	1	-180.39	0.07
	10	-180.34	0.09
90%	0	-180.36	0.09
	1	-180.36	0.11
	10	-180.37	0.16

B	
Average Best	*St. Dev.*
-180.41	0.05
-180.42	0.08
-180.44	0.00
-180.41	0.04
-180.45	0.00
-180.41	0.00
-180.43	0.06
-180.43	0.00
-180.44	0.00

C				
GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Average Best*	*Std. Dev.*
Fixed	–	1	-180.45	0.00
TGA	–	10	-180.45	0.08
Fixed	75%	1	-180.47	0.10
MGA		10	-180.47	0.05

Table 5.14: **10 variable F8 dynamic crossover optimisation results for direct fitness distance calculation at 0% (A) and 75% (B) individual crossover, compared to the fixed rate results (C).**

Whilst little discrimination in average best values was seen between 0% and 75% individual level crossover, a marginal improvement in best ever performance was indicated, with 6 cases finding the global optimum at 75% and only 1 at 0% (table 5.15).

A		
Chrom Xover	*Degree of Elitism*	*Best Ever*
60%	0	-180.49
	1	-180.50
	10	-180.47
75%	0	-180.48
	1	-180.49
	10	-180.49
90%	0	-180.49
	1	-180.49
	10	-180.47

B
Best Ever
-180.49
-180.50
-180.50
-180.50
-180.50
-180.48
-180.49
-180.50
-180.50

Table 5.15: **10 variable F8 dynamic crossover best ever optimisation results for direct fitness distance calculation at 0% (A) and 75% (B) individual crossover.**

The trend for reduced convergence time was also repeated, with some results giving statistically significant reductions in the average number of generations required for convergence (table 5.16).

A				B	
Chrom Xover	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*	*Avg. Best Gen.*	*St. Dev.*
60%	0	438.05	44.22	359.30	63.89
	1	320.10	64.26	253.55	69.40
	10	304.20	90.27	239.70	70.41
75%	0	415.80	60.24	374.85	55.84
	1	341.55	68.79	216.90	34.48
	10	300.35	58.12	188.00	51.10
90%	0	408.00	49.67	375.55	52.24
	1	305.45	74.12	257.15	53.92
	10	290.40	63.92	200.75	45.76

Table 5.16: 10 variable F8 dynamic crossover convergence time results for direct fitness distance calculation at 0% (A) and 75% (B) individual crossover.

Experiments were then repeated using inverse fitness distance, with the results shown in full in appendix C, tables C.15 and C.16. The results indicated little distinction, with comparable performance again illustrating the apparent ease with which the F8 problem surface was solved. Average best performance was a little improved over direct fitness distance, although the discrimination was not as distinct as for the F7 surfaces. Again, performance was comparable to the fixed rate traditional and Multi-GAs (data from table 4.8, illustrated in table 5.17). Other trends described earlier were repeated, with elitism indicating a drop in the average number of generations to converge.

A			
Chrom Xover	*Degree of Elitism*	*Average Best*	*St. Dev.*
60%	0	-180.41	0.05
	1	-180.42	0.08
	10	-180.44	0.00
75%	0	-180.41	0.04
	1	-180.45	0.00
	10	-180.41	0.00
90%	0	-180.43	0.06
	1	-180.43	0.00
	10	-180.44	0.00

B	
Average Best	*St. Dev.*
-180.41	0.13
-180.47	0.04
-180.47	0.10
-180.43	0.04
-180.47	0.04
-180.46	0.08
-180.42	0.00
-180.45	0.06
-180.48	0.00

C				
GA Tested	*Chromsome Xover rate*	*Degree of Elitism*	*Average Best*	*Std. Dev.*
Fixed	–	1	-180.45	0.00
TGA	–	10	-180.45	0.08
Fixed	75%	1	-180.47	0.10
MGA		10	-180.47	0.05

Table 5.17: 10 variable F8 dynamic crossover optimisation results for 75% individual crossover, comparing direct fitness distance calculation (A), inverse fitness distance calculation (B) and the fixed rate results (C).

5.2.5 *A summary of dynamic crossover*

In summary, the dynamic crossover experiments revealed a number of trends. The 2 variable F7 and F8 surfaces showed good performance from the dynamic rate Multi-GA, which found the global solution in most, if not all, cases. F8 provided very little discriminating behaviour, in contrast to the observations of its difficulty made by Mühlenbein (1991), with both traditional and Multi-GAs performing rather well. Analysis of Multi-GA individual level crossover showed a distinct improvement on the 2 variable F7, with an improved average best performance at the 75% rate over the 0% rate. The F8 surface showed little discrimination however, possibly due to the nature of the surface.

Study of the general shape of F8 reveals little need for the global searching required with F7, as F8 leads down towards the location of the global optimum. Searching is required however at the local level – a function which is better fulfilled by chromosome level crossover. In removing individual level crossover, the only remaining distinction would then relate the chromosome searching to the rate calculation method, where a difference is evident only in the time taken for the local (chromosome level) search to achieve its goal. The results indicated this difference, lending weight to the conclusion that individual level crossover

demonstrates a more global search, with chromosome level crossover performing local searching. Study of the alternative methods of rate calculation showed a slight improvement in F7 average best performance through the use of the inverse fitness distance method over that of direct fitness distance. The number of generations taken to converge was reduced with inverse fitness distance, producing a more consistent (smaller standard deviation) convergence in approximately half the time of direct fitness distance. Indications of this trend were found for the 2 variable F8 surface. Overall performance of dynamic chromosome crossover rates showed little distinction between the three alternative rates of 60%, 75% and 90%, indicating comparable performance to the fixed rate GA results presented in chapter 4.

Extension of the experiments to the more complex 10 variable surfaces verified the trends described above, especially on the F7 surface. In particular, the increased complexity of 10 variable F7 produced greater discrimination in the results, showing the use of individual level crossover to provide a statistically significant Multi-GA performance improvement. Single member elitism also showed improvements, both in terms of best ever results and in the number of generations taken to achieve comparable results. Finally, there was a repetition of the trends of improved performance with inverse fitness distance calculation and little discrimination with alternative rate application.

In addition, the apparent simplicity of F8 was repeated in the 10 variable case, with both the individual level crossover rates of 75% and 0% and the three dynamic crossover rates showing similar performance. A small, but not statistically significant, improvement in average best performance was seen with inverse fitness distance compared to direct fitness distance, but on the whole the tests on both 2 and 10 variable F8 illustrated comparable performance and added little to the results, in contrast to Mühlenbein's (1991) observations.

5.3 Dynamic Mutation

Having obtained the results from the dynamic crossover experiments on each surface, investigations into dynamic mutation were then carried out. Experiments were performed by fixing chromosome and individual level crossover rates at 75%. Application of the four methods of dynamic rate calculation were conducted, with mutation rates floating up to maximum levels of 10%, 30% and 50% across the entire individual.

Due to time and CPU constraints, a full and exhaustive test of each possible parameter combination was not a practical option. In addition, results of the

experiments into dynamic crossover indicated little distinction between dynamic rates applied at 60, 75 and 90%. In order to accurately assess the effects of dynamic mutation alone, crossover rates at both individual and chromosome level had to be fixed. Results on previous experiments indicated that individual level crossover provided a beneficial input, so the rate of 75% was again used and experiments without individual level crossover were not performed.

In order to verify the lack of distinction between chromosome crossover rates, the initial series of experiments into direct rank dynamic mutation on the 2 variable surfaces were performed at three fixed rates of 60%, 75% and 90%. Following the results of these preliminary experiments, with both rank based and chromosome fitness distance based dynamic mutation being tested, a single chromosome level crossover rate was selected for remaining tests. 75% chromosome crossover was chosen, being the median of the three rates applied and one that falls between the parameters recommended by De Jong (1975) and Grefenstette (1986).

Experiments were performed on each of the two combinations of the F7 and F8 surfaces, calculating the dynamic mutation rate using both chromosome fitness distance and rank based calculation methods in direct and inverse proportion. The other parameters were kept consistent with previous runs and statistically significant results obtained by execution of 20 trials per experiment.

The main objectives of the dynamic mutation test series were to:

- Assess the effects of dynamic rate calculation at varying maximum rates.
- Assess the usefulness of applying dynamic rates in direct and inverse proportion to relative fitness and population ranking.
- Assess the effects of elitism at increasing dynamic mutation rates.

Sections 5.3.1 - 5.3.4 present the results, as before, for each surface in turn.

5.3.1 *2 variable, F7*

The first series of tests was carried out using direct rank based calculation at the three chromosome crossover rates, with the full results shown in appendix C, table C.17.

The results confirmed those obtained for dynamic crossover, showing little distinction between chromosome crossover rates of 60%, 75% and 90%. Experiments on F7 carried out after this point utilised only the 75%

chromosome crossover rate. The application of dynamic mutation revealed interesting results. For higher levels of mutation, up to 30% and 50%, the results were good with the global minimum being found in almost every GA trial. Consistency was also good, indicated by average best standard deviation results too small to express in 2 decimal places. In all cases, the best ever result showed location of the global minimum, also indicated by the average best at higher mutation rates.

Max Mtn.	*Degree of Elitism*	*Average Best*	*St. Dev.*	*Best Ever*
10%	0	-1677.02	4.61	-1677.97
	1	-1677.81	0.00	-1677.97
	10	-1677.64	1.03	-1677.97
30%	0	-1677.94	0.00	-1677.97
	1	-1677.97	0.00	-1677.97
	10	-1677.97	0.00	-1677.97
50%	0	-1677.97	0.00	-1677.97
	1	-1677.97	0.00	-1677.97
	10	-1677.97	0.00	-1677.97

Table 5.18: 2 variable F7 dynamic mutation performance for direct rank based calculation at 75% chromosome crossover.

Interesting effects were noticed in the number of generations taken to converge, with 10% elitist mutation taking fewer average generations to converge than the higher rates (table 5.19). This indicated confirmation of the previously identified effects of higher mutation rates, namely increased disruption leading to greater convergence times. However, the simplicity of the problem surface did not lead to a loss of performance by high mutation in this case, with the increased searching encouraging location of the global minimum.

Max Mtn.	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*
10%	0	120.35	129.93
	1	29.90	8.23
	10	24.55	6.38
30%	0	110.05	96.22
	1	41.25	14.88
	10	45.75	80.69
50%	0	222.20	115.50
	1	71.75	17.80
	10	29.00	7.31

Table 5.19: 2 variable F7 dynamic mutation convergence times for direct rank based calculation at 75% chromosome crossover.

Results for the 10 variable surface were expected to show a more noticeable indication of the expected behaviour in terms of global performance, as well as convergence time. The elitism results also backed this conclusion, indicating a fall in average convergence times in each of the three crossover tests at higher mutation rates. This confirmed the previous hypothesis, indicating that GA searches with high disruption through large scale mutation and no mechanism for retaining good solutions take longer to find good results. On the whole, the addition of elitism led to more consistent convergence times, with lower standard deviations (table 5.19).

The next step was to perform experiments for inverse rank based calculation, with the full results given in appendix C, table C.18. The results were very similar to those of the previous table, with the global optimum being found in all cases under best ever result and, at higher mutation, in all trials performed.

Max Mtn.	*Degree of Elitism*	*Average Best*	*St. Dev.*	*Best Ever*
10%	0	-1677.50	0.91	-1677.97
	1	-1677.81	0.00	-1677.97
	10	-1677.94	0.20	-1677.97
30%	0	-1677.96	0.00	-1677.97
	1	-1677.97	0.00	-1677.97
	10	-1677.97	0.00	-1677.97
50%	0	-1677.97	0.00	-1677.97
	1	-1677.97	0.00	-1677.97
	10	-1677.97	0.00	-1677.97

Table 5.20: 2 variable F7 dynamic mutation performance for inverse rank based calculation at 75% chromosome crossover.

The trend in elitism was repeated, generally leading to a fall in the average number of generations to converge and, on the whole, a tightening of the associated standard deviation. As indicated by the dynamic crossover series, inverse calculation suggested reduced average convergence times for methods without elitism and an indicated performance improvement for 1 elitism results. Considering the previous tables, illustrating longer convergence times at high mutation rates (due to the increased disruption), the adoption of inverse rank based calculation appeared to counter this effect, bringing convergence times down to more sensible figures for comparably high mutation rates (table 5.21B).

A				B	
Max Mtn.	*Degree of Elitism*	*Avg. Best Gen.*	*St. Dev.*	*Avg. Best Gen.*	*St. Dev.*
10%	0	120.35	129.93	109.75	135.94
	1	29.90	8.23	39.15	30.06
	10	24.55	6.38	24.60	6.51
30%	0	110.05	96.22	62.60	72.45
	1	41.25	14.88	31.90	9.31
	10	45.75	80.69	32.60	21.88
50%	0	222.20	115.50	68.50	63.71
	1	71.75	17.80	35.70	9.70
	10	29.00	7.31	41.70	63.24

Table 5.21: 2 variable F7 dynamic mutation convergence times for direct rank based (A) and inverse rank based (B) calculation at 75% chromosome crossover.

Following experiments into rank based calculation, the method adopted in the dynamic crossover series – fitness distance based calculation – was tested, providing a mutation rate based solely on relative chromosome performance. Full results for direct and inverse fitness distance calculation at 75% chromosome crossover are given in appendix C, tables C.19 and C.20. The results again illustrated the trend of an improved performance from the inverse approach over the direct approach, with a more consistent average best location of the global optimum, which was not achieved at all with direct fitness distance calculation (table 5.22).

A				B	
Max Mtn.	*Degree of Elitism*	*Average Best*	*St. Dev.*	*Average Best*	*St. Dev.*
10%	0	-1671.64	25.75	-1677.58	0.84
	1	-1677.44	1.15	-1677.97	0.00
	10	-1677.68	0.00	-1677.65	0.98
30%	0	-1671.96	22.59	-1677.97	0.38
	1	-1671.49	25.71	-1677.97	0.00
	10	-1677.60	0.82	-1677.97	0.00
50%	0	-1677.51	1.01	-1677.96	0.00
	1	-1677.75	0.00	-1677.97	0.00
	10	-1677.55	0.65	-1677.97	0.00

Table 5.22: 2 variable F7 dynamic mutation convergence times for direct fitness distance based (A) and inverse fitness distance based (B) calculation at 75% chrom. crossover.

Fitness diversity, indicated by a larger range between average best and worst values, coupled with a noticeable difference between average best values,

suggested better performance both in terms of optima achieved and diversity of search space (table 5.23).

A						B			
Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*
10%	0	-1671.64	25.75	-97.53	74.94	-1677.58	0.84	-2.67	1.62
	1	-1677.44	1.15	-123.02	71.39	-1677.97	0.00	-2.04	0.00
	10	-1677.68	0.00	-128.36	88.83	-1677.65	0.98	-2.35	0.96
30%	0	-1671.96	22.59	-117.72	81.35	-1677.97	0.38	-2.11	0.10
	1	-1671.49	25.71	-99.18	72.58	-1677.97	0.00	-2.04	0.00
	10	-1677.60	0.82	-165.58	77.60	-1677.97	0.00	-2.04	0.00
50%	0	-1677.51	1.01	-100.37	69.69	-1677.96	0.00	-2.57	0.72
	1	-1677.75	0.00	-111.22	59.19	-1677.97	0.00	-2.05	0.03
	10	-1677.55	0.65	-129.06	65.41	-1677.97	0.00	-2.04	0.00

Table 5.23: 2 variable F7 dynamic mutation best and worst optimisation results for direct fitness distance based (A) and inverse fitness distance based (B) calculation at 75% chrom. crossover.

Inverse fitness distance calculation again showed the trend of elitism to reduce convergence times for all rates of mutation. Interestingly, direct fitness distance calculation showed very fast convergence times, explaining the poorer average best values to be caused by premature convergence in many of the 20 trials performed (table 5.24).

A				B	
Max Mtn.	*Degree of Elitism*	*Avg. Best Gens.*	*St. Dev.*	*Avg. Best Gens.*	*St. Dev.*
10%	0	31.85	11.31	78.30	77.43
	1	28.20	8.00	36.70	9.80
	10	21.15	5.43	30.10	21.80
30%	0	27.95	14.35	154.30	65.12
	1	30.30	5.06	60.75	16.82
	10	21.40	3.63	31.35	17.49
50%	0	33.50	11.75	217.15	144.43
	1	25.55	5.71	121.90	58.14
	10	19.25	4.77	34.20	9.49

Table 5.24: 2 variable F7 dynamic mutation convergence times for direct fitness distance based (A) and inverse fitness distance based (B) calculation at 75% chrom. crossover.

Comparison with the rank based results indicated comparable performance in most cases. Inverse ranking produced faster convergence times in many cases, particularly at higher mutation rates (table 5.25).

A			
Max Mtn.	*Degree of Elitism*	*Avg. Best Gens.*	*St. Dev.*
10%	0	109.75	135.94
	1	39.15	30.06
	10	24.60	6.51
30%	0	62.60	72.45
	1	31.90	9.31
	10	32.60	21.88
50%	0	68.50	63.71
	1	35.70	9.70
	10	41.70	63.24

B	
Avg. Best Gens.	*St. Dev.*
78.30	77.43
36.70	9.80
30.10	21.80
154.30	65.12
60.75	16.82
31.35	17.49
217.15	144.43
121.90	58.14
34.20	9.49

Table 5.25: 2 variable F7 dynamic mutation convergence times for inverse rank based (A) and inverse fitness distance based (B) calculation at 75% chrom. crossover.

Inverse fitness distance produced a greater, more consistent fitness diversity at the 10% mutation rate (table 5.26).

A					
Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*
10%	0	-1677.50	0.91	-30.22	49.44
	1	-1677.81	0.00	-8.73	15.79
	10	-1677.94	0.20	-16.76	38.82

B			
Average Best	*St. Dev.*	*Average Worst*	*St. Dev.*
-1677.58	0.84	-2.67	1.62
-1677.97	0.00	-2.04	0.00
-1677.65	0.98	-2.35	0.96

Table 5.26: 2 variable F7 dynamic mutation fitness diversity for inverse rank based (A) and inverse fitness distance based (B) calculation at 75% chrom. crossover.

Comparison to the fixed rate GAs described by table 4.5 revealed good performance across the board (for similar experimental conditions). Average best values marginally outperformed those of the fixed rate GAs, with a reduced standard deviation over the traditional GA in most cases. Convergence times were comparable to the fixed rate Multi-GA and improved over the fixed rate traditional GA (table 5.27).

A					
Max Mtn.	*Degree of Elitism*	*Average Best*	*St. Dev.*	*Avg. Best Gens.*	*St. Dev.*
10%	1	-1677.97	0.00	36.70	9.80
	10	-1677.65	0.98	30.10	21.80
30%	1	-1677.97	0.00	60.75	16.82
	10	-1677.97	0.00	31.35	17.49
50%	1	-1677.97	0.00	121.90	58.14
	10	-1677.97	0.00	34.20	9.49

B					
GA Tested	*Degree of Elitism*	*Average Best*	*Std. Dev.*	*Avg. Best Gen.*	*Std. Dev.*
Fixed	1	-1677.79	0.25	61.60	38.46
TGA	10	-1676.98	1.70	347.40	60.35
Fixed	1	-1677.58	0.93	30.95	5.72
MGA	10	-1677.46	0.77	49.75	103.57

Table 5.27: 2 variable F7 dynamic mutation convergence times for inverse fitness distance calculation at 75% chrom. crossover (A) and fixed rate GAs (B).

5.3.2 *2 variable, F8*

Experiments on the F8 surface were performed initially at all three chromosome crossover rates, as with the F7 results presented in section 5.3.1, in order to verify the expected behaviour on this very differently shaped problem surface. The first set of results, for direct rank based calculation, are shown in full in appendix C, table C.21. The results confirmed the conclusions drawn from other experiments so far that there is little difference between the three rates of chromosome crossover. This led to the decision to run remaining experiments at 75% chromosome crossover only for the F8 surface as well.

The results also showed similar trends to those exhibited by F7 experiments, with increasing mutation rate leading to marginally better average best performance and an increase in the average number of generations required to converge, particularly for experiments with no elitism. Identification of the other trends noted with F7 were difficult to pick out here, due again to the simple nature of the problem surface and consequently the good performance of the GA. In all cases, the results were very close to, or at, the global minimum with no real evidence of discrimination apart from the effect of elitism on convergence times. Experiments were then carried out on the alternative rate calculation methods to complete the result sets and investigate whether or not any discrimination was introduced by these other methods. The results for inverse rank based calculation are given in appendix C, table C.22. The results

added little to the debate, showing comparable performance to direct rank based calculation, although demonstrating marginally increased standard deviation values for some average best values and convergence rates. Trends in slightly lower average convergence times for single and 10 member elitism over no elitism were mainly indicated (table 5.28) but the results in other areas were, on the whole, comparable.

Max Mtn.	*El'tm*	*Avg. Best Gen.*	*St. Dev.*
10%	0	64.45	61.14
	1	55.65	94.24
	10	46.65	74.46
30%	0	131.90	106.61
	1	122.65	134.05
	10	188.50	161.89
50%	0	142.75	115.02
	1	96.50	94.63
	10	91.40	105.53

Table 5.28: 2 variable F8 dynamic mutation convergence times for inverse rank based calculation at 75% chromosome crossover.

With little new information added, the remaining 2 experiments into direct and inverse chromosome fitness distance calculation were performed. These results are presented in appendix C, tables C.23 and C.24 respectively. Direct fitness distance results showed a similar trend to those presented for F7 (table 5.22), producing a decline in the average best performance even on this simple surface. In addition, average convergence times were consistently low across all three mutation rates (table 5.29), indicating rapid convergence to some sub-optimal results. In other respects, the results were comparable to others presented for F8, indicating the lack of distinction provided by this problem surface.

Max Mtn.	*El'tm*	*Avg. Best Gen.*	*St. Dev.*
10%	0	48.10	81.08
	1	33.00	5.98
	10	31.40	35.81
30%	0	35.80	28.72
	1	38.05	16.83
	10	26.55	19.72
50%	0	32.60	9.91
	1	30.90	11.24
	10	27.80	23.29

Table 5.29: 2 variable F8 dynamic mutation convergence times for direct fitness distance based calculation at 75% chromosome crossover.

Results for inverse fitness distance calculation were also very much comparable to those already presented. The previously identified trends regarding 0 and 1 member elitism were repeated, whilst good, comparable performance was demonstrated across the range of dynamic mutation rates tested. The average best value obtained indicated marginally better performance than direct fitness distance, although within many standard deviation ranges (table 5.30). Results were comparable with both the rank based dynamic mutation methods and the fixed rate GAs (data from table 4.6), with slightly improved average best performance over the traditional GA, but no real significant differences.

A			
Max Mtn.	*Deg. of El'tm*	*Average Best*	*St. Dev.*
10%	0	-180.45	0.00
	1	-180.47	0.07
	10	-180.47	0.10
30%	0	-180.45	0.08
	1	-180.48	0.11
	10	-180.47	0.09
50%	0	-180.47	0.04
	1	-180.47	0.10
	10	-180.46	0.00

B	
Average Best	*St. Dev.*
-180.48	0.13
-180.49	0.10
-180.48	0.06
-180.50	0.00
-180.50	0.05
-180.50	0.08
-180.49	0.06
-180.50	0.12
-180.50	0.07

C			
GA Tested	*Deg. of El'tm*	*Average Best*	*Std. Dev.*
Fixed	1	-180.46	0.10
TGA	10	-180.40	0.09
Fixed	1	-180.48	0.12
MGA	10	-180.46	0.00

Table 5.30: **2 variable F7 dynamic mutation convergence times for direct fitness distance based (A), inverse fitness distance based (B) calculation at 75% chrom. crossover and fixed rate GAs (C).**

The trend identified in the F7 tests of longer average convergence times for inverse fitness distance was also repeated here, along with greater standard deviations on the convergence times (table 5.31).

A			
Max Mtn.	*Degree of Elitism*	*Avg. Best Gens.*	*St. Dev.*
10%	0	48.10	81.08
	1	33.00	5.98
	10	31.40	35.81
30%	0	35.80	28.72
	1	38.05	16.83
	10	26.55	19.72
50%	0	32.60	9.91
	1	30.90	11.24
	10	27.80	23.29

B	
Avg. Best Gens.	*St. Dev.*
137.85	117.49
82.30	90.57
54.15	90.53
231.70	133.51
157.20	117.82
77.85	55.89
268.30	129.07
214.45	120.62
90.10	94.28

Table 5.31: **2 variable F8 dynamic mutation convergence times for direct fitness (A) and inverse fitness distance based (B) calculation at 75% chrom. crossover.**

5.3.3 *10 variable, F7*

Following experiments on the 2 variable surface, greater discrimination and complexity was introduced and tested with both the rank based and fitness distance dynamic mutation, applying chromosome and individual level crossover at 75%. The results for direct rank calculation are given in full in appendix C, table C.25. The results showed comparable performance across the three mutation rates, with smaller average best standard deviations for the 30% rate (table 5.32A). Interestingly, the trend of a fall in convergence time with elitism was reversed here, although the use of elitism did introduce more consistency to the average convergence time (indicated by a fall in standard deviation as the level of elitism rose, shown in table 5.32B). Overall, average best performance improved as elitism increased, whilst best ever performance showed the best results for 10% dynamic mutation, being comparable elsewhere (table 5.32C).

Average Performance			
Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*
10%	0	-4854.05	117.22
	1	-4921.26	101.97
	10	-4924.08	98.44
30%	0	-4855.29	55.50
	1	-4950.10	68.18
	10	-5012.51	35.94
50%	0	-4381.06	109.04
	1	-4762.79	155.32
	10	-5008.80	34.81

Best Ever
Best Ever
-5022.12
-5029.65
-5029.83
-4983.87
-5019.45
-5029.77
-4610.29
-4968.24
-5028.88

Av. Convergence	
Avg. Best Gen.	*St. Dev.*
362.30	101.72
459.20	31.50
400.50	90.46
384.05	88.70
429.80	52.45
485.30	13.19
225.70	112.32
398.40	73.52
475.45	20.68

Table 5.32: 10 variable F7 dynamic mutation results for direct rank calculation at 75% chromosome crossover, showing average performance, Best ever performance and average convergence time.

To a large extent the observations made from direct rank based calculation were repeated in the experiments with inverse rank based calculation (given in full in appendix C, table C.26). Average best performance showed a slight improvement in a number of cases over direct rank calculation, although within standard deviation ranges. Elitism, on the whole, indicated a slightly improved average best value (table 5.34). Best ever values were more noticeably improved at the higher mutation rates of 30% and 50% (with inverse ranking), shown in table 5.33. Overall, there was little statistically significant difference in performance, but indications of improved best ever results at higher mutation rates were seen when compared to direct ranking.

A		
Max Mtn.	*Degree of Elitism*	*Best Ever*
10%	0	-5022.12
	1	-5029.65
	10	-5029.83
30%	0	-4983.87
	1	-5019.45
	10	-5029.77
50%	0	-4610.29
	1	-4968.24
	10	-5028.88

B
Best Ever
-5025.33
-5028.53
-5028.67
-5027.06
-5029.80
-5029.83
-5006.29
-5029.19
-5026.64

C		
GA Tested	*Degree of Elitism*	*Best Ever*
Fixed	1	-4901.57
TGA	10	-4932.30
Fixed	1	-5020.01
MGA	10	-5016.82

Table 5.33: **10 variable F7 dyn. mt'n best ever optimisation results for direct rank based (A), inverse rank based (B) calculation at 75% chrom. crossover and fixed rate GAs (C)**

Comparison with both fixed rate experiments (data from table 4.7) showed good results. Performance, measured in terms of average best (table 5.34) and best ever (table 5.33), was improved with dynamic mutation rates in almost all cases of both direct and inverse ranking, some with statistical significance. The difference was more noticeable when looking at inverse rank based calculation, with more results indicating statistically significant performance improvements. Best ever performance was also encouraging, with the majority of direct rank experiments producing comparable or better results and all but one of the inverse rank experiments outperforming both fixed rate GAs (table 5.33).

A			
Max Mtn.	*Degree of Elitism*	*Average Best*	*St. Dev.*
10%	0	-4854.05	117.22
	1	-4921.26	101.97
	10	-4924.08	98.44
30%	0	-4855.29	55.50
	1	-4950.10	68.18
	10	-5012.51	35.94
50%	0	-4381.06	109.04
	1	-4762.79	155.32
	10	-5008.80	34.81

B	
Average Best	*St. Dev.*
-4900.97	79.27
-4891.56	131.14
-4930.90	104.66
-4973.31	66.67
-5025.21	5.54
-5009.52	40.53
-4975.80	38.88
-5025.66	0.00
-5026.18	3.66

C				
GA Tested	*Degree of Elitism*	*Average Best*	*Std. Dev.*	*Best Ever*
Fixed	1	-4640.83	139.57	-4901.57
TGA	10	-4710.48	142.74	-4932.30
Fixed	1	-4718.59	172.31	-5020.01
MGA	10	-4685.29	170.79	-5016.82

Table 5.34: **10 variable F7 dynamic mutation optimisation results for direct (A) and inverse rank (B) calculation at 75% chrom. crossover vs. fixed rate GAs (C).**

The final set of tests applied direct and inverse fitness distance, with the results given in full in appendix C, tables C.27 and C.28. Direct fitness distance showed a statistically significant degradation in average best performance from its rank based counterparts across many experiments tested. Performance for direct fitness calculation showed slightly poorer average best performance than for the fixed rate GAs, although within standard deviation ranges (table 5.35).

A			
Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*
10%	0	-4615.54	194.00
	1	-4577.87	177.80
	10	-4612.34	205.98
30%	0	-4674.82	166.18
	1	-4682.59	159.06
	10	-4591.07	202.09
50%	0	-4693.79	178.51
	1	-4665.67	168.22
	10	-4632.70	206.42

B	
Average Best	*St. Dev.*
-4854.05	117.22
-4921.26	101.97
-4924.08	98.44
-4855.29	55.50
-4950.10	68.18
-5012.51	35.94
-4381.06	109.04
-4762.79	155.32
-5008.80	34.81

C			
GA Tested	*El'tm*	*Average Best*	*Std. Dev.*
Fixed TGA	1	-4640.83	139.57
	10	-4710.48	142.74
Fixed MGA	1	-4718.59	172.31
	10	-4685.29	170.79

Table 5.35: **10 variable F7 dynamic mutation optimisation results for direct fitness distance (A) and direct rank based (B) calculation at 75% chrom. crossover vs. fixed rate GAs (C).**

In addition, the drop in performance was again matched by a substantial fall in convergence times, indicating direct fitness distance to be a method encouraging fast convergence, but arriving at frequently sub-optimal solutions. Convergence time was also substantially below that of the fixed rate results.

A			
Max Mtn.	*El'tm*	*Avg. Best Gens*	*St. Dev.*
10%	0	95.70	23.56
	1	91.55	7.69
	10	65.70	4.65
30%	0	109.40	20.02
	1	96.15	6.95
	10	64.60	5.96
50%	0	107.70	24.60
	1	100.30	8.57
	10	67.15	8.55

B	
Avg. Best Gens	*St. Dev.*
362.30	101.72
459.20	31.50
400.50	90.46
384.05	88.70
429.80	52.45
485.30	13.19
225.70	112.32
398.40	73.52
475.45	20.68

C			
GA Tested	*El'tm*	*Avg. Best Gens*	*Std. Dev.*
Fixed TGA	1	347.40	60.35
	10	310.90	54.31
Fixed MGA	1	275.55	57.41
	10	237.65	81.90

Table 5.36: **10 variable F7 dynamic mutation convergence times for direct fitness distance (A), direct rank based (B) calculation at 75% chrom. crossover and fixed rate GAs (C).**

Inverse fitness distance calculation showed an improvement over direct fitness distance in most results. Comparison to the ranked based methods revealed inverse fitness distance performing comparably, but indicating greater consistency in a lower standard deviation for elitist results at the lower 10% rate. At higher mutation rates, rank based methods appeared to give slightly better performance on the whole (table 5.37). Average best results were reasonable, with high consistency indicated by a small standard deviation.

A				B		C			
Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Best*	*St. Dev.*	*GA Tested*	*El'tm*	*Average Best*	*Std. Dev.*
10%	0	-4810.95	132.47	-4900.97	79.27	Fixed	1	-4640.83	139.57
	1	-5020.59	11.54	-4891.56	131.14	TGA	10	-4710.48	142.74
	10	-4993.54	61.41	-4930.90	104.66	Fixed	1	-4718.59	172.31
30%	0	-4596.38	70.22	-4973.31	66.67	MGA	10	-4685.29	170.79
	1	-4950.59	50.57	-5025.21	5.54				
	10	-5017.67	25.57	-5009.52	40.53				
50%	0	-4117.96	160.46	-4975.80	38.88				
	1	-4789.00	112.14	-5025.66	0.00				
	10	-4985.61	52.84	-5026.18	3.66				

Table 5.37: 10 variable F7 dynamic mutation optimisation results for inverse fitness distance (A) and inverse rank based (B) calculation at 75% chrom. crossover vs. fixed rate GAs (C).

Best ever performance was also good (table 5.38), producing comparable and in some cases better results than inverse rank based calculation, although the generations taken to converge were again reasonably high. Comparison to the fixed rate GAs was also good, exhibiting comparable performance and in a number of cases improved results - a result similar to that of Srinivas and Patnaik (1994), who also saw improved performance with dynamic rates on highly multimodal problem surfaces.

A		
Max Mtn.	*Degree of Elitism*	*Best Ever*
10%	0	-5016.77
	1	-5029.17
	10	-5029.83
30%	0	-4702.81
	1	-5020.32
	10	-5029.59
50%	0	-4652.17
	1	-4959.91
	10	-5025.99

B
Best Ever
-5025.33
-5028.53
-5028.67
-5027.06
-5029.80
-5029.83
-5006.29
-5029.19
-5026.64

C		
GA Tested	*Degree of Elitism*	*Best Ever*
Fixed	1	-4901.57
TGA	10	-4932.30
Fixed	1	-5020.01
MGA	10	-5016.82

Table 5.38: 10 variable F7 dynamic mutation best ever optimisation results for inverse fitness distance based (A) and inverse rank based (B) calculation at 75% chrom. crossover vs. fixed rate GAs (C)

5.3.4 *10 variable, F8*

Completing the series of tests, the experiments were repeated on the 10 variable F8 surface, with the results for direct and inverse rank based calculation given in appendix C, tables C.29 and C.30 respectively.

Results reflected a number of the trends indicated previously. Both direct and inverse ranking showed comparable performance between 0 and 1 member elitism, whilst repeating the trend of an increase in the number of generations to converge (illustrated in table 5.39 below).

A			
Max Mtn.	*Degree of Elitism*	*Avg. Best Gens*	*St. Dev.*
10%	0	270.90	142.70
	1	401.65	83.01
	10	332.15	104.75
30%	0	281.25	128.99
	1	370.85	99.81
	10	474.80	21.36
50%	0	270.60	139.70
	1	281.45	127.50
	10	395.60	84.36

B	
Avg. Best Gens	*St. Dev.*
238.55	148.65
469.00	34.85
245.50	114.24
295.30	124.73
387.35	74.11
313.90	92.90
241.55	121.35
388.45	88.30
355.55	97.95

Table 5.39: 10 variable F8 dynamic mutation convergence times for direct rank based (A) and inverse rank based (B) calculation at 75% chrom. crossover.

Inverse rank based calculation also showed comparable performance across the rates, but with indications of a marginally improved average best result over that achieved by direct ranking (figure 5.40).

A				B	
Max Mtn.	*Degree of Elitism*	*Average Best*	*St. Dev.*	*Average Best*	*St. Dev.*
10%	0	-180.17	0.05	-180.15	0.09
	1	-180.39	0.00	-180.45	0.00
	10	-180.49	0.00	-180.48	0.00
30%	0	-180.15	0.09	-180.17	0.05
	1	-180.22	0.09	-180.37	0.11
	10	-180.47	0.11	-180.49	0.00
50%	0	-180.11	0.11	-180.16	0.06
	1	-180.13	0.11	-180.31	0.07
	10	-180.39	0.00	-180.49	0.01

Table 5.40: 10 variable F8 dynamic mutation optimisation results for direct rank based(A) and inverse rank based (B) calculation at 75% chrom. crossover.

However, whilst inverse and direct ranking showed similar performance in average best, there was a slight improvement seen in a number of the best ever results for inverse ranking over direct ranking, as shown in table 5.41 below.

A			B
Max Mtn.	*Degree of Elitism*	*Best Ever*	*Best Ever*
10%	0	-180.37	-180.23
	1	-180.47	-180.49
	10	-180.50	-180.50
30%	0	-180.24	-180.28
	1	-180.36	-180.46
	10	-180.49	-180.50
50%	0	-180.25	-180.29
	1	-180.25	-180.40
	10	-180.45	-180.50

Table 5.41: 10 variable F8 dynamic mutation best ever optimisation results for direct rank (A) and inverse rank based (B) calculation at 75% chrom. crossover.

Experiments for direct and inverse fitness distance calculation methods were then performed, with the results shown in appendix C, tables C.31 and C.32 respectively.

Moving to fitness distance based calculation revealed little new information, confirming trends already seen. Average best performance improved as the elitism rate increased in both direct and inverse methods, remaining comparable

between methods, exhibiting a slightly reduced standard deviation for inverse fitness distance (table 5.42). Comparable performance was also indicated between the three mutation rates in both methods.

A				B	
Max Mtn.	*Degree of Elitism*	*Average Best*	*St. Dev.*	*Average Best*	*St. Dev.*
10%	0	-180.07	0.10	-180.16	0.04
	1	-180.37	0.03	-180.35	0.02
	10	-180.41	0.04	-180.49	0.00
30%	0	-180.12	0.14	-180.08	0.00
	1	-180.38	0.10	-180.23	0.08
	10	-180.41	0.03	-180.47	0.00
50%	0	-180.17	0.12	-180.08	0.01
	1	-180.37	0.09	-180.12	0.00
	10	-180.41	0.07	-180.37	0.08

Table 5.42: 10 variable F8 dynamic mutation optimisation results for direct (A) and inverse fitness distance based (B) calculation at 75% chrom. crossover.

The previously noted trend of faster convergence with direct fitness distance was again seen, but had less of an effect than in the F7 surface, mainly due to rapid convergence to the area of F8's global minimum, demonstrated in earlier experiments. The upward effect of convergence times with the introduction of one member elitism was also repeated (illustrated in table 5.43).

A				B	
Max Mtn.	*Degree of Elitism*	*Avg. Best Gens*	*St. Dev.*	*Avg. Best Gens*	*St. Dev.*
10%	0	128.35	75.14	241.45	108.18
	1	168.35	27.59	417.00	55.90
	10	74.50	20.47	358.90	105.89
30%	0	187.15	120.28	245.90	145.05
	1	210.55	30.36	320.65	133.26
	10	73.30	10.85	465.80	20.58
50%	0	192.30	146.00	236.70	138.07
	1	225.85	34.84	223.80	158.92
	10	108.45	104.89	394.55	104.97

Table 5.43: 10 variable F8 dynamic mutation convergence times for direct (A) and inverse fitness distance based (B) calculation at 75% chrom. crossover.

Comparison to rank based calculation showed a slight improvement in average best and, in places, best ever performance for the inverse rank based method over the inverse fitness distance method.

A				
Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Best Ever*
10%	0	-180.16	0.04	-180.36
	1	-180.35	0.02	-180.44
	10	-180.49	0.00	-180.50
30%	0	-180.08	0.00	-180.22
	1	-180.23	0.08	-180.34
	10	-180.47	0.00	-180.49
50%	0	-180.08	0.01	-180.21
	1	-180.12	0.00	-180.28
	10	-180.37	0.08	-180.45

B		
Average Best	*St. Dev.*	*Best Ever*
-180.15	0.09	-180.23
-180.45	0.00	-180.49
-180.48	0.00	-180.50
-180.17	0.05	-180.28
-180.37	0.11	-180.46
-180.49	0.00	-180.50
-180.16	0.06	-180.29
-180.31	0.07	-180.40
-180.49	0.01	-180.50

Table 5.44: 10 variable F8 dynamic mutation average best and best ever performance for inverse fitness distance (A) and inverse rank based (B) calculation at 75% chrom. crossover.

In all other cases, comparable performance was seen. Comparison to the fixed rate tests presented in table 4.8 demonstrated similar, if marginally improved, performance from inverse rank based dynamic mutation at the 10% rate, but in most other cases, the fixed rate GAs performed equally well or slightly better.

A			
GA Tested	*Degree of Elitism*	*Average Best*	*Std. Dev.*
Fixed TGA	1	-180.45	0.00
	10	-180.45	0.08
Fixed MGA	1	-180.47	0.10
	10	-180.47	0.05

B		
Degree of Elitism	*Average Best*	*Std. Dev.*
1	-180.45	0.00
10	-180.48	0.00

Table 5.45: 10 variable F8 performance for fixed rate GAs (A) and 10% dynamic mutation inverse rank based Multi-GA at 75% chrom. crossover (B).

5.3.5 *A summary of dynamic mutation*

The results indicated fairly easy solution of the 2 variable F7 and F8 problems, with little discrimination shown. Increasing mutation rate appeared, as expected, to improve performance although with increasing convergence times as the mutation rate rose. Elitism countered this effect to some degree and again showed a trend of improving performance when introduced in 1 member form. The previously indicated trend of little discrimination between chromosome crossover rates was repeated in early experiments for both 2 variable surfaces, leading to later tests being carried out at a 75% chromosome crossover rate only.

Analysis of the dynamic rate calculation methods revealed a similar trend to that seen in dynamic crossover, with indications of improved performance from inverse calculation methods over direct calculation methods. On the 2 variable F7 surface, inverse rank based calculation indicated a fall in average convergence even at the higher mutation rates of 30% and 50%. Direct fitness distance calculation performed relatively badly compared to its inverse and the rank based methods, all of which indicated improved fitness diversity. The results also compared favourably to the fixed rate GA experiments given in table 4.5, with all dynamic mutation experiments (except direct fitness distance) indicating comparable performance against the fixed rate Multi-GA, but with improved fitness diversity indicated by the greater range of values found for the average best and worst results. The fixed rate traditional GA lost out on these criteria, with average convergence times and standard deviations better in the dynamic mutation Multi-GA experiments. This reflects a similar observation made by Bäck (1992), who noted the use of ES based self adapting mutation rates produced *"very small standard deviation ... [and reached within 1% of optimum] a factor larger than six times faster than the standard GA"* in his experiments.

The F8 surface again showed little discrimination. Overall, with the exception of direct fitness distance calculation which showed a decrease in convergence times with elitism, comparable results across the range of experiments and to the fixed rate traditional and Multi-GAs were seen. The similarity of the results reinforced earlier indications of the simplicity of F8, with most experiments finding the global optimum at one point or another.

The extension to 10 variables in both surfaces verified the trends indicated by the 2 variable results, with inverse rank based calculation indicating a slight improvement over other methods and direct fitness distance showing a rapid convergence to poor results. Comparison to the fixed rate GAs was good, with performance at least comparable and in a number of cases, showing statistically significant improved average best values. Elitism played little part in the results, showing a trend towards increasing convergence time with a slight performance gain in most cases.

These observations present a similar conclusion to that of Starkweather *et al.* (1992), who stated that *"adaptive mutation is critical"* to the improved performance of their distributed GA over and above the traditional GA. Their study also showed an improved time performance with dynamically applied mutation, mirrored here in a number of cases with improved convergence times under dynamic mutation. These similarities may also suggest that Starkweather

et al.'s prediction of similar dynamic mutation behaviour from architectures implementing parallel evaluation of an individual could be correct. Other areas highlighted by Starkweather *et al.*, specifically the relationship of dynamic mutation to a dynamically controlled population size, also warrant further investigation with the Multi-GA and, in particular, its use of addition and deletion operators to dynamically control an individual's chromosomes (explored in chapter 6). As discussed earlier in this chapter, work by authors such as Fogarty (1989) and White & Oppacher (1994) has shown a relationship between dynamic mutation performance and population size, so an extension to examine both dynamic population and individual size in the Multi-GA architecture would be worthwhile.

5.4 Alternative alphabets

As discussed in chapter 2, the issue of alternative alphabetic representations is one that many authors (e.g. Davis, 1991) have investigated with promising conclusions. Chapter 3 outlined the Multi-GA structure and its ability to utilise non-binary feature chromosomes. In studies of non-binary representations carried out previously, it has been pointed out that alternative operators are required, performing the essence of crossover and mutation but in a coherent, non-binary context.

Application problems such as the travelling salesman frequently used order based operators, specific to the problem representation. In the investigations carried out here, the problem surfaces tested with binary chromosomes were also tested with real encoded chromosomes, with specific real valued genetic operators. In addition to providing a performance comparison against the best attempted binary encoded methods, these experiments also served to indicate whether or not incorporation of non-binary chromosomes actually proved of any benefit, with important consequences for the encoding of the problem applications described in chapter 6.

Before beginning experiments, suitable crossover and mutation operators had to be designed. In doing so, the principle of universal applicability was paramount, with the intention of designing crossover and mutation mechanisms suitable for general purpose problem application within real encoded Multi-GA feature chromosomes.

5.4.1 *Real crossover and mutation*

The design of real encoded crossover operators should, as indicated by De Jong (1985), maintain the principles of the genetic metaphor if they intend to be applied to evolutionary algorithms. One common approach has been to produce averaging operators (e.g. Janikow & Michalewicz, 1991) that take the average of two real values and produce a child that is the average of the two. This principle, relating one value to the other, was incorporated in the design of a new real valued crossover operator – Quotient crossover.

The quotient crossover method retains the principle of incorporation of information from the two parents, but produces two children, with alleles related to the relative values of the two parents. Rather than picking a point directly in between as with averaging, the children's alleles are scaled by a quotient value, obtained from the parental chromosomes. The choice of whether to scale the children up or down is made by selecting the fitter or weaker parent as denominator with 50% probability. Consequently, the children will be scaled up or down the range of chromosome values by an amount obtained from the encoded value in each parent. The operation of quotient crossover is illustrated in figure 5.1.

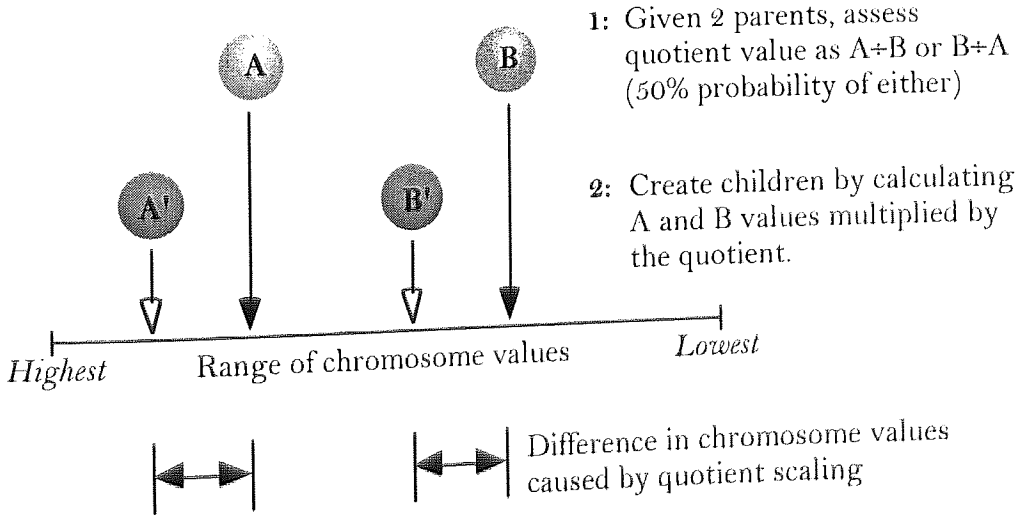


Figure 5.1: Real valued quotient crossover.

Mutation again required changes from binary encoding and a common scheme utilised by other authors (e.g. Williams *et al.*, 1994) was followed, with a small perturbation being made to the selected real value. In the scheme utilised by the Multi-GA, the value used represented a maximum of 10% of the chromosome range, scaled over the zero axis. So, with chromosomes representing a range of

±500, a random number between -25 and +25 would be selected and applied to the chosen real value. The rate at which mutation was applied required no need for change, just the mechanisms by which the selected value was altered.

5.4.2 *Real valued performance*

Having established an additional crossover method, along with averaging and creep mutation, a limited number of experiments were performed on each of the two variants of the F7 and F8 surfaces. As for previous experiments, 20 GA trials were performed for each experiment with the general settings of a 100 member population, executed for 500 generations repeated. The previous results in sections 5.2 and 5.3 led to the choice of experiments using a fixed 75% individual and chromosome crossover, with 4% mutation and dynamic mutation experiments with mutation floating at up to 10%. In all cases, single member elitism was applied. The full results of these experiments on each surface in turn are presented in appendix C, tables C.33 and C.34.

The results again demonstrated the lack of discrimination provided by F8, with both 2 and 10 variable problems producing very similar results. Similar performance was seen from both fixed and dynamic mutation rates and quotient and averaging methods of rate calculation (table 5.46).

A				B	
Surface	*Xover*	*Average Best*	*St. Dev.*	*Average Best*	*St. Dev.*
F8:2	Quot.	-180.50	0.03	-180.50	0.00
	Avg.	-180.50	0.06	-180.50	0.08
F8:10	Quot.	-180.50	0.00	-180.50	0.07
	Avg.	-180.49	0.00	-180.49	0.00

Table 5.46: F8 optimisation results for real valued Multi-GA with fixed 4%(A) and dynamic 10%(B) mutation rates using both quotient and averaging crossover.

F7 proved to be more discriminating, with the 2 variable surface indicating a number of trends that were greatly exaggerated in the 10 variable case. Averaging crossover proved, in all cases, to be of particularly poor performance, failing to achieve the global optimum in the 2 variable case (-1677.97) and falling significantly short in the 10 variable problem (-5029.83). Quotient crossover performed better, with extremely good performance on the 2 variable F7 surface, finding the global optimum in every fixed rate case. With dynamic mutation, average performance fell slightly but still outperformed averaging crossover for both fixed and dynamic mutation rates. In the 10 variable case, quotient crossover again outperformed averaging crossover at both fixed and

dynamic rates, but failed to provide good performance in terms of optima reached. These results are illustrated in table 5.47 below.

A				B	
Surface	*Xover*	*Average Best*	*St. Dev.*	*Average Best*	*St. Dev.*
F7:2	Quot.	-1677.97	0.00	-1672.05	25.80
	Avg.	-1586.70	77.29	-1536.61	63.84
F7:10	Quot.	-4204.52	170.28	-4272.91	170.89
	Avg.	-2982.22	388.15	-2446.69	228.81

Table 5.47: F7 optimisation results for real valued Multi-GA with fixed 4%(A) and dynamic 10%(B) mutation rates using both quotient and averaging crossover.

Averaging crossover showed an extremely fast convergence time of less than 100 generations for the F7 problems studied, in both dynamic and fixed rate mutation experiments. The poor results seen here suggest that premature convergence is taking place, forcing an evidently suboptimal solution. Quotient crossover showed convergence times similar to the fixed rate binary GAs. These conclusions are illustrated by table 5.47 below.

A				B		C		
Surface	*Xover*	*Avg. Best Gens*	*St. Dev.*	*Avg. Best Gens*	*St. Dev.*	*GA Tested*	*Avg. Best Gens*	*St. Dev.*
F7:2	Quot.	306.30	114.31	348.45	89.19	TGA	61.60	38.46
	Avg.	70.95	114.55	96.70	149.22	MGA	30.95	5.72
F7:10	Quot.	443.50	86.41	410.75	86.29	TGA	347.40	60.35
	Avg.	56.90	91.88	91.85	156.68	MGA	275.55	57.41

Table 5.48: F7 convergence times for real valued Multi-GA with fixed 4%(A) and dynamic 10%(B) mutation rates using both quotient and averaging crossover, plus fixed rate binary GAs (C).

Whilst converging in a comparable time to binary encoded experiments performed earlier, the averaging real encoded method fell well below the performance of the binary GAs.

A			
Surface	*Xover*	*Average Best*	*Std. Dev.*
F7:2	Quot.	-1677.97	0.00
	Avg.	-1586.70	77.29
F7:10	Quot.	-4204.52	170.28
	Avg.	-2982.22	388.15

B	
Average Best	*St. Dev.*
-1672.05	25.80
-1536.61	63.84
-4272.91	170.89
-2446.69	228.81

C		
GA Tested	*Average Best*	*St. Dev.*
TGA	-1677.79	0.25
MGA	-1677.58	0.93
TGA	-4640.83	139.57
MGA	-4718.59	172.31

Table 5.49: **F7 optimisation results for real valued Multi-GA with fixed 4%(A) and dynamic 10%(B) mutation rates using both quotient and averaging crossover, plus fixed rate binary GAs (C).**

This was seen in both average (table 5.49) and best ever (table 5.50) values. Quotient crossover put up a better performance, but still failed to match the performance of the binary GAs in most cases.

A		
Surface	*Xover*	*Best Ever*
F7:2	Quot.	-1677.97
	Avg.	-1674.79
F7:10	Quot.	-4541.30
	Avg.	-3288.49

B
Best Ever
-1677.97
-1657.34
-4595.01
-3053.80

C	
GA Tested	*Best Ever*
TGA	-1677.97
MGA	-1677.97
TGA	-4901.57
MGA	-5020.01

Table 5.50: **F7 best ever optimisation results for real valued Multi-GA with fixed 4%(A) and dynamic 10%(B) mutation rates using both quotient and averaging crossover, plus fixed rate binary GAs (C).**

F8 provided little discrimination, finding the global optimum in most cases. The trend for fast convergence was again illustrated in the F8 2 variable problem, where the global optimum was located in all cases, dynamic quotient crossover finding it in less than 25 generations! However, the trend for faster convergence with averaging crossover was reversed, indicating longer convergence times on average. In all other cases, no real distinction was seen between methods applied on F8.

A			
Surface	*Xover*	*Avg. Best Gens*	*St. Dev.*
F8:2	Quot.	70.95	123.02
	Avg.	106.70	90.86
F8:10	Quot.	427.05	86.02
	Avg.	469.05	32.06

B	
Avg. Best Gens	*St. Dev.*
15.45	7.79
206.15	129.58
416.45	81.41
464.05	42.42

Table 5.51: **F8 convergence times for real valued Multi-GA with fixed 4%(A) and dynamic 10%(B) mutation rates using both quotient and averaging crossover.**

5.4.3 *A summary of real encoded experiments*

In summary, the real valued experiments served to demonstrate the ability of the Multi-GA to work with non-binary encodings. Quotient crossover, providing a scaling factor based on parental information and used to create real encoded children was introduced, with results indicating improved performance over an averaging based method. Trends illustrated in previous experiments were repeated, with F7 providing results confirming trends indicated by binary encoded experiments. F8 again showed little discrimination. Whilst indicating better performance from quotient crossover, results in the discriminating 10 variable F7 problem indicated poorer performance from real encoding compared to binary encoding for the experiments performed here. However, the purpose of the experiments – to illustrate the successful application and methods for manipulation of non-binary chromosome representations – was fulfilled.

5.5 Recent discoveries concerning F8

One of the repeating concerns arising throughout the test series has been the lack of discrimination shown by the test results carried out on the F8 surface, in both 2 and 10 variable form. Whilst F7 showed reasonably effective discriminating behaviour, especially in the more complex 10 variable form, F8 has indicated little discrimination. Although the surface maintains an overall basin shape, visual analysis (figure 4.5) suggests a complex surface with an extremely large number of hills and valleys. In addition, recommendations from Mühlenbein and his PGA studies (1991) both indicated F8 as a good choice of scalable test surface, suggesting it to be a difficult problem surface to optimise due to its complex shape and scalable nature.

However, a recent paper (Whitley *et al.*, 1995) presented at the 6th International Conference on Genetic Algorithms throws considerable light onto the results obtained in the experiments performed here, opening a debate into the complexity of test suite surfaces, including specific mention of F8. The study carried out by Whitley *et al.* (1995) investigated common optimisation surfaces F1 - F10 in detail, with interesting results. Their analysis of F8 revealed surface behaviour that goes a long way to explaining the ease of solution seen in many of the experiments presented in the last two chapters. In addition to presenting arguments showing that Mühlenbein's (1993) Breeder GA can be outperformed by other methods, the study also presented a theoretical illustration of why F8 is of little use in providing discriminating GA behaviour. By plotting the n dimensional surface as 1-D slices along the diagonal of the hyperplane, Whitley

et al. (1995) were able to provide cross-sectional analysis of the F8 surface in both 1 and 10 dimensions.

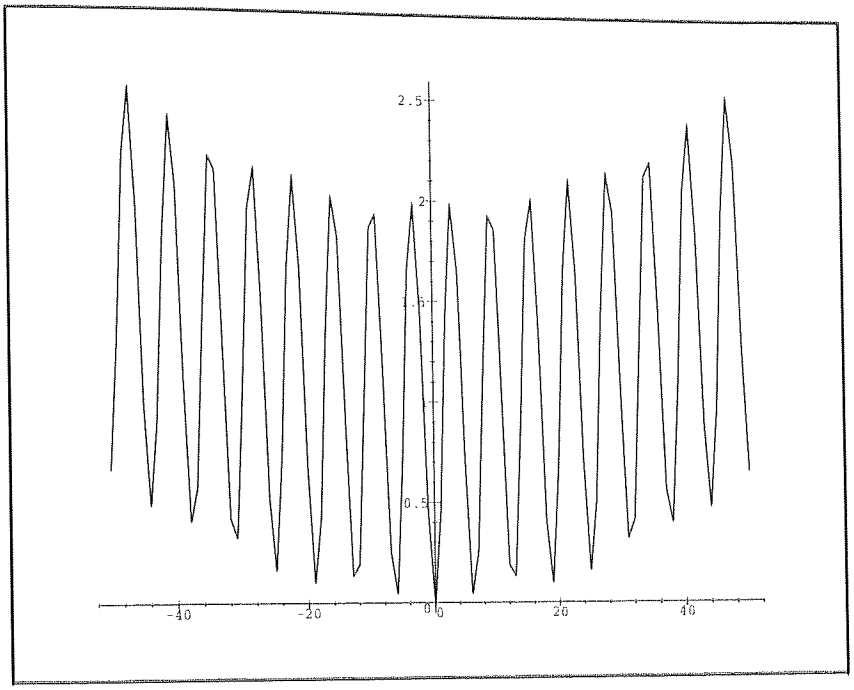


Figure 5.2: Cross section of the 1 variable F8 surface, given by Whitley *et al.* (1995), shows multiple peaks and valleys.

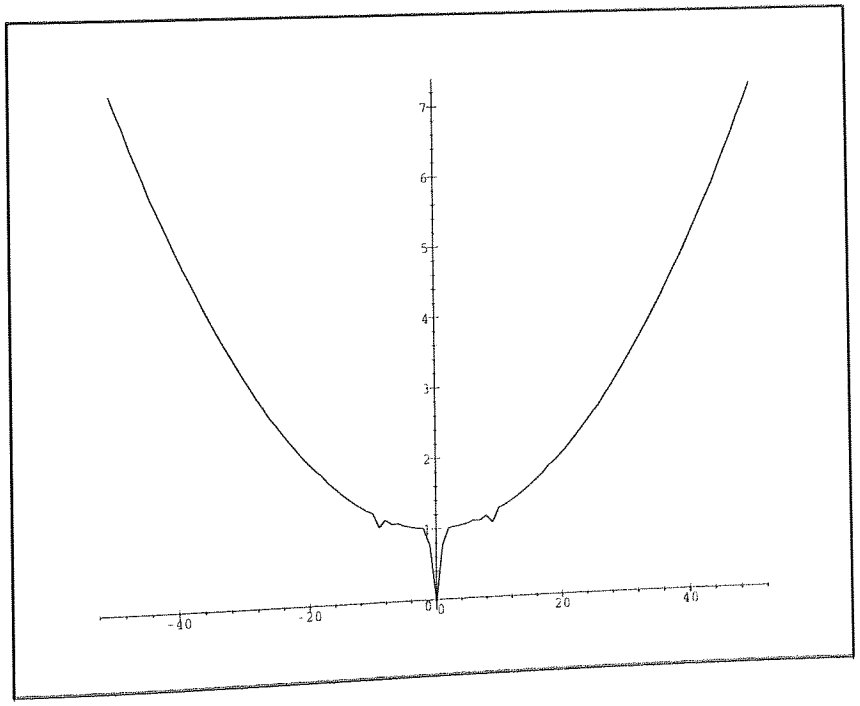


Figure 5.3: Cross section of the 10 variable F8 surface, given by Whitley *et al.* (1995) shows a much simpler parabola than the 1 variable case.

Examination of the results (reproduced in figures 5.2 and 5.3 with kind permission of Soraya Rana) showed a 1 variable F8 cross section clearly reflecting the apparently complex nature of the surface seen in figure 4.5. However, the 10 variable cross section shows the surface to *decrease* in complexity with increasing dimensionality! This reduction in complexity was identified by Whitley *et al.* (1995) as due to the decreasing influence of the product term, leading to a reduction in size of the local minima that Mühlenbein (1991) identified as giving the surface its complexity. Their conclusion was that *"the use of these test functions (F1 - F10) in comparative studies may lead to suspect conclusions."*

Taking this new information into account, the reason for a lack of increased discrimination at higher dimensionality in the earlier experiments becomes clear. In addition, the reasons for Mühlenbein's (1991) reports of good performance can also be understood when one recalls that his PGA (described in section 2.5.2.2.) was hybridised with a hill climbing algorithm that engaged after a specified period of time, rather than maintaining a purely genetic approach. Visual examination of the surface clearly indicates why hill climbers would find this problem difficult, with extremely large numbers of local maxima in between the next (local) minimum.

However, the tests on F8 carried out here, although lacking experimental significance, reinforce the results of experiments in which individual level crossover was removed. In addition, indicative results were also given, in line with the F7 2 variable problem, that illustrated trends shown with statistical significance in the F7 10 variable surfaces. Whilst the recent information provided by Whitley *et al.* (1995) gives much insight into the results obtained for F8, the studies performed here were not without merit, providing a widely different problem surface on which useful experiments were conducted.

Chapter 6: Applications

6.1 Introduction

The results of the tests performed in chapters 4 and 5 illustrated the potential of the Multi-GA and its comparable performance to traditional GA methods. However, as explained in section 4.2.3, direct application in a traditional GA manner restricts full utilisation of the Multi-GA structure. In order to fully explore the concepts laid out in chapter 3, applications problems with a more multi-dimensional nature were investigated. This chapter describes the application of the Multi-GA to two types of problem – network optimisation and spatial analysis, demonstrating the use of feature chromosomes of different type representations, chromosome feature functions and analysis of new genetic operators developed.

6.2 Network machine placement

Chapter 3 outlined the potential for the Multi-GA structure to use chromosomes of different type representations. The network machine placement problem was used to demonstrate this potential in an applications context.

6.2.1 *Related research*

The network placement problem is similar in its nature to those in a wider class of problem currently investigated by the GA community – namely, the facility layout problem (FLP). Facility layout, being a type of problem involving large search spaces and therefore many possible solutions, has been the focus of mathematical algorithms for some time. The extent of this research can be seen by analysis of literature in the field, summarised effectively in papers by Tam (1992) and Kusiak & Heragu (1987). Their surveys identify a number of different mathematically based approaches to tackling FLPs, dating back to work by Armour and Buffa (1963). Naturally, with developments in computing and artificial intelligence techniques, new techniques have been continually applied in an attempt to improve the quality of solutions, with GAs being a recent addition.

A common approach to FLPs by the GA community is that identified by Tam (1992) in the use of a slicing tree structure - a binary tree defining the order of operations that make up the layout required for optimisation - encoded into a GA operator string. Allowing a direct, legal representation of valid slicing tree structures, the traditional GA can then be applied to generate potential

solutions. Indications from studies such as Tam (1992) and Kado *et al.* (1995) demonstrate the success of the GA in this area, compared to existing mathematical techiques and newer approaches such as simulated annealing (Kirkpatrick *et al.*, 1983).

As with many other application areas, as mentioned in chapter 2, further research has taken place into the actual GA processes used to refine the GA's solution attempts. Tam's (1992) approach used only a simple genetic process with little optimisation and parameter tuning, yet still produced good results. Other authors, such as Tate & Smith (1993b) have taken this research further, investigating variations of the FLP and suggesting the use of interesting techniques such as dynamic penalty functions to better attack more complex FLPs – in their case, the inclusion of infeasible regions of search space. The scope for further work remains open, with Tam (1992) identifying possible benefit from ideas such as the *"maintaining a list of good solutions encountered so far,"* the freezing out of operators on *"parts of a layout (i.e. subtrees) [which] are good enough and need not be changed"* and *"reducing the crossover and mutation rate if existing solutions are acceptable."*

6.2.2 *Application of the Multi-GA to network placement*

As discussed earlier in this thesis, it is believed that the Multi-GA structure laid out in chapter 3 draws together a number of current approaches that may provide benefit when dealing with multi-objective optimisation problems; particularly those which may benefit from more flexible representation or finer parameter control for independent problem sections. The conclusions into the FLP study by Tam (1992) suggest a number of areas that have direct relevance to the Multi-GA structure investigated here. In particular, his observation of potential benefit from a finer parameter control, the reduction of crossover and mutation rates on particular sections of the problem currently well solved and the 'freezing out' of good problem subtrees are all areas identified in chapter 7 as particularly easy to implement within the framework of the Multi-GA structure.

The particular problem selected here for Multi-GA application, namely network placement, did not attempt to directly compare Multi-GA performance to that of the traditional GA approaches, which it is acknowledged perform particularly well in comparison to other methods on the class of problems represented by FLPs. The intention of this particular problem was to demonstrate a mechanism by which the Multi-GA structure – identified in previous chapters as having the potential for handling a mixture of different type representations

within a single structure – could be successfully applied to an optimisation problem with real world applicability that may contain different type representations. By selecting a problem closely related to an area of current GA investigation, it can be seen that the ideas emerging throughout different areas of the GA field - drawn together by the Multi-GA structure used here - provide for direct applicability to an area in which GAs are producing increasingly better solutions with time. In addition, the further investigation into the structural potential and genetic self-modification operators undertaken later in this chapter go some way towards identifying a mechanism by which the Multi-GA structure could indeed be directly applied to the FLP, addressing exactly those research issues raised by Tam (1992).

6.2.3 *Problem description*

In the real world, a common requirement in many organisations is to efficiently install a number of computers in an ethernet style network. Undertaking this task, the business would usually wish to minimise the distance between machines, with some machines possibly restricted in their location (e.g. to certain floors of a building). This task, involving different types to represent the building, was given to the Multi-GA. A hypothetical building was created, with real valued x and y co-ordinates defining the position of a machine within each floor. The floor level position of a machine was defined by a binary encoded z chromosome. A large search space was created, with the overall building dimensions being (0..20,000), (0..30,000) and (0..120) for the x, y and z chromosomes respectively.

Two problems were tested, varying the restrictions and number of machines placed within the building. The precise locations and ranges of the machines for the 8 and 15 machine placement problems are given in appendix A, tables A.1 and A.2 respectively. A graphical illustration of the 8 machine problem is given in figure 6.1, with the 15 machine problem displayed in appendix A, figure A.1.

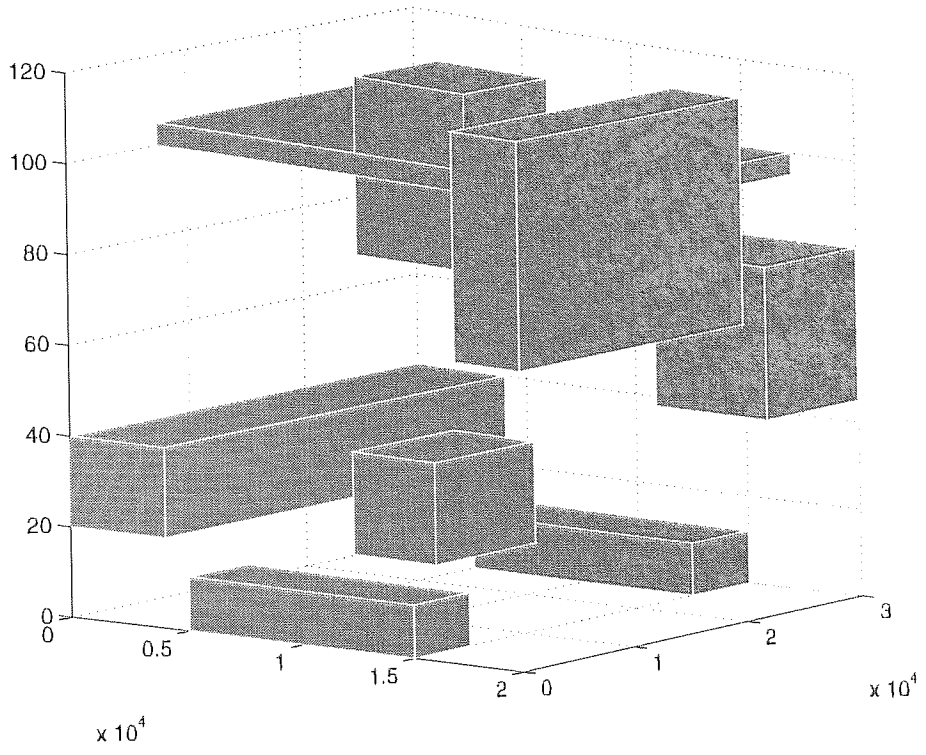


Figure 6.1: Legal machine distributions for the 8 machine location problem.

6.2.4 *Mixed representation encoding in the Multi-GA*

The combination of mixed representation chromosomes was achieved by representation of the integer valued floor level (z co-ordinate) as a binary encoded chromosome, forming the first chromosome in a triple representing each machine. The remaining two chromosomes represented the real valued x and y ranges within the floor specified by the z chromosome.

A number of machines were defined by each problem, with each machine being represented by the three chromosomes for z, x and y co-ordinates. Each Multi-GA individual consisted of a number of these chromosome triples, so for the 8 machine case the individual contained 24 chromosomes with the first chromosome in each triple taking a binary representation amongst the remaining real represented chromosomes, as illustrated in figure 6.2.

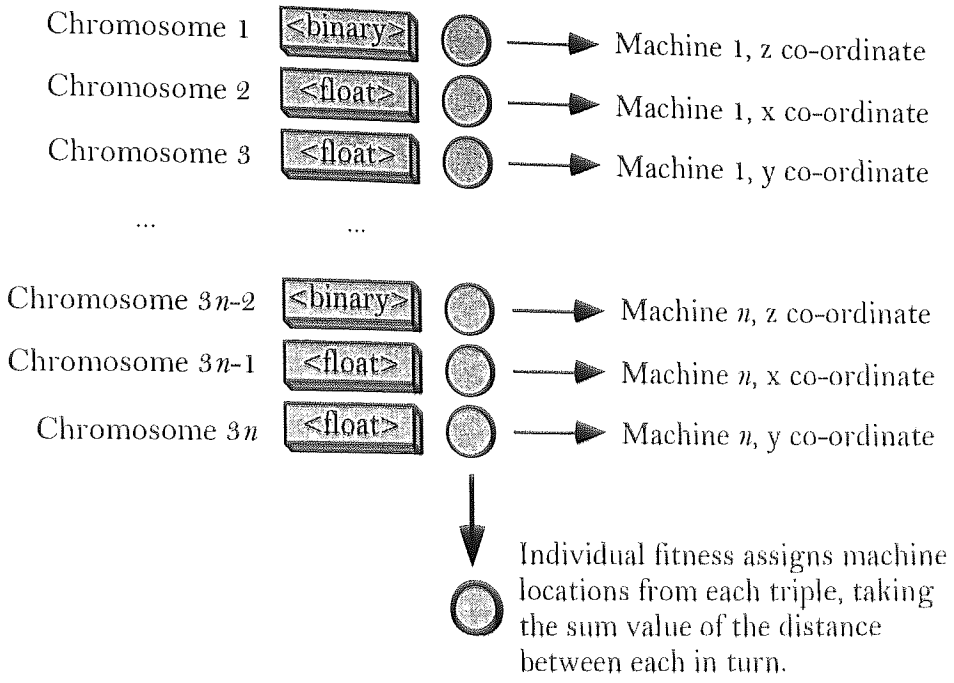


Figure 6.2: Machine location encoding of chromosomes in a Multi-GA individual.

Interpretation of the chromosome values was performed by the individual level fitness function, taking each triple in turn and translating it into a point in the 3D building. The overall fitness of the individual was calculated by the distance between each machine in sequence, with the objective being to minimise the total distance (and hence cabling required in a real world problem).

This was achieved by combining the chromosome fitness values, which contained a penalty function applied to the chromosome where distinct range limits were specified. The chromosome fitness function imposed on each x, y or z co-ordinate an additional penalty of d^2, where d represents the distance outside the legal co-ordinate range generated by the chromosome. In a minimisation problem, this was seen to effectively penalise chromosome values the further away from the legal region they drifted. The use of this fitness penalisation mechanism was particularly appropriate given the conclusions drawn by Tate & Smith (1993b), namely that in a problem of this type the best solutions frequently lie on the boundaries of the feasible regions. In addition, it further demonstrated the structural flexibility of the Multi-GA, illustrating the use of a chromosome fitness function which could potentially differ from chromosome to chromosome as appropriately required (say on the real valued x and y co-ordinate chromosomes, but not the integer valued z co-ordinate, for example).

6.2.5 Experimental results

Experiments were performed, as described, on 8 and 15 machine location problems. As with previous tests, 20 GA trials were performed for each experiment with a population size of 20. In the 8 machine problem, whilst some machines were restricted to narrower ranges, there were no fixed chromosomes. In the 15 machine problem, a number of machines were restricted to single floors, as detailed in appendix A, table A.2. The results for both problems are presented in turn.

6.2.5.1 8 Machine problem

The results for the simpler 8 machine problem were promising, with the Multi-GA producing good results. Tables 6.1a and b show the results obtained from the average of the 20 trials and then the specifics of the best ever solution obtained, illustrated in graphical form in figure 6.3.

Average Best	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Best Ever*	*Best Ever Gens.*	*Avg. Best Gen.*	*Std. Dev.*
37962.40	99.03	141068.00	7314.40	37924.3	445.00	449.40	53.50

Machine	*z*	*x*	*y*	*Machine*	*z*	*x*	*y*
1	1	5000.00	5000.00	5	100	17000.00	20000.00
2	22	4000.00	6070.66	6	60	15718.41	22000.00
3	81	7921.12	10204.33	7	15	12200.00	16500.00
4	105	13331.54	16101.24	8	4	12174.59	25000.00

Tables 6.1a and b: Results of the 8 machine network minimisation.

The results show the sum of the distances between each machine, with the fittest member obtaining the minimum possible distance. The results were good, with the average worst value showing a wide range of values found during the Multi-GA search. Examination of the graphical output of the results, given in 3D in figure 6.3 (and in 2D views in appendix A, figures A2 and A3) show the Multi-GA to have located a highly acceptable solution, with no problems caused by use of multi-representational chromosomes within the individual.

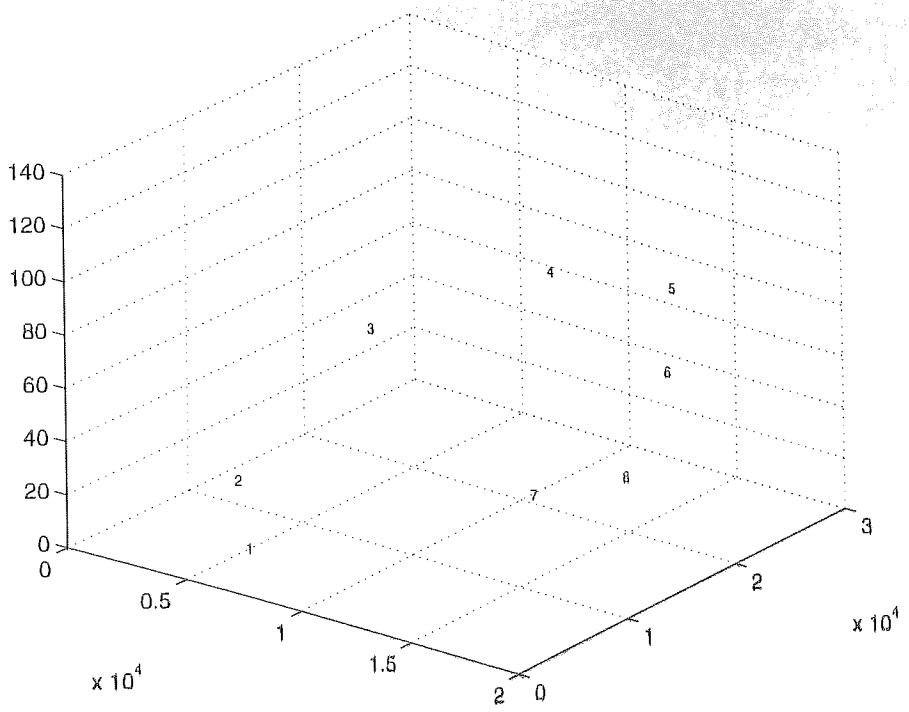


Figure 6.3: 3D illustration of 8 machine best result locations.

The 3D view demonstrates the relationship between each of the machines found by the Multi-GA solution. Examination of the results presented in appendix A, figures A2 and A3 shows the extent to which the Multi-GA has positioned many machines on the very limits of their legal ranges in order to minimise the distance to the next machine in the sequence. In addition, the Multi-GA converges to a good solution in under 500 generations for a fairly small population size. As such, no indication is given of unacceptable or problematic behaviour through the use of independently represented chromosomes within Multi-GA individuals.

6.2.5.2 *15 Machine problem*

Following experiments on the 8 machine placement problem, with the Multi-GA showing good performance with representationally diverse individuals, further complexity was introduced by increasing the number of machines and restricting the legal positioning of the machines. Consequently, the number of chromosomes increased to 45 per individual, presenting the Multi-GA with significantly more information to handle. The results are presented in table 6.2a averaged over the 20 trials, with table 6.2b showing the machine co-ordinates determined by the best run.

Average Best	*Std. Dev.*	*Average Worst*	*Std. Dev.*	*Best Ever*	*Best Ever Gens.*	*Avg. Best Gen.*	*Std. Dev.*
93350.20	4775.81	230127.00	8840.79	91506.90	353.00	450.15	37.81

Machine	*z*	*x*	*y*	*Machine*	*z*	*x*	*y*
1	96	4976.37	9915.86	9	38	8838.72	2000.00
2	86	5000.00	10000.00	10	120	10000.00	10000.00
3	86	7378.75	18000.00	11	91	13104.22	7000.00
4	86	8786.70	20000.00	12	10	15000.00	10000.00
5	86	15000.00	10000.00	13	60	17000.00	12352.37
6	8	2500.00	12349.74	14	60	8000.00	27000.00
7	65	7500.00	12800.00	15	33	12000.00	27000.00
8	0	8453.06	5000.00	–	–	–	–

Tables 6.2a and b: Results of the 15 machine network minimisation.

The results show similar trends to those indicated in the 8 machine problem, with many of the locations found to be on the borders of the acceptable regions in an attempt to reduce the distance as much as possible. As with the previous experiment, a high range between average best and worst indicates the degree to which the GA has searched the problem space, converging in a similar number of generations to a good solution. A more instructive 3D illustration of the solution found is given in figure 6.4

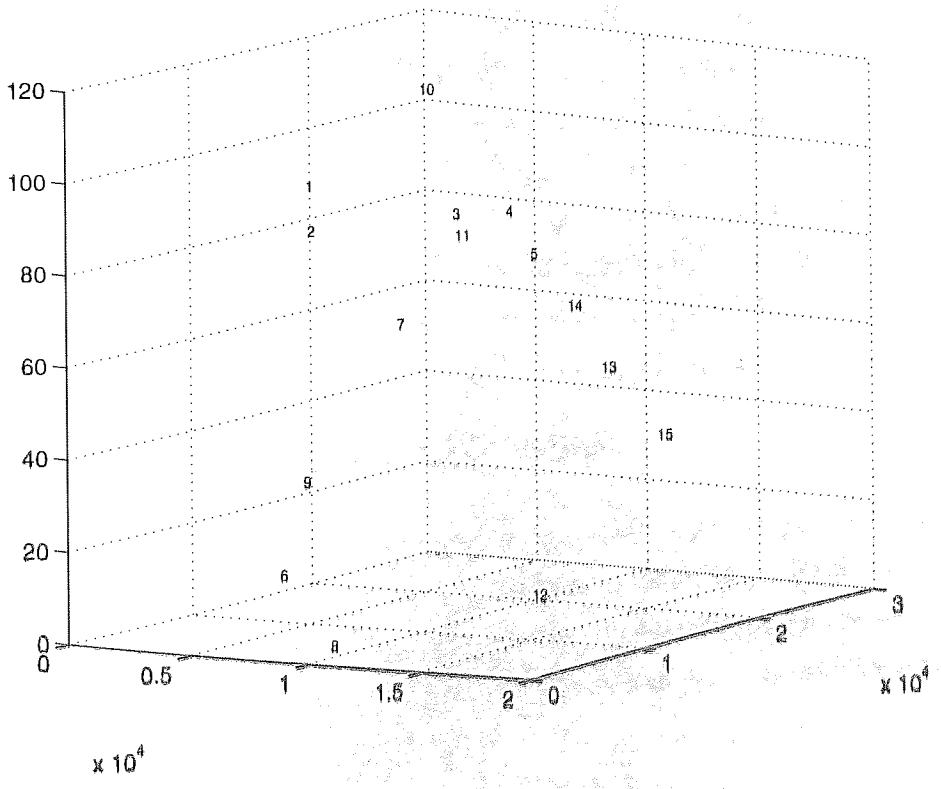


Figure 6.4: 3D illustration of 15 machine best result locations.

As with the 8 machine problem, an insight can also be gained into the precise location of each machine with respect to its legal ranges by examination in 2D along vertical and horizontal planes, given in appendix A, figures A4 and A5. Once more, the graphical results show that the Multi-GA has found a good solution to this more complex problem, consistently locating the solution within 500 generations.

6.2.6 *Summary of Multi-GA application to network placement*

The network machine placement problem was applied in order to demonstrate the ability of the Multi-GA to successfully tackle an optimisation task involving chromosomes of different representations within a single individual. Application to 8 and 15 machine placement problems was carried out, using a mixed set of chromosome triples consisting of binary and real encoded chromosomes defining the location of each machine. In addition, the ability of the newly introduced quotient crossover to effectively manipulate schemata was tested, with the real encoded chromosomes being manipulated only by this mechanism.

The results showed that the Multi-GA was able to produce good, consistent solutions to both problems, within 500 generations for a relatively small population size of 20. The differences of representation appeared to provide no obstacle, with the Multi-GA easily applying binary and quotient crossover to the relevant chromosomes within each individual. Whilst by no means providing an exhaustive or comprehensive test suite on this particular problem, the intention of illustrating successful application of a mixed chromosome representation, using different chromosome crossover mechanisms for the different types was successfully achieved, with good results. These results provide a positive basis for further investigations of the Multi-GA structure, particularly with application to those areas mentioned by researchers such as Tam (1992) for which the structure provides a natural operational mechanism.

6.3 Spatial analysis and the site selection problem

Spatial analysis is a common goal of Geographical Information System (GIS) based techniques, typically utilising GIS information hybridised with other search mechanisms to provide spatial (and other) information. However, current GIS methods suffer from a number of limitations and recent research has endeavoured to tackle these difficulties.

The following sections describe GIS, current approaches and the application of the Multi-GA as a new approach, under the following headings:

- Geographical Information Systems and spatial analysis
- Limitations of GIS
- Spatial decision support systems
- A neural network approach
- Applying the Multi-GA

6.3.1 *Geographical Information Systems and spatial analysis*

The GIS field is an area of diverse research and application, making it difficult to obtain a single, all-encompassing definition. In general, GIS are a class of information systems relating to the storage, display, presentation and analysis of geographically related data, which have found particular benefit from the use of advanced computing technology. The interested reader is referred to Maguire *et al.* (1991) for an excellent introduction to the GIS field. GIS provide a wealth of information about the geographical and human environments, with application in areas from provision of postal data (e.g. Ordnance Survey's ADDRESS-POINT system; O.S., 1995) through to health service provision, defence planning and local government administration (Chorley & Buxton, 1991).

In particular, the latter applications illustrate the particular use of GIS in spatial analysis, defined by Johnston *et al.* (1986, p.446) to be *"quantitative (mainly statistical) procedures and techniques applied in local analytical work."* Spatial analysis has been identified as *"extremely relevant to GIS"* with *"the gradual absorption of spatial analysis tools into GIS [seen as] inevitable"* (Openshaw, 1991, p. 389). These observations are confirmed by examining the real-world applications of GIS. Chorley & Buxton (1991, p.76) described how GIS have been applied in the provision of health service care, providing information on the distribution and characteristics of patients and health care facilities, subsequently used to determine funding allocations. In local government, the Chorley Report (DoE, 1987) identified spatial information as useful for a number of forecasting requirements, including local community services, land usage and resource management. Chorley & Buxton (1991, p.78) have also identified the future potential of spatial analysis, essentially interconnected with GIS, noting that *"the potential for its more widespread use and exploitation are increasingly becoming recognised."*

6.3.1.1 Limitations of GIS

Despite the widespread and successful use of GIS as information provision systems, they suffer from a number of difficulties – specifically in their ability to provide effective data analysis. This issue was highlighted by Aangeenbrug (1991), who pointed out a *"need to pay more attention to spatial analysis."* Openshaw (1991, p.389) presented a starker analysis, stating that *"the existing spatial analytical toolbox is largely inadequate"*. This analytical failing of the GIS is repeatedly identified throughout Maguire *et al.* (1991) gaining recognition from funding authorities that has led to investment in spatial analysis research, particularly in areas like statistical analysis and neural networks.

It is the lack of ability to intelligently interrogate the vast source of data provided by GIS that currently handicaps their descriptive power, in particular the inability to provide data extrapolation for a wide range of planning applications. Indeed, Openshaw (1991, p. 400) identified *"the previous neglect of spatial analysis [as] a major impediment to the full exploitation of GIS."* However, some attempts to resolve these weaknesses have been undertaken, with spatial decision support systems mentioned by Openshaw (1991, p. 391) as an approach illustrating developments in *"analysis for purposes of decision support and spatial planning"*.

6.3.1.2 Spatial decision support systems

Spatial Decision Support Systems (SDSS) seek to address the *"complex spatial problem often [having] multiple, conflicting objectives for its solution"* (Densham, 1991, p.403). Based upon similar principles to current AI approaches into Decision Support Systems, SDSS attempt to augment their supportive GIS, providing systems *"explicitly designed to provide the user with a decision making environment that enables the analysis of geographical information to be carried out in a flexible manner"* (Densham, 1991, p.405). Specific objectives of traditional decision support systems include to *"help the user explore the solution space by using the models in the system to generate a series of feasible alternatives"* (Geffrion, 1988). These basic principles are mirrored in the SDSS, but with additions that take account of the spatial nature of the data. However, the power of current approaches is limited, confined to traditional artificial intelligence and database query style information systems. Some attempts have been made to incorporate expert system type approaches, but the current scope of SDSS remains restricted.

6.3.1.3 A neural network approach

Recent research approaches have included neural networks to provide pattern classification analysis. Queries of the kind typically demanded by GIS were identified by Densham (1991), who used the example of a bank branch location problem, involving the siting of a particular facility based upon the surrounding population. Typical questions asked might be "how many branches should there be?" and "where should I locate these branches?" The limited analytical power of current GIS and SDSS reduces their ability to provide satisfactory answers to such queries.

One interesting new approach is that taken by Murnion (1995), who applied a Hopfield network to this class of problem – specifically, retail site location. Murnion's approach attempted to answer the question "how many stores of a minimum profitability can be supported by a given population grid?" – a question that current GIS find difficult to answer. The results were interesting, showing good performance from the Hopfield net, indicating coverage of the population surface and location of the optimum number of sites for a variety of profitability levels.

However, more advanced questions concerning maximisation of population coverage in a minimum number of sites were not addressed by the Hopfield network, being identified as difficult to implement (Murnion, 1995a).

6.3.1.4 Related GA research

Although the use of GAs as a technique for GIS is a novel one, the general class of problem presented by site selection is similar to those covered by timetabling and scheduling - areas already attracting significant GA research. GAs have been identified (as described in chapter 2) as optimisation algorithms with high potential for tackling objective and multi-objective optimisation. Areas such as timetabling or scheduling, which involve the calculation of a sequence of items from a potentially vast problem space, have found themselves to benefit particularly from GA application.

Abramson's (1992) paper is a good example, illustrating the successful application of a GA to school timetabling. Working with a chromosome string of tuples representing teachers, classes and room allocations, Abramson's GA calculates a combination according to the objective of minimising resource conflicts. This approach, of encoding linear programming optimisation constraints into a fitness function and formulating an ordering (or other

problem specific) operator to perform genetic search on a single chromosome string, has been applied in a similar way by other authors (e.g. Abela *et al.*, 1993; Burke *et al.*, 1995; Bagchi *et al.*, 1991). Studies such as these have served to demonstrate the effectiveness of GAs as an alternative to the widely accepted and tried artificial intelligence approaches to scheduling and linear constraint optimisation. This effectiveness is further reinforced by positive comparative results, such as those obtained in studies of the type performed by Abela *et al.*, 1993 and Easton & Mansour, 1995 (for example).

There are a wide variety of scheduling applications and approaches, particularly in the real-world context, from which GAs have been able to draw inspiration and potential solutions. As such, the application of GAs to scheduling style problems is not a new phenomenon, with a wide variety of research having been carried out from typical problem areas such as knapsack (e.g. Khuri *et al.*, 1994) and job shop scheduling (e.g. Bachi *et al*, 1991; Nakano & Yamada, 1991; Biegel & Davern, 1990; Davis, 1985) through to more problem specific direct application (e.g. Langdon, 1995; Abela *et al.*, 1993; Abramson *et al.*, 1993). As with timetabling, many GA strategies have found much success in this class of problem. The work presented here demonstrates the performance of the Multi-GA architecture on a class of problem in which its structural flexibility gives it a distinct applicability.

6.3.1.5 *Applying the Multi-GA*

The intention of Multi-GA application to the GIS problem is twofold. Firstly, it serves to demonstrate to researchers in the GIS field that the increasingly widely accepted metaphor of evolutionary computation may well have direct application and solve a number of the concerns of data management identified in GIS literature.

From the perspective of GA research, the GIS site selection problem provides a highly appropriate vehicle for study and analysis of the dynamics of some of the mechanisms implemented in the Multi-GA structure outlined in chapter 3. With a general trend of much GA research towards structural adaptation that more naturally reflects a multi-dimensional or multi-objective problem, the identified ability of a distinct multi-chromosome approach to self-adapt would be useful to explore. In addition, the GIS problem provides a mechanism for studying the effects of the self-adaptation mechanisms identified earlier in the thesis, bringing an additional level of understanding into the dynamics of the genetically controlled structural adaptation. With the Multi-GA approach

providing a co-ordination and refocusing of many currently active research strategies into a single genetic structure, the results of a study into this particular mechanism for self-adaptation may provide lessons or ideas for those researching alternative structural mechanisms to the traditional linear GA. The GIS application also allows a greater exploration of the potential flexibility of the Multi-GA discussed in chapter 3 and not exhibited by the comparative studies presented in chapters 4 and 5. The site selection problem, as implemented by Murnion (1992), provides a problem which inherently contains an element of potential self-adaptation and is therefore appropriate to this particular facet of the Multi-GA. The resulting analysis of self-control of genetic material involved in obtaining the Multi-GA solutions to GIS site selection, and the associated study of the behaviour of the genetic modification mechanisms proposed here, are presented in the remainder of this chapter.

It is accepted that much work is still to be done and a number of problems could have been attempted within this particular class, geared towards exploring other Multi-GA concepts that hold potential. For example, a comparison with subpopulationary evolution on a timetabling problem (discussed in more detail in chapter 7) would provide a valuable insight into the possibility of a benefit being gained from the evaluation of independent fitness contributions. In addition, time constraints did not allow for a full and comparative study of a traditional GA adapted to work on this particular problem.

Densham (1991) and Geffrion's (1993) descriptions of spatial analysis identified a classical GA optimisation problem, which it was felt would benefit from Multi-GA application. In particular, the increased flexibility of representation and the use of extended genetic operators provided by the Multi-GA structure were seen to provide the potential for extending the questions tackled by Murnion (1995) to an even more advanced state. Using Murnion's data, the Multi-GA was applied in order to demonstrate a number of key objectives:

- The use of distinct chromosome and individual level fitness functions within the Multi-GA.
- The effects of automatic manipulation of genetic material through the use of extended genetic operators, addition and deletion.
- The applicability of the Multi-GA to the analysis of GIS data, establishing the potential for future investigation of GAs as an analysis tool within the GIS field.

In carrying out the experiments, the Multi-GA expanded the information demanded from the simple "where can I best place these *n* stores?" to the more advanced "what's the least number of stores I require to obtain population coverage and where should I place them?"

6.3.2 *The site selection problem*

Two problem surfaces were used for site selection, consisting of a simple 10 x 10 population grid and a more complex 100 x 60 surface. Figure 6.5 shows both the precise population distribution over the simple grid (right) and a 3D representation of this distribution (left).

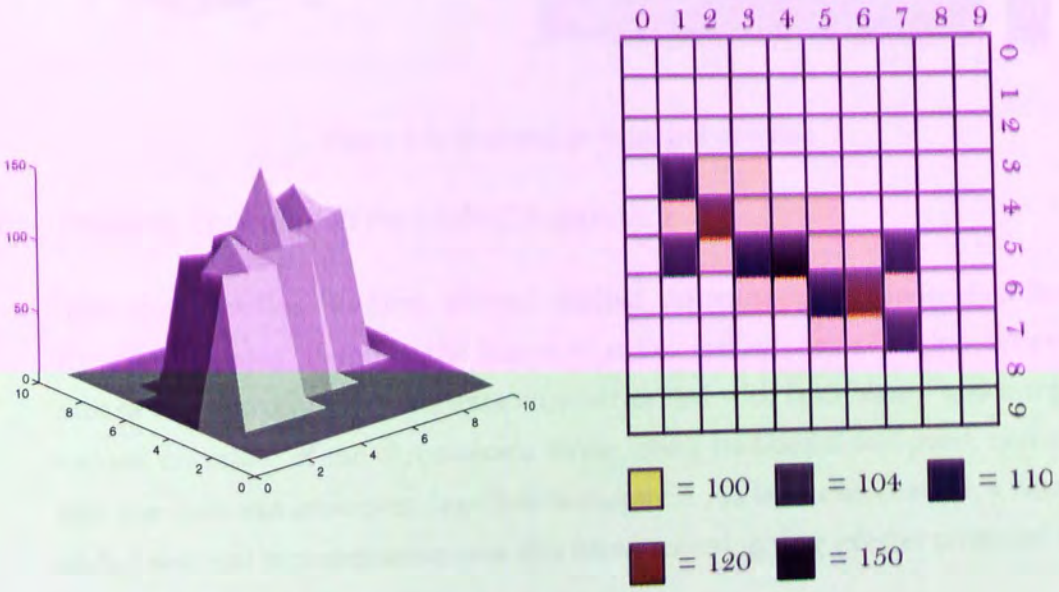


Figure 6.5: Retail store location problem, simple grid structure

In evaluating the relative fitness of stores, the rules defining the behaviour of customers within the population used by Murnion (1995) were adopted.

These rules stated that:

- Customers travel a maximum of one square – so, the customer base of a site consists of its own square and the surrounding 8 squares.
- Customers always travel to the nearest store.
- If *n* stores are equidistant from a grid square, the customer base of that square is divided equally amongst the *n* stores.

A number of variations to the basic query were tested, covering combinations of optimisation of profitability, store location and number of profitable stores. The details of the tests performed are given along with their results in section 6.3.5.

After studying the simple grid, the problem was then extended to the larger 100 x 60 population grid, illustrated in figure 6.6.

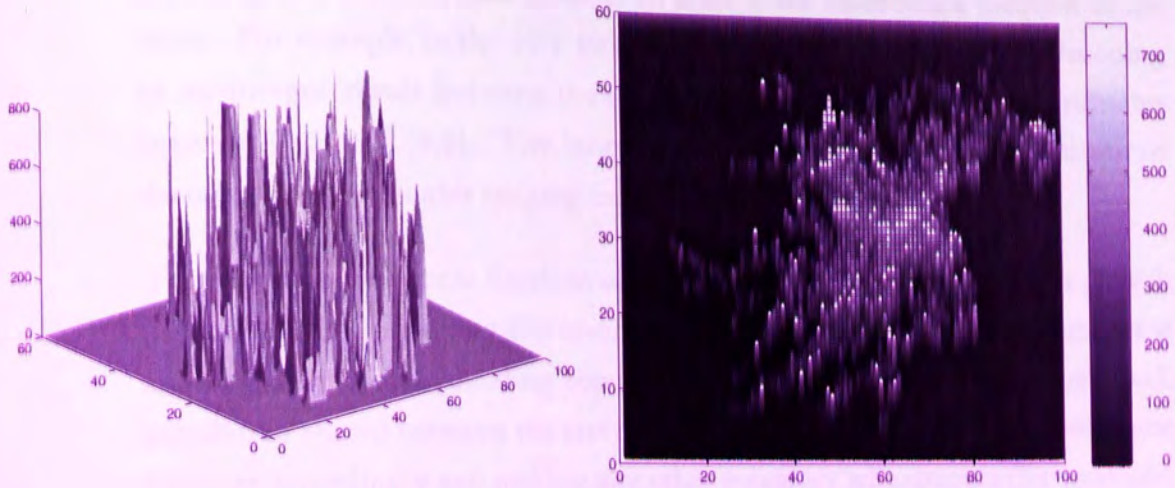


Figure 6.6: Expanded problem grid structure

6.3.3 *Problem Encoding in the Multi-GA approach*

The site selection problem allowed distinct chromosome and individual level fitness functions, defining the fitness of stores and inter-relationship between stores respectively. Experiments were performed with both binary and integer valued encoding of the chromosome string, using traditional one-point, uniform and the quotient crossover described in chapter 5. In later experiments, a binary coded decimal representation was also implemented, giving greater potential for improved schema propagation.

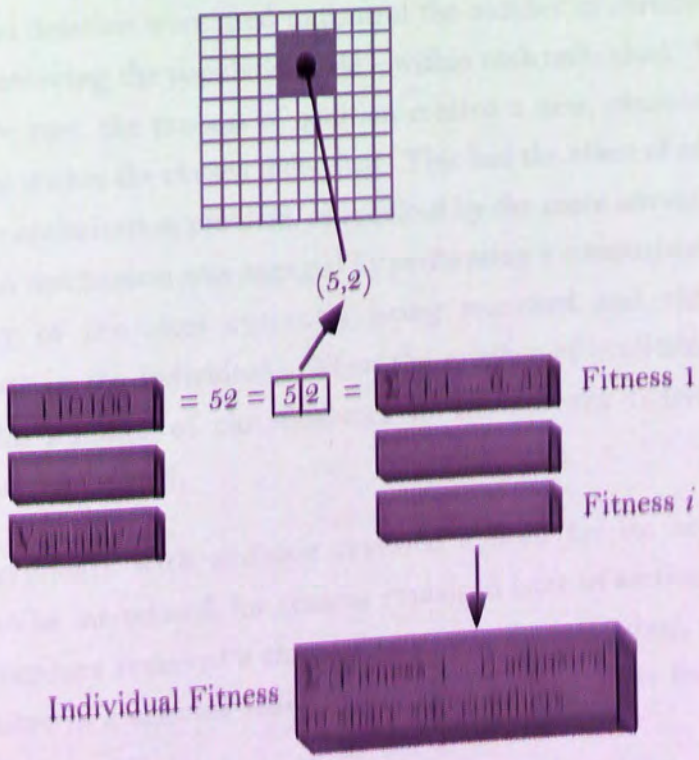


Figure 6.7: Site selection encoding in the Multi-GA.

Each chromosome defined a single store location, the value of the chromosome decoding to a number, split in order to achieve the co-ordinate location of the store. For example, in the 10 x 10 grid, chromosome representations decoding to an integer result between 0 and 99 were translated to grid co-ordinates between (0,0) and (9,9). The larger 100 x 60 grid involved representations decoding to co-ordinates ranging from (0,0) to (99,59).

The chromosome fitness function calculated the population distribution of each particular store, obtaining the co-ordinate value from the decoded chromosome and summing the surrounding population grid squares. At the individual level, population shared between the stores was calculated, adjusting the chromosome fitnesses accordingly and making any other necessary adjustments (for example, relating to the number of chromosomes in an individual). The encoding method and fitness function relationship are illustrated in figure 6.7.

6.3.4 *Extended operators: Addition and Deletion*

During the course of the simple grid experiments, it became necessary to further control the behaviour of the chromosome structure, through the use of the extended operators, addition and deletion. Outlined briefly in chapter 3, addition and deletion were incorporated into the initial multi-chromosome concept as a mechanism for self-adaptation and control. These concepts were realised in application to the more complex site selection optimisation questions.

Addition and deletion were used to control the number of chromosomes – and hence sites covering the population grid – within each individual. When certain criteria were met, the process of addition created a new, randomly initialised chromosome within the chosen individual. This had the effect of adding another site into the optimisation problem, as required by the more advanced questions. The addition mechanism was engaged by performing a comparison between the profitability of the sites currently being searched and the number of chromosomes in the individual. When the number of profitable sites found matched the number of chromosomes in the current individual, a new chromosome was created.

Early experiments with addition revealed a need for its complementary procedure to be introduced, for reasons explained later in section 6.3.5.2. The deletion procedure removed a chromosome from the individual, with the non-profitable sites in a selected individual marked as candidates for deletion. In

order to allow time for a newly introduced chromosome to perform an effective search, deletion was not activated until an individual contained at least *two* unprofitable chromosomes. Once this criterion was met, the first unprofitable store was removed, as illustrated in figure 6.8.

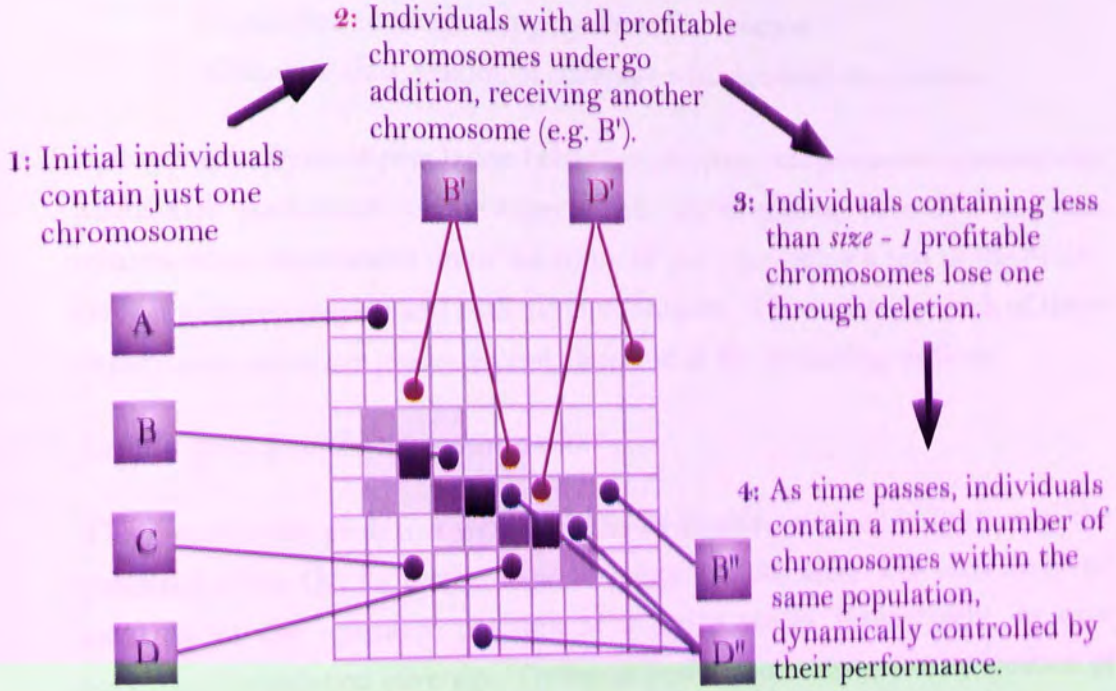


Figure 6.8: Mechanism of addition and deletion for site selection experiments.

Activation of the deletion procedure was achieved in the same way as addition, comparing the number of profitable chromosomes to the number of chromosomes in the current individual. If the difference was above a pre-defined level (in this case 2 or more), deletion was engaged.

6.3.5 *Experimental results*

The basic questions outlined in section 6.3.1, combined with the development and application of the extended operators, provided a series of experiments investigating application and performance of the Multi-GA structure and an analysis of automatic chromosome control. Surfaces exhibiting simple and more complex behaviour provided a range of problem complexity. The experiments were performed over a series of 20 GA trials, with the specific site examples being taken from the run producing the best ever result. Fixed parameters of 75% individual and chromosome crossover level were applied, with mutation applied to each chromosome at 4%. A population size of 20 was used for the 10 x 10 grid, rising to 100 in the 100 x 60 grid. Single member elitism was applied in all cases. Experiments were carried out with one point and uniform crossover

on a single binary encoding, with quotient crossover on integer encoding and one point crossover on binary coded decimal. Initial experiments controlled the direction of later tests, with the following series being carried out on the 10 x 10 population grid:

- Optimisation of 2 and 3 fixed site location
- Population coverage using dynamic site creation
- Minimum sites, maximum coverage with dynamic site creation

Following analysis of population behaviour, dynamic chromosome creation and Multi-GA performance, the experiments investigating coverage and site minimisation were scaled up to the 100 x 60 grid, providing a test of the Multi-GA on a more complex and realistic environment. The results for each of these experiment series are presented and discussed in the remaining sections.

6.3.5.1 *Simple grid, 2 and 3 site optimisation*

The 2 and 3 site problems were undertaken to give an initial indication of the potential of the GA for further experimentation in this area. The tests involved location of the optimum position of 2 and 3 stores respectively, to give maximum population coverage. Optimum performance was given by location of sites at co-ordinates of [(2,4), (6,6)] and [(2,4), (4,5), (6,6)] in the two problems respectively. In this set of problems, the individual fitness was given by the conflict adjusted sum of the two chromosome fitnesses, the optimum co-ordinates giving fitnesses of 1876 and 2426 respectively, with the results shown in table 6.3.

Chrom. Xover	*Sites*	*Average Best*	*Std. Dev.*	*Best Ever*	*Best Ever Gens.*	*Avg. Best Gen.*	*Std. Dev.*	*Store 1*	*Store 2*	*Store 3*
1pt bin	2	1735.3	132.58	1876	33	56.95	110.14	2,4	6,6	–
u/f bin	2	1722.1	157.24	1876	11	99.20	159.91	2,4	6,6	–
Qu. int	2	1873.1	12.64	1876	135	178.50	178.50	2,4	6,6	–
1pt bin	3	2224.8	115.60	2426	59	170.35	175.08	6,6	4,5	2,4
u/f bin	3	2173.4	159.28	2326	428	97.75	113.13	6,6	4,5	1,4
Qu. int	3	2401.0	76.64	2426	191	190.60	125.66	6,6	4,5	2,4

Table 6.3: Multi-GA results for the 2 and 3 fixed site selection problems.

The results indicated all experiments finding the optimum solution on at least one trial, with average best performance indicated as slightly improved (although within standard deviation ranges) from Uniform binary to 1 point binary to Quotient based integer coding. The non-binary encoding scheme showed much improved consistency (that is to say lower standard deviation),

although taking a greater average number of generations to converge when compared to the better of the two binary coded experiments.

6.3.5.2 *Simple grid, dynamic site population coverage*

Following the promising results obtained in the simple, fixed site problems, experiments were undertaken to incorporate the self-generation of genetic material through the use of automatic addition of new stores. In doing so, the question being answered was extended from "where can I best place these n stores?" to questions involving calculation of the number of stores that can profitably be placed on the grid, given a minimum profit criteria. Having placed a single store profitably, the addition routine was called as outlined earlier to add a new chromosome into the population. Experiments were performed at three levels of minimum profitability, 200, 400 and 600 at each of the three encoding and crossover combinations used earlier.

The results are shown in tables 6.4a - d, showing the number of chromosomes found in the best performing member of the 'best ever' experiment performed, and the site co-ordinates of the first found optimum member.

Chrom. Xover	*Min. Profit*	*Average Best*	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Max. Chrom's*	*Test label*
1pt bin	200	2426.00	0.00	2426.00	54	40.30	22.17	7	A
u/f bin	200	2426.00	0.00	2426.00	46	38.65	22.70	6	B
Qu. int	200	2426.00	0.00	2426.00	63	84.35	72.18	6	C
1pt bin	400	2421.00	21.76	2426.00	18	80.25	83.45	4	D
u/f bin	400	2426.00	0.00	2426.00	43	95.35	116.44	5	E
Qu. int	400	2426.00	0.00	2426.00	105	130.40	79.36	4	F
1pt bin	600	2246.60	302.03	2426.00	59	163.75	174.42	4	G
u/f bin	600	2317.90	159.44	2426.00	273	96.00	107.51	4	H
Qu. int	600	2411.00	65.37	2426.00	81	105.15	77.09	4	I

Table 6.4a: Using addition to create and place stores, with unadjusted fitness.

Label	A		B		C	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	7,6	628	2,4	723	6,6	786
Store 2	3,5	550	3,5	665	2,4	723
Store 3	4,4	480	6,5	569	3,5	665
Store 4	4,7	310	6,7	469	6,4	252
Store 5	0,4	308	1,7	0	0,0	0
Store 6	3,2	150	0,0	0	5,2	0
Store 7	9,7	0	–	–	–	–

Table 6.4 b: Site locations for results presented in table 6.4a.

Label	D		E		F	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	2,4	833	4,5	690	2,4	938
Store 2	4,5	691	5,6	650	5,5	665
Store 3	6,6	602	1,4	416	7,6	518
Store 4	5,6	246	2,3	362	4,7	305
Store 5	8,4	52	8,6	308	–	–

Label	G		H		I	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	6,6	938	6,6	786	6,6	833
Store 2	1,4	578	2,4	723	2,4	833
Store 3	3,6	510	3,5	665	4,5	760
Store 4	4,3	400	6,4	252	0,0	0

Tables 6.4 c and d: Site locations for results presented in table 6.4a.

The results indicated good, comparable performance for all methods, with population coverage being achieved by at least one run for all results, in almost all cases for 200 and 400 profitability. 600 profitability began to show some variation, with standard deviation values increasing markedly. Where larger standard deviations were seen, quotient crossover on integer encoding was seen to provide greatest consistency, both in average best and time taken to converge.

Two specific points were raised by the site locations found under the use of addition. Firstly, a number of individuals contained more than one chromosome below the level of profitability. Whilst the creation mechanism was intended to create individuals containing a number of profitable chromosomes, with a single newly created chromosome, this was not seen to continue as the generational cycle progressed. Analysis of the reasons for this was carried out and is explained in more detail in section 6.3.5.3.

Investigations into a method of controlling the spread of unprofitable chromosomes were carried out, resulting in development of the deletion operator to prune out non-profitable chromosomes at the time of fitness evaluation. In designing the deletion operator, account was taken of the need to retain at least one non-profitable chromosome in order to allow adequate time for searching to occur. Failure to account for this would have led to thrashing between addition and deletion procedures, with deletion removing a recently added chromosome before it had time to search and evolve! Consequently, the deletion call was made only when 2 or more unprofitable chromosomes were

present. The results of experiments using addition and deletion are shown in tables 6.5a - d.

Chrom. Xover	*Min. Profit*	*Average Best*	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Max. Chrom's*	*Test label*
1pt bin	200	2426.00	0.00	2426.00	128	53.40	38.39	5	J
u/f bin	200	2426.00	0.00	2426.00	42	54.25	33.16	4	K
Qu. int	200	2426.00	0.00	2426.00	60	63.80	31.73	4	L
1pt bin	400	2426.00	0.00	2426.00	46	63.25	51.23	4	M
u/f bin	400	2426.00	0.00	2426.00	154	90.85	72.05	4	N
Qu. int	400	2426.00	0.00	2426.00	126	110.10	91.58	4	O
1pt bin	600	2370.80	116.67	2426.00	237	228.05	157.04	4	P
u/f bin	600	2385.95	79.97	2426.00	49	163.20	110.07	4	Q
Qu. int	600	2396.00	78.09	2426.00	129	162.75	86.16	4	R

Table 6.5a: Using addition and deletion to create and place stores, with unadjusted fitness.

Label	J		K		L	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	6,6	833	6,6	938	2,4	833
Store 2	4,5	710	2,4	631	4,5	760
Store 3	2,3	574	3,6	455	6,6	731
Store 4	2,6	309	4,3	300	7,8	102
Store 5	0,1	0	–	–	–	–

Label	M		N		O	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	6,6	833	2,4	833	2,4	833
Store 2	4,5	815	4,5	810	6,6	833
Store 3	1,4	416	6,7	579	4,5	760
Store 4	2,3	362	7,4	204	0,0	0

Label	P		Q		R	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	6,6	786	6,6	723	2,4	833
Store 2	2,4	723	5,5	665	6,6	833
Store 3	3,5	665	2,3	624	4,5	760
Store 4	6,4	252	2,6	414	9,0	0

Tables 6.5 b,c,d: Site locations for results presented in table 6.5a.

The introduction of deletion proved successful, with the number of chromosomes reduced in a number of cases, particularly the 200 profitability example. In addition, a slight increase in the performance, and a larger increase in standard deviation, of the binary representations at 600 profitability was seen over the previous experiments, although within standard deviation limits. A similar trend was seen in average generations to converge, with slightly better

performance and a narrowing of the associated standard deviation in most cases using deletion. However, the chromosomes that had not yet reached minimum profitability were still contributing their sub-optimal fitnesses to the individual fitness function. Whilst deletion controlled the number of these chromosomes, it was still considered inappropriate to have unprofitable sites making a positive contribution. Consequently an alternative method of control, more directed towards this objective, was applied with the fitness contribution of non-profitable stores being entirely removed.

Chrom. Xover	*Min. Profit*	*Average Best*	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Max. Chrom's*	*Test label*
1pt bin	200	2426.00	0.00	2426.00	39	57.90	54.91	7	AA
u/f bin	200	2426.00	0.00	2426.00	102	42.95	27.33	8	AB
Qu. int	200	2426.00	0.00	2426.00	17	77.25	42.27	6	AC
1pt bin	400	2410.55	47.97	2426.00	82	117.65	113.69	4	AD
u/f bin	400	2415.45	31.14	2426.00	29	118.25	96.07	4	AE
Qu. int	400	2426.00	0.00	2426.00	153	165.90	127.29	4	AF
1pt bin	600	2069.90	264.48	2426.00	27	57.15	83.83	4	AG
u/f bin	600	2239.00	190.96	2426.00	23	139.45	151.13	4	AH
Qu. int	600	2323.10	200.07	2426.00	54	159.80	138.59	4	AI

Table 6.6a: Using addition to create and place stores, with adjusted fitness.

Label	AA		AB		AC	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	2,4	723	3,5	650	2,4	833
Store 2	6,5	569	6,7	634	4,5	535
Store 3	3,5	565	1,4	448	5,7	425
Store 4	7,7	314	6,4	404	5,4	325
Store 5	4,7	255	3,3	290	8,6	308
Store 6	3,0	0	9,8	0	9,9	0
Store 7	5,1	0	7,2	0	–	–
Store 8	–	–	1,1	0	–	–

Label	AD		AE		AF	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	2,4	833	2,4	833	2,4	833
Store 2	6,6	833	6,6	833	6,6	833
Store 3	4,5	760	4,5	760	4,5	760
Store 4	2,2	0	1,7	0	3,1	0

Label	AG		AH		AI	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	2,4	833	2,4	833	2,4	833
Store 2	6,6	833	6,6	833	6,6	833
Store 3	4,5	760	4,5	760	4,5	760
Store 4	0,0	0	3,9	0	9,5	0

Tables 6.6 b,c,d: Site locations for results presented in table 6.6a.

In order to assess the effects of this behaviour on the extended operators, these experiments were performed with addition only and then with both addition and deletion, with the results of the first of these two experiments given in tables 6.6a - d.

Application of fitness adjustment, zeroing the fitness contribution of non-profitable stores, gave comparable performance to non-adjusted fitness results, with a slight fall in average best performance indicated at 400 and 600 fitnesses. The number of chromosomes was reduced in one case, but matched by an increase in another case. However, the average number of generations to converge showed a slight increase over non-fitness adjusted methods in most cases, although all within standard deviation values of one another. The previously identified situation of multiple non-profitable chromosomes was again seen. The next experiment introduced deletion to control the spread of these chromosomes, with the results shown in tables 6.7a - d.

Chrom. Xover	*Min. Profit*	*Average Best*	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Max. Chrom's*	*Test label*
1pt bin	200	2426.00	0.00	2426.00	66	38.00	20.71	5	AJ
u/f bin	200	2426.00	0.00	2426.00	29	73.55	56.70	5	AK
Qu. int	200	2426.00	0.00	2426.00	151	60.05	39.38	6	AL
1pt bin	400	2426.00	0.00	2426.00	141	125.20	95.31	5	AM
u/f bin	400	2421.00	21.76	2426.00	149	146.40	91.06	5	AN
Qu. int	400	2426.00	0.00	2426.00	113	165.95	118.08	4	AO
1pt bin	600	2396.00	45.81	2426.00	66	189.60	138.46	4	AP
u/f bin	600	2400.90	53.94	2426.00	109	217.20	139.04	4	AQ
Qu. int	600	2405.60	69.46	2426.00	42	213.05	142.43	4	AR

Table 6.7a: Using addition and deletion to find the number of stores, with adjusted fitness.

Label	AJ		AK		AL	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	2,4	723	6,6	833	2,4	673
Store 2	3,5	665	4,5	760	4,6	665
Store 3	6,6	621	2,4	519	7,7	424
Store 4	6,5	417	1,4	314	6,4	354
Store 5	1,1	0	6,9	0	3,3	310
Store 6	–	–	–	–	3,1	0

Label	AM		AN		AO	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	6,6	833	2,4	833	6,6	833
Store 2	4,5	635	4,5	635	2,4	833
Store 3	3,4	490	5,6	490	4,5	760
Store 4	1,4	468	7,6	468	0,7	0
Store 5	2,6	0	8,7	0	–	–

Tables 6.7 b,c,d: Site locations for results presented in table 6.7a.

Label	AP		AQ		AR	
	Co-ord	*fitness*	*Co-ord*	*fitness*	*Co-ord*	*fitness*
Store 1	2,4	833	6,6	833	2,4	833
Store 2	6,6	833	2,4	833	6,6	833
Store 3	4,5	760	4,5	760	4,5	760
Store 4	7,7	0	1,8	0	9,1	0

Tables 6.7 b,c,d: Site locations for results presented in table 6.7a.

As expected, the introduction of deletion showed indications of improved average best performance (although within standard deviations) but produced markedly smaller standard deviation values, indicating greater consistency. The combination of fitness adjustment and deletion, leading to the removal of more zero valued chromosomes produced better results in the lower profitability tests, resulting in population coverage in fewer chromosomes. Performance comparable to previous experiments was seen, but with indications of increased convergence times.

6.3.5.3 *Population behaviour of addition and deletion*

The results presented in section 6.3.5.2 described a number of circumstances in which the dynamic control of genetic material led to unwanted behaviour, unanticipated at the design stage. Primarily, this referred to the propagation of unprofitable chromosomes within the population, rather than the creation of a single searching chromosome within an already fit individual, as intended. With addition only ever creating a single chromosome in an already wholly profitable individual, the question of how multiple unfit chromosomes came about required answering. In addition, the analysis of deletion and the mechanism by which it achieved removal of this problem was also required, to gain a thorough understanding of the behaviour of these new operators.

Over time, the effect of addition was to produce a population consisting of individuals containing a differing number of chromosomes. Whilst the initial aim of producing more chromosomes when the conditions of addition were fulfilled, it failed to take account of the effects of the other genetic processes. It was the genetic operators of crossover and mutation that led to the evolution of individuals with multiple unprofitable chromosomes.

With individuals selected for genetic manipulation on the basis of their individual fitness, some selected individuals contained an unprofitable chromosome (either because they were 1 chromosome individuals, or a

chromosome individuals subjected to addition). In some cases, individual level crossover acted to mix chromosomes from two individuals containing an unprofitable mixture introducing an additional (unprofitable) chromosome without a call to the addition procedure. However, this alone did not explain the widespread propagation seen in the population.

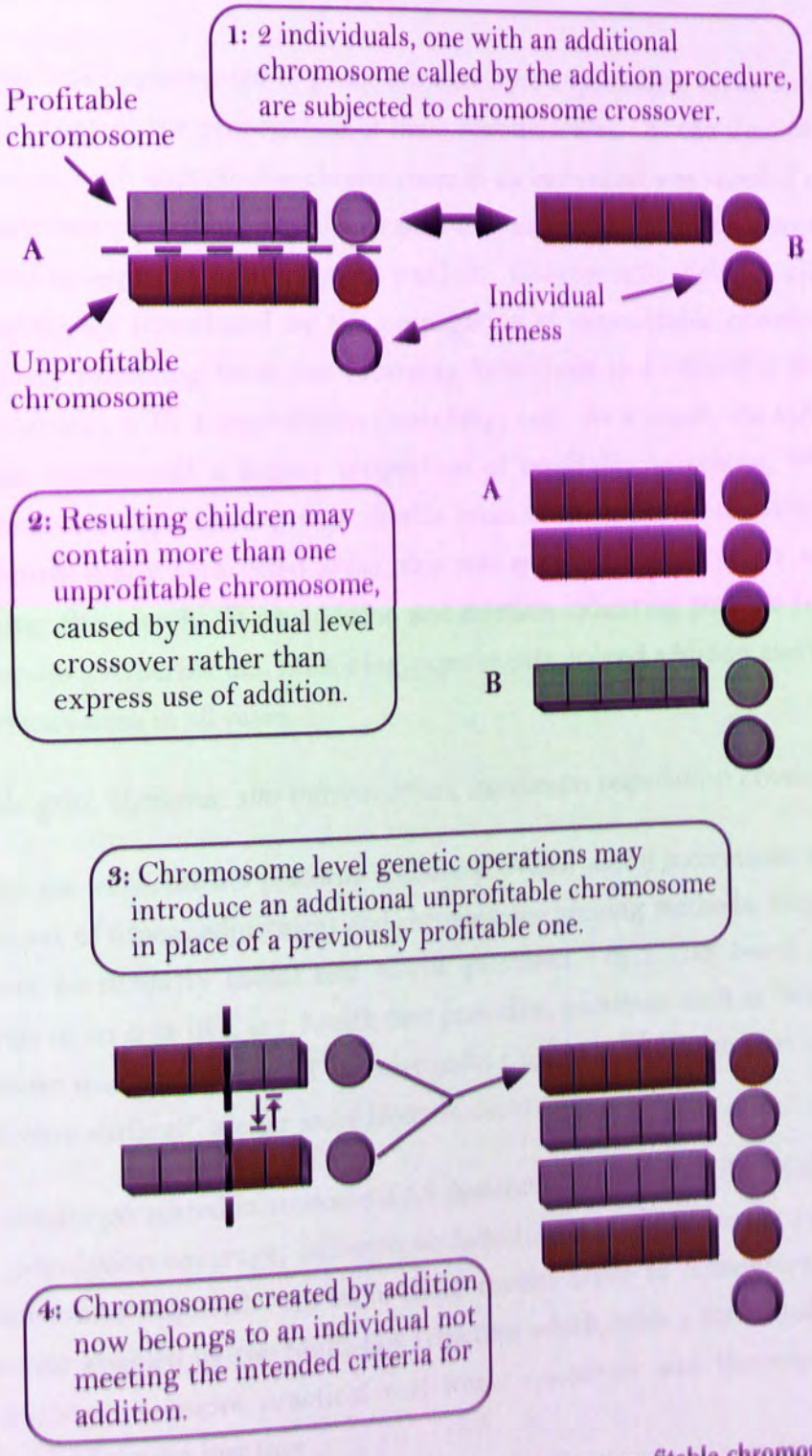


Figure 6.9: **Genetic operations act to propagate unprofitable chromosomes without calling the addition procedure.**

Having selected an individual previously containing (say) 3 profitable and 1 unprofitable (added) chromosomes, crossover and mutation between the feature chromosomes resulted in the production of children containing unprofitable members within the pre-existing 3 chromosomes. With no mechanism for deletion of the extra genetic material, individuals with more than one unprofitable chromosome spread through the population, as illustrated in figure 6.9.

Deletion was implemented to prune out excessive unprofitable chromosomes, in order to control the propagation of these chromosomes. At the time of fitness evaluation, each unprofitable chromosome in an individual was labelled as such. The deletion procedure was then called and unprofitable chromosomes were removed in sequence until only one was left. Consequently, deletion redressed the imbalance introduced by the propagation of unprofitable chromosomes, selectively removing them and returning individuals to a state of n profitable chromosomes with 1 unprofitable (searching) one. As a result, the GA search process maintained a higher proportion of profitable searching, with less chromosomes representing unprofitable areas of the grid. In the larger grid, containing many zero rated areas, this was seen to be particularly useful in directing the search. With addition and deletion indicating positive results in the previous experiments, remaining experiments utilised addition and deletion of chromosomes in all cases.

6.3.5.4 *Simple grid, dynamic site minimisation, maximum population coverage*

Whilst the experiments performed above provided useful indications as to the behaviour of fitness adjustment and chromosome pruning methods, they did not address particularly useful real world questions. In a GIS based problem analysis in an area like, say, health care provision, questions such as "what is the minimum number of hospital casualty units I need to provide to best cover the population surface?" are far more likely to occur.

The results presented in section 6.3.5.2 illustrated attempts by the Multi-GA to gain population coverage, but with no importance attached to the number of chromosomes required. As such, these results serve to demonstrate useful behaviour enabled by the Multi-GA structure which, with a little modification, can answer the more practical real-world questions and the experiments performed here did just that.

In order to maximise coverage and simultaneously minimise the number of sites, the fitness function used in the earlier experiments was modified to incorporate

a measurement of the number of chromosomes. However, careful thought had to be given to the exact formulation of the fitness function. Initial tests using solely the number of chromosomes as a fitness function resulted in the GA producing a population of perfectly placed single sites, but not obtaining population coverage by any measure. Adjusting the fitness function to reflect population coverage by dividing this result by the number of chromosomes performed better, but still produced solutions with less than optimum population coverage. In order to truly answer the question, the GA must first gain optimum population coverage, and then selectively reduce the number of chromosomes to as few as possible.

This was achieved by means of a three-stage process in fitness function evaluation, operating as follows:

1: Optimise population coverage, as with previous experiments, measured by chromosome fitness.
2: For individuals with optimal coverage, add a substantial fitness bonus.
3: Divide the fitness value of individuals with the bonus by the number of profitable chromosomes they contain.

The value of the bonus was sufficient to ensure that, even after division, the resulting fitness was still higher than those which had not yet reached population coverage, ensuring that individuals with full coverage were not lost from the population. However, given a number of individuals with optimum coverage, the division effect means that the overall goal (maximisation of fitness) would favour those achieving coverage with less chromosomes.

The addition of the bonus did not adversely affect the selection pressure of the GA, due to the implementation of rank based selection ensuring an even selection policy. Having obtained a suitable fitness function, experiments were performed under the same conditions as the previous tests, with one exception. In previous experiments, no explicit duplicate handling was implemented, control being left to the evolutionary process. However, with duplicate locations not usually a practical option in this type of real-world problem, duplicate locations were explicitly flagged for deletion and removed by the deletion mechanism. Furthermore, the addition of a bonus value of 10,000 led to the maximum achievable value changing from a fitness of 2426 to 4142. The results of these experiments are presented in tables 6.8a and b.

Chrom. Xover	*Min. Profit*	*Average Best*	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Max. Chrom's*
1pt bin	200	3779.57	493.90	4142.00	80	129.70	100.38	4
u/f bin	200	3748.51	551.44	4142.00	417	169.05	145.89	4
Qu. int	200	3665.67	607.69	4142.00	166	140.85	115.85	4
1pt bin	400	3727.80	507.29	4142.00	102	152.20	117.36	4
u/f bin	400	3934.90	414.20	4142.00	73	129.70	121.86	4
Qu. int	400	3934.90	414.20	4142.00	86	185.25	105.70	4
1pt bin	600	3392.40	1056.20	4142.00	121	163.00	139.27	4
u/f bin	600	3869.60	648.44	4142.00	334	213.00	142.43	4
Qu. int	600	3665.50	830.21	4142.00	340	212.15	113.27	4

Table 6.8a: Maximising population coverage in the minimum number of sites possible.

Label	*Co-ord*	*fitness*
Store 1	2,4	833
Store 2	6,6	833
Store 3	4,5	760

Tables 6.8b: Site locations for results presented in table 6.8a.

The results demonstrated all methods achieving the optimum coverage on at least one of the GA trials, with average best values not far behind. In addition, the major objective of the experiment, to do so in the minimum number of chromosomes possible, was also achieved, as indicated by table 6.8b. In each case, the best result produced convergence to the three optimal locations for all profitability levels. This is in contrast to the previous experiments, where optimum coverage at the lower profitability levels (where fitnesses other than the optimum three were allowed) contained individuals of 4 and 5 profitable chromosomes, as opposed to the optimum 3. Individual results for each run are not presented here, as all achieved the three locations shown in table 6.8b, plus an additional, unprofitable searching chromosome.

In an attempt to improve average best performance on the binary chromosomes, and following discussions with Phil Barrett (1995), it was decided to implement an alternative binary encoded chromosome, utilising a binary coded decimal representation (BCD). The binary representation used so far – encoding an integer between 0 and 99, then split by the chromosome fitness function, is not necessarily wholly appropriate, representing potentially good co-ordinates (e.g. (3, #)) over a binary hamming cliff. In order to allow more effective schema propagation, BCD was used representing the two co-ordinates as distinct binary values, decoded separately on a concatenated chromosome string.

The experiments performed above, for binary one point and uniform crossover, were then repeated with the results shown in table 6.9.

Chrom. Xover	*Min. Profit*	*Average Best*	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Max. Chrom's*
1pt bin	200	4142.00	0.00	4142.00	240	102.80	81.20	4
u/f bin	200	4038.45	310.65	4142.00	194	93.35	69.38	4
1pt bin	400	4142.00	0.00	4142.00	36	157.30	109.92	4
u/f bin	400	4142.00	0.00	4142.00	85	67.10	43.55	4
1pt bin	600	4142.00	0.00	4142.00	283	235.95	93.78	4
u/f bin	600	3847.10	706.79	4142.00	195	233.95	152.45	4

Table 6.9: Maximising population coverage in the minimum number of sites possible, using a BCD representation.

The results in terms of number of chromosomes showed no difference, with the site locations being exactly as those illustrated in table 6.8b. However, as expected, average best performance showed a huge improvement, with all 20 GA trials finding the optimum solution in 3 profitable chromosomes in 4 of the 6 experiments performed and all one point crossover methods finding the optimum result. Convergence times were comparable with those using non-BCD binary encoding, but with a number of results indicating tighter standard deviation results. These results confirmed the observations repeatedly made by Davis (1991) and others, that the choice of representation most appropriate to the problem in hand can have a significant effect on performance.

6.3.5.6 *Expanded grid, maximum coverage, minimum chromosomes*

Having performed a comprehensive series of tests investigating representations and extended operator behaviour, the Multi-GA was applied to the more complicated 100 x 60 population grid. The results obtained in the previous sections were taken into account, resulting in only BCD encoding with one point crossover, duplicate deletion and automatic addition and deletion being used. This combination was seen to give best performance on the smaller grid and, for computational and time reasons, further experiments were limited to this parameter combination. Other parameter settings remained the same, with the exception of population size, which was increased to 100 in order to reflect the increased size of the problem being tackled.

Application to the larger population grid also resulted in a change in the fitness values defining profitability, reflecting the more complex population distribution. Whilst retaining the same method of calculating site fitness, the minimum level of profitability was raised to a level considered to be realistic for

a grid of that size, namely 2,000. The fitness value obtained by complete population coverage was also significantly higher at 465,550. However, due to the nature of the population distribution, achievement of full population coverage was not possible with a minimum site profitability of 2,000 as outlying sites are not grouped together. In order to assist with accurate evaluation of GA performance, Matlab was used to provide a guide to the achievable area within the larger surface (figure 6.6, right), illustrated in figure 6.10.

Whilst providing an illustrative guide, the Matlab procedure used to produce this grid presumes no overlap of the catchment areas of sites. As such, figures presented later in this chapter and in appendix B that illustrate the distribution of sites over the grid would, in fact, result in a perturbation of the surface which cannot be reflected in the grid shown in figure 6.10. That taken into account, the reduced grid does give a better indication of the acceptable search space, reducing the noise given by the raw data grid. However, examples of distribution on both the smoothed, 2,000 fitness grid and the raw data grid are presented in appendix B.

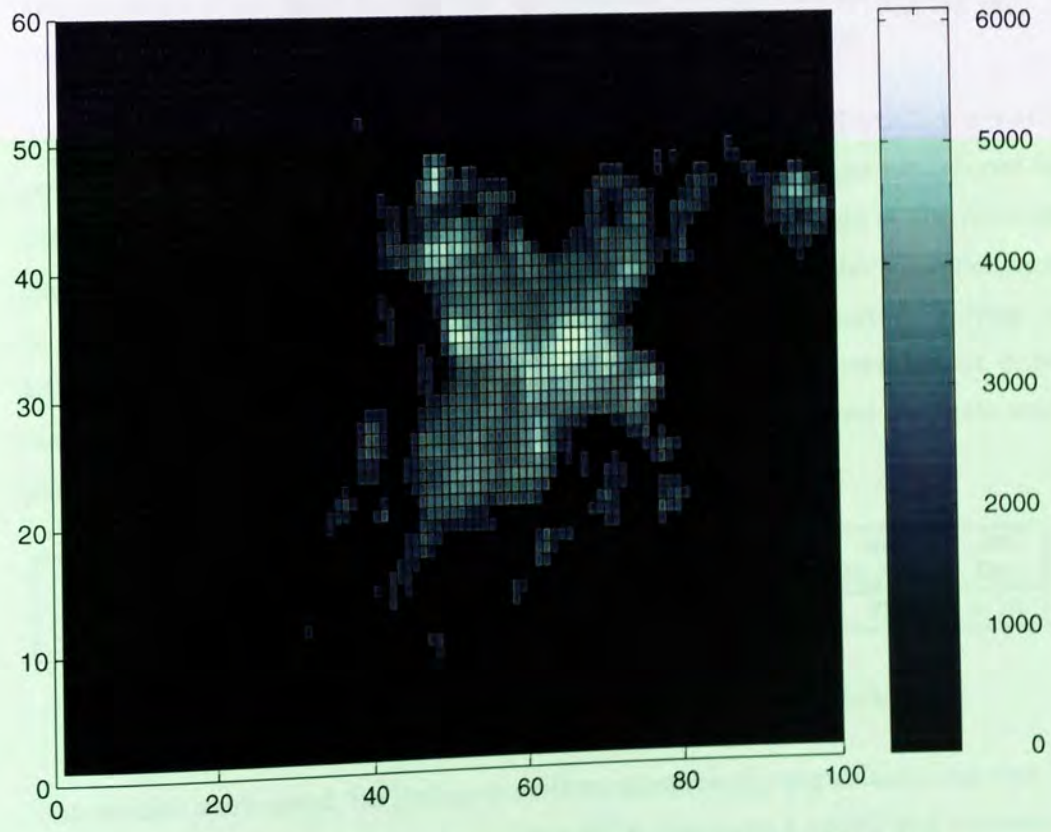


Figure 6.10: Population distribution for site fitnesses of 2,000 or greater.

Having established the level and expected distribution areas in the 100 x 60 grid, preliminary experiments were performed, applying the identical parameter

set used in the small grid. The results of a small series of 6 tests showed disappointing performance, with the Multi-GA operators of addition and deletion appearing to not advance towards population coverage as well as indicated by the smaller grid.

Average Best	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Avg. Prof. sites*	*Std. Dev.*
70825.00	2387.94	75100.00	490.00	428.83	67.59	16	0.57

Table 6.10: Small grid parameter settings extended to the large population grid.

The results shown in table 6.10 were analysed, in order to determine why the Multi-GA was not engaging dynamic adaptation in the efficient manner indicated by the earlier experiments. Following this analysis, it was revealed that the GA was performing adequately, but taking a large amount of time, due to the size of the search space. With such a large search space, the single chromosome produced by addition required much longer to locate the profitable centre of the search space – evident from figure 6.10 as a comparably small portion of the total space. The focus of investigation consequently moved on to examination of methods to increase the efficiency of the search, resulting in a change to the parameters of the addition and deletion mechanism.

The next set of experiments were designed to allow the Multi-GA greater powers of exploration by increasing the number of chromosomes introduced by the addition procedure. This also required an equivalent change in the deletion procedure, allowing the same increased number of non-profitable chromosomes to survive deletion. A level of 10 chromosomes was selected, giving an additional boost to the searching introduced by the incorporation of extra chromosomes. The other GA parameters remained the same and the tests were performed over 20 trials.

Average Best	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Avg. Prof. sites*	*Std. Dev.*
118389.00	6898.05	131500.00	487.00	479.20	14.17	29.70	2.79

Table 6.11: 10 chromosome addition applied to the large population grid.

The results were good, indicating a definite, statistically significant increase in average fitness by increasing the number of chromosomes added and deleted at each step. Given an equivalent number of generations, it was clearly demonstrated that the increased searching introduced by the addition of multiple chromosomes rather than a single chromosome led to faster evolution

of improved solutions. The improvement in fitness is reflected by better population coverage, as illustrated by overlaying the site locations of the best performing run onto the raw and smoothed fitness grids. Figure 6.11 shows these results for the smoothed fitness grid, with the full grid results given in appendix B, figure B1.

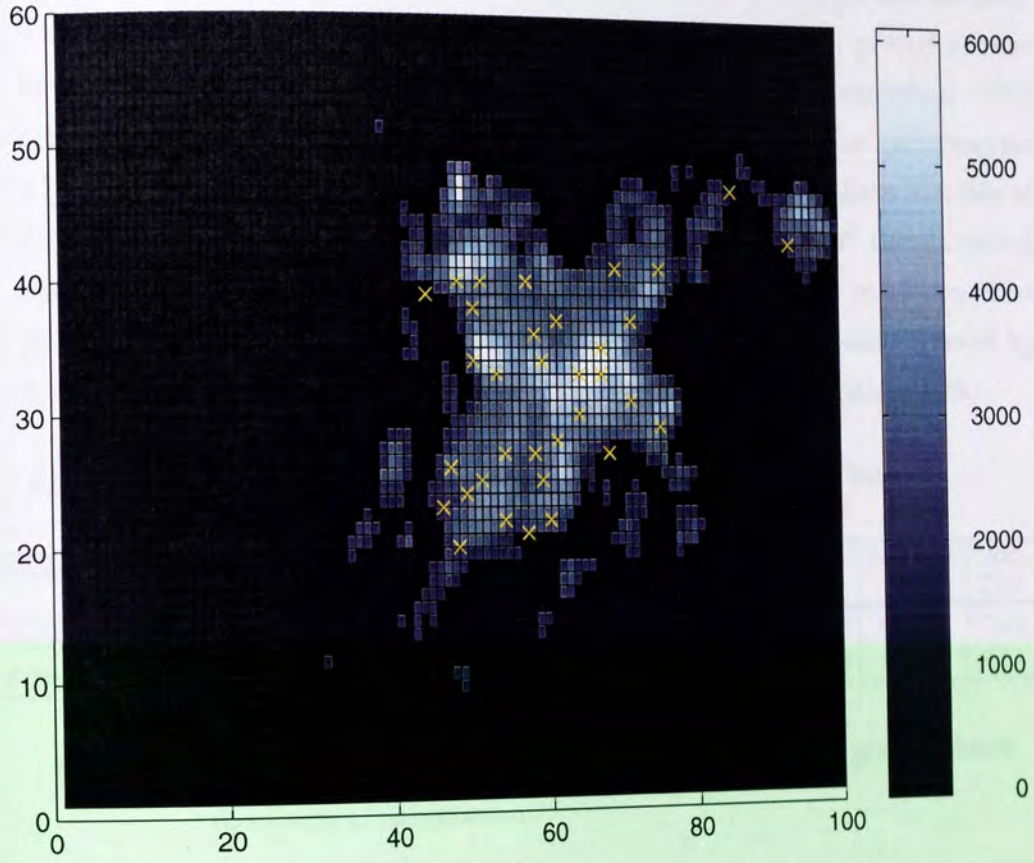


Figure 6.11: Best run locations found by the Multi-GA displayed on the 2000 fitness smoothed population grid at the 100% threshold.

However, the results also show a best achieved fitness that falls significantly short of giving population coverage of both the entire area and the area of profitability given by the smoothed grid. This result prompted further analysis of the mechanisms of addition and deletion, in an attempt to further improve distribution over the population surface and consequently fitness. Examination of the GA output showed that the requirement of obtaining minimum profitability for 100% of the individual's chromosomes was difficult to achieve, hampering the ability of the individual to obtain additional chromosomes. The result was to effectively return to the situation being given in the earlier single add/delete experiment after a brief period of time.

The best performing individuals contained a significant number of profitable chromosomes, but were being prevented from obtaining an additional 10

chromosomes by (in some cases) as few as one or two unprofitable chromosomes. Given the large area of non-profitable space in the problem surface, this was considered to be an unfair handicap leading to an unrealistically large execution time in order to obtain adequate results.

This led to further investigations designed to continue the improvement shown by the move from a 1 to 10 chromosome threshold for addition and deletion. The above results indicated that engagement of the addition procedure was being held back by a small minority of the individual's chromosomes. The response to this behaviour, intended to relax the criteria for chromosome addition and hence allow potential improvement, was to perform a series of experiments with a lower requirement for engagement of the extended operators. Specifically, experiments were performed with the 10 chromosome threshold for addition and deletion applied when profitability was achieved by 75% and 50% of the individual's chromosomes, instead of the stricter 100%.

The results of this experiment series are presented in table 6.12 below.

Threshold	*Average Best*	*Std. Dev.*	*Best Ever*	*Gens for Best ever*	*Avg. Best Gen.*	*Std. Dev.*	*Avg. Prof. sites*	*Std. Dev.*
75%	265037.00	40198.30	301121.00	474.00	464.90	44.10	93.60	7.36
50%	268514.00	8239.58	284162.00	456.00	381.90	84.83	88.30	6.91

Table 6.12: 10 chromosome addition applied to the large population grid at reduced thresholds of engagement.

As shown, the average best results showed a significant improvement from those presented in the previous table, with the fitness obtained more than doubling. The expected behaviour – that is to say, allowing the GA to more quickly evolve an individual containing a large enough number of chromosomes to offer an acceptable and realistic solution – was demonstrated by the introduction of lower requirements for calling the extended operators. This can clearly be seen from the average number of sites found, increasing threefold from around 30 to 90 sites.

The result, seen particularly clearly in a graphical analysis, is to produce solutions giving a fairly even coverage of the problem surface. The GA succeeds in locating the more isolated island areas of the problem space, demonstrating its searching behaviour. Figure 6.12 demonstrates the typical coverage obtained with the 75% threshold on the smoothed population grid. The remaining illustrations, showing 75% results on the full population grid and the similar 50% results are given in appendix B, figures B2 - B4.

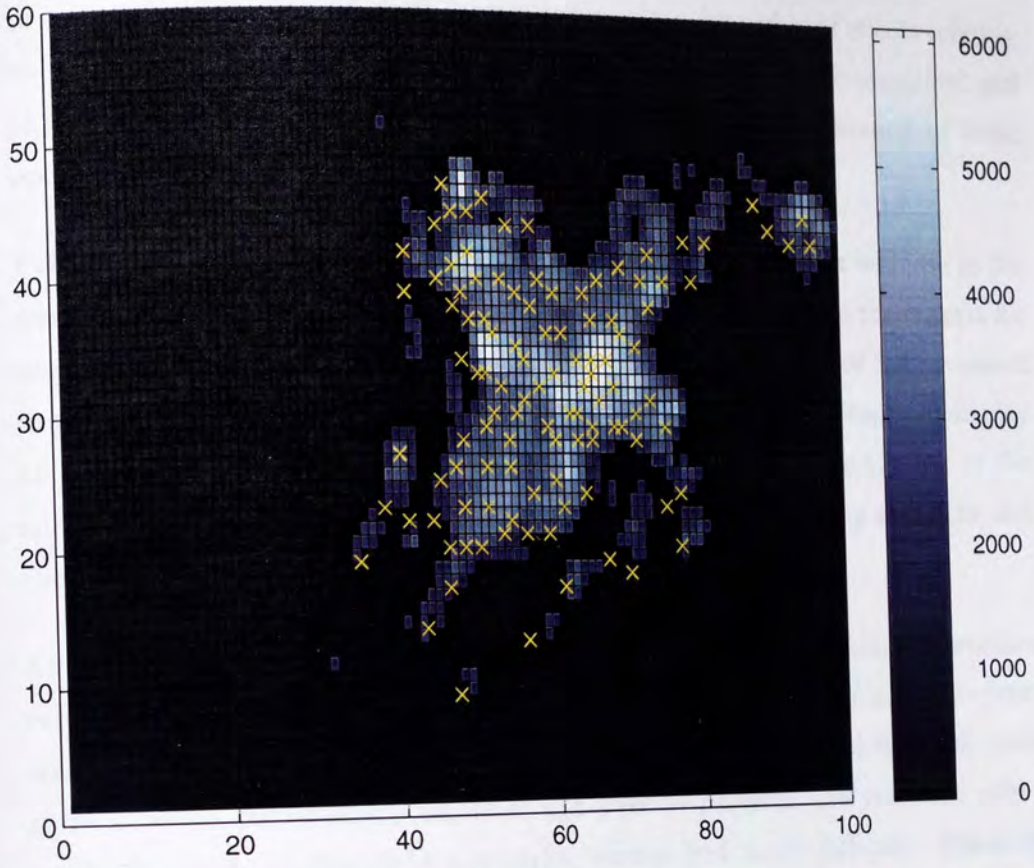


Figure 6.12: **Site locations found using 10 chromosome addition and deletion, engaged at the 75% threshold displayed on the smoothed grid.**

Study of the output superimposed on the raw data surface (appendix B, figures B2 and B3) shows how the GA has found the areas of interest in relation to the other areas of population on the surface, more clearly demonstrating how the GA has successfully picked out the areas of profitability from amongst the outlying, but non-profitable, population distribution.

6.3.6 *Summary of Multi-GA application to spatial analysis*

Several experiments were carried out on two population distribution surfaces in an effort to provide a GA based method of answering the type of question typically put by users of GIS, providing a new approach to the development of GIS analysis tools. The Multi-GA structural flexibility was utilised, implementing a limited chromosome fitness function and introducing the extended addition and deletion operators to self-adapt the genetic material in an individual. By following this method, the Multi-GA was able to search the problem space, expanding its genetic material to incorporate new sites as the profitability requirements became met.

Initial experiments indicating success on a limited 10 x 10 grid were extended to a larger 100 x 60 grid, with interesting results. A number of effects relating to the operation of the addition and deletion procedures were identified and experiments performed to improve the efficiency of implementation of these operators.

Following a series of experiments, it was demonstrated that an increase in the number of chromosomes assigned by addition and a lowering of the criteria for engaging of the addition and deletion procedures led to a marked improvement in performance within an equivalent number of generations. Graphical display of typical solutions obtained from the Multi-GA revealed the success of the addition and deletion procedures in evolving a solution producing a realistic and even coverage of the problem surface.

As suggested by the success of existing GA approaches to the class of problem represented by site selection, the Multi-GA structure was shown to perform well in tackling the problems presented to it. GAs, as a tool for GIS, were shown to be particularly suitable and able to perform analysis that other methods, such as Hopfield networks, would find more difficult. Looking specifically at the Multi-GA architecture, its applicability to a real world problem was illustrated and a number of areas identified in Chapter 3 as holding future research potential were touched upon. The results presented, and the analysis of operators such as addition and deletion, reveal several areas in which there is scope for yet further study into the application of multi-dimensional structures, such as the Multi-GA, both in GIS and the wider class of problems they represent.

Chapter 7: Conclusions

7.1 The Multi-chromosome approach

The adoption of a multi-chromosome approach was undertaken following consideration of the complexity of natural organisms, not currently reflected in traditional GA architectures. In addition, examination of Holland's (1975) proposals revealed a number of ideas suggesting a multi-chromosome approach that have not been fully exploited in the literature to date.

Recent trends in GA research have shown a movement towards a number of more structured interpretations of the traditional GA architecture (e.g. Davis, 1993; Dasgupta, 1992; Harvey, 1992a, 1992b; Juliff, 1993). On closer examination, it can be seen that much of the recent structural work in GAs has a close parallel to research taking place in the other fields of EC (e.g. Smith, 1980), although a distinct comparison and cross-fertilisation of ideas has frequently been overlooked. Despite this wealth of structural alternatives and their associated lessons and future potential, there has been relatively little work in the GA field towards the development of a non-hybridised multi-chromosome architecture operating purely according to genetic principles.

The Multi-GA proposed in this thesis has brought together a number of the different approaches into a distinct, flexible and purely evolutionary mechanism. The genetic process was modified to reflect the new structure, introducing individual level crossover to exchange entire feature chromosomes between individuals, whilst improving the independent feature chromosomes through traditional chromosome level crossover and mutation. Holland's (1975) suggestion of using 'sets of chromosomes' with translocation and segregation operators was realised in the structure of the Multi-GA. In addition, independence of representation, operator application and rates of operator application were introduced, giving the potential for application of the structure to complex, multi-dimensional problems currently difficult to encode in a traditional GA manner.

Comparative testing against a traditional GA was undertaken, along with extension of the structure to areas of traditional GA research identified as potentially gaining benefit from the features provided by the Multi-GA structure. Further developments to exploit the improved flexibility were made, with new genetic operators being developed and applied to real world applications problems, in addition to giving consideration to other areas of development.

7.2 Traditional GA performance comparison

A performance comparison between the traditional and multi-chromosome GAs was carried out, using a number of test optimisation surfaces of increasing complexity. Both GAs showed good performance and located the global minimum in most cases. The traditional and Multi-GAs gave mostly comparable performance, with results on both the simpler and more complex test surfaces producing results within standard deviation values of each other. In some areas, indications of slightly improved performance were seen from the Multi-GA.

Considering that the application of the Multi-GA in a traditional GA manner did not utilise the full representational benefits or flexibility of the new structure, the results clearly demonstrated that no degradation in performance was seen from its application in a traditional GA environment.

7.3 Improved flexibility

The comparative testing carried out, whilst demonstrating no loss of performance, was not able to fully explore the Multi-GA's flexibility. Consequently, a number of additional tests were performed on areas expected to gain from utilisation of this potential.

7.3.1 Dynamic Parameterisation

The ability of the Multi-GA to apply different parameter rates to each independent feature chromosome was seen as having particular application in the field of dynamic parameterisation. Other authors (e.g. Grefenstette, 1986; Srinivas and Patnaik, 1994; Fogarty, 1993; White & Oppacher 1994; Starkweather *et al.*, 1990) have reported various successes in studies of the dynamic configuration of operator rates and their methods of application.

A number of experiments were carried out on the Multi-GA using both dynamic crossover and mutation. Several alternative methods of calculating the dynamic rate were developed, relating the feature chromosome's dynamic rate both to the fitness of the individual containing it and its peers. Specifically, these calculations related to the rank of the current individual in the population and to the fitness difference between the current chromosome and its peer in the best performing individual.

In addition, experiments investigated direct and inversely proportional relationships in the calculation of the dynamic rate and study of the behaviour of the newly introduced individual level crossover mechanism. The results revealed

a significant performance improvement resulting from the use of individual level crossover over experiments in which it was excluded. No real distinction was seen by varying the rate of dynamic chromosome crossover, with 75% giving good results. Dynamically assigned mutation showed good results in a number of cases, with the best performing combinations showing improvements over the fixed rate results for both traditional and Multi-GA. The improved performance was dependent upon the rate calculation method chosen, with inverse rank based calculation emerging as giving the best overall performance. Direct fitness distance calculation proved to give extremely fast convergence to a sub-optimal solution in most cases.

In addition to identifying an appropriate method of applying dynamic parameterisation in a Multi-GA context, the results identified similarities with a number of observations previously reported by other authors studying dynamic parameterisation in the traditional GA (e.g. Starkweather *et al.*, 1990). Most importantly, the results demonstrated the successful application of the flexible structure and its independent chromosome parameter rates in a dynamically controlled context - a finding which has important implications for other areas.

7.3.2 *Non-binary representations*

In addition to applications involving dynamic parameterisation, current GA research in the fields of alternative encoding methods were tested in the Multi-GA context. The Multi-GA's ability to use independent feature chromosomes of differing type representations required testing, before full implementation in a mixed type applications context. Problems tested for dynamic parameterisation with binary chromosomes were repeated with real valued chromosomes. In addition, a new non-binary Quotient crossover operator was developed, with child values being produced from scaling by the relative fitness distance between the two parents. Tests were performed with the new quotient crossover operator and the traditional averaging operator used in many non-binary applications.

The results showed the quotient crossover operator to outperform averaging in most cases. Averaging crossover very quickly converged to an area of interest, accounting for its poor performance as being due to premature convergence. However, the results obtained by real encoded experiments were not nearly as good as those obtained using a binary representation, indicating the binary encoding method to be more appropriate for the problems tested here. In addition to demonstrating the success of quotient crossover as a new non-binary

operator, the results also demonstrated the capability of the Multi-GA to work with chromosomes of a non-binary representation.

7.3.3 *Mixed representation individuals*

In order to further demonstrate the Multi-GA's ability to exploit independent feature chromosomes, it was applied to a real world problem requiring chromosomes of mixed type within each individual. This served to demonstrate the ease in which a number of chromosomes could be manipulated within an individual, whilst using different representations and operators.

The network placement problem used three chromosomes to define the position of a computer in a building, with real valued x and y chromosomes defining its position on each floor, determined by a binary encoded z chromosome. Each Multi-GA individual consisted of several of these chromosome triples for each machine, representing 8 and 15 machines respectively in the two problems tested. Whilst being a newly formulated problem, this application is similar to the class of problems represented by facility layout on which a number of traditional GA experiments have been carried out (e.g. Tam, 1992; Tate & Smith, 1993b).

The results were very satisfactory, with the Multi-GA handling the mixed representation with little difficulty. Good solutions, producing highly acceptable locations given the restricted ranges of the machines within the building, were produced. The results showed the Multi-GA to have no apparent problems in dealing with a number of chromosomes of different type, with different operators applied, whilst contained within a single individual. The genetic process succeeded in evolving good solutions in all of the chromosomes represented, illustrating the independence of feature chromosomes.

7.3.4 *Self-adaptation of genetic material*

As well as application to the multi-representational problem described in the previous section, the Multi-GA was applied to a spatial analysis task involving the optimum location of a minimum number of sites on a population grid. This task allowed exploration of two key features of the Multi-GA. Firstly, the use of a chromosome level fitness function of limited independence was demonstrated, with the fitness of each site being determined at the chromosome level. This result was important, showing the potential for this type of feature chromosome independence that may well be crucial in successful application to more naturally independent multi-dimensional problems.

Secondly, the Multi-GA structure was further increased in flexibility by incorporating the ability to dynamically control, generate and delete its own feature chromosomes as required by the problem in hand. Two new genetic operators were created – addition and deletion – which allowed the Multi-GA to dynamically generate new chromosomes to be added into the search process. A series of investigations into the parameterisation and efficient usage of the addition and deletion operators were carried out, resulting in a greater understanding of the new and existing genetic operators, the best method by which to apply them and the Multi-GA structure itself. In particular, the relationship between the self-generation of new genetic material and the effects of individual and chromosome level crossover on the parameterisation of the addition and deletion operators were identified.

The resulting operators were then applied to the site selection problem – a representative of the class of problems including time tabling and scheduling – with highly promising results. The Multi-GA achieved optimal solution of a small population grid experiment and produced highly acceptable solutions on a larger and far more complex grid.

7.4 Real world applicability

The example discussed in the previous section also served to present the Multi-GA structure as a serious tool for real world optimisation tasks. In successfully applying the self-adaptation operators to site selection and spatial analysis, the Multi-GA demonstrated an ability to tackle a class of problems frequently examined in fields such as Geographical Information Systems. A review of the current state of GIS revealed much criticism by leading researchers (e.g. McGuire *et al.*, 1991) of the lack of analytical tools available to interpret the data provided by GIS. The analysis desired is frequently exactly that demonstrated by the Multi-GA in the last series of experiments. In addition, the use of addition and deletion provided the Multi-GA with the ability to answer advanced spatial analysis questions, identified by the authors of alternative approaches such as Hopfield networks as difficult to implement (Murnion, 1995a).

As discussed in the next section, the successful application of the Multi-GA to a real world multi-dimensional task illustrates the potential for future extension to other complex, representationally diverse optimisation problems in the same class and perhaps beyond.

7.5 Future work

The Multi-GA structure proposed here, and the subsequent studies that have been carried out, leave a number of areas open for future investigation.

7.5.1 *The multi-chromosome approach*

The approach and structure of the Multi-GA itself has much potential for future investigation into the dynamics and mechanisms of its operation. In particular, areas such as the theoretical analysis of schema propagation, the relationship of the individual and chromosome levels to schema manipulation and a reformulation of the schema theory to account for the more advanced structure are all areas that would provide a great deal of useful information.

The interdependence of chromosomes and their relationship in more naturally multi-dimensional problem environments would also provide much scope for investigation, with greater study of the means by which different feature chromosomes affect the fitness of the individual as a whole being an area proving beneficial. Increased study of the operators at both individual and chromosome levels and their effects on the genetic process would provide greater insight into the successful application of the Multi-GA and no doubt associated performance improvements.

An investigation of hybridisation might also be a worthwhile study, with a number of authors having shown beneficial results from combining GA techniques with non-genetic search methods. In areas that might benefit from the application of the Multi-GA as a genetic based method, it is not inconceivable that hybridisation with other problem specific methods could produce good results.

7.5.2 *Parameterisation*

Parameterisation has been an area of ongoing research in the GA community for some time and the Multi-GA structure has a number of additional parameters that would benefit from a detailed study. In particular, a full and thorough analysis of dynamic parameterisation within the Multi-GA structure and in comparison to a dynamically parameterised traditional GA would be of use.

Analysis of the optimum parameterisation of the new genetic operators would be a useful study to undertake, along with determining the inter-relationship between the parameter rates at individual and chromosome level more precisely. Initial investigations were carried out into the use of parameter rates as a

mechanism for excluding information from the genetic search, by reducing the rate of crossover and mutation to 0%. This would be particularly useful in a multi-dimensional application where a particular feature chromosome is found to be very good and accordingly it might be desirable that the genetic information should be retained unaltered. Appropriate lowering of the rates of genetic exchange would be one method of achieving this and a study would be particularly beneficial as a method for preventing disruption of good schemata.

The newly introduced operators of addition and deletion would benefit from further study, in particular defining the relationship of self-adaptation to population size and rates of addition and deletion, all of these being possibly dynamically determined. Further scope also exists for extending the use of addition and deletion to generate chromosomes that would themselves be used to evolve parameter rates for other chromosomes, following the lead given by researchers such as Schaffer & Morishima (1987) who advocated inclusion of parameters into the genetic process.

7.5.3 *New genetic operators*

Briefly mentioned in the previous section, the new genetic operators of addition and deletion have scope for further development within the Multi-GA context. In addition to parameterisation studies already mentioned, improvements in the operator mechanisms to ensure more efficient, beneficial application would be a potentially useful area for research. Other uses of the add and delete operators for internal housekeeping within the individual – that is to say, control and retention of good genetic material through copying and evolution of internal parameter rates – are possible and would provide an interesting avenue of investigation. Extension of the scope of self-generation to the chromosome level may also be desirable in certain problem applications and the existing addition and deletion mechanisms may provide a useful framework for such a process.

Quotient crossover was also introduced as a new non-binary mechanism and a detailed study of its effects and comparison with averaging crossover would prove useful. Other analyses, related to schemata in non-binary mechanisms and the relationship of the new crossover mechanism to schema propagation would again be of interest.

7.5.3 *Extending traditional GA applications*

A number of areas currently investigated by GAs contain great potential for exploration in a Multi-GA context. In particular, the structure could easily be

applied to areas utilising co-evolutionary strategies, with different chromosomes being defined as different members of a sub population. Specific applications currently investigated by GAs in lower dimensions, such as time tabling, could be greatly extended by the Multi-GA. For example, many time tabling applications currently work with the goal of optimising a single timetable. In the real world, a school or university may have a timetable for each year and may wish to optimise, for example, a number of timetables in a department.

This is an area which could be appropriately studied in the Multi-GA structure, with the chromosome level fitness function defining the optimisation for each year group timetable, with the individual function governing conflicts across timetables and producing a solution that is better at a departmental level. It could be conceivable that the optimum solution for the entire department consists of a particular year group with a sub optimal timetable, which allows others to reach optimality. This would be difficult to locate with most traditional GA applications, whereas encoding would be more easily facilitated in the Multi-GA application. Generally, any representationally diverse multi-dimensional optimisation task is expected to gain benefit from the Multi-GA structure and investigations to verify this would be worthwhile.

7.5.4 *An analysis tool for Geographical Information Systems*

Finally, the area of Geographical Information Systems has already been identified as gaining benefit from the Multi-GA structure. However, given the current state of the art in GIS spatial analysis tools, the indications of the studies carried out here are that GAs generally, and the Multi-GA in particular, may hold significant potential for advancing the analysis tools available to GIS. Consequently, a more detailed study of the GIS field and the areas in which the Multi-GA could be successfully applied holds a great deal of potential for providing real world benefits. These studies would inevitably link back to a number of the other areas already discussed, depending upon the specific problems discovered.

List of References

Aangeenbrug, R.T. (1991), "A Critique of GIS",*in Maguire et al. (1991)*, pp. 101 - 107.

Abela, J., Abramson, D., Krishnamoorthy, M., De Silva, A., Mills, G. (1993), "Computing optimal schedules for Landing Aircraft", *School of Computing and Information Technology report, Griffith University, Queensland.*

Abramson, D., Abela, J. (1992), "AParallel Genetic Algorithm for solving the School Timetable problem", *in Proceedings of 15th Australian Computer Science Conference*, Hobart.

Abramson, D., Mills, G., Perkins, S. (1993), "Parallelisation of a Genetic Algorithm for the Computation of Efficient Train Schedules", *in Proceedings of 1993 Parallel Computing & Transputers Conference*, Brisbane: IOS Press.

Antonisse, J. (1989), "A New Interpretation of Schema Notion that Overturns the Binary Encoding Constraint", *in Shaffer (1989)*, pp. 86 - 91.

Armour, G.C., Buffa, E.S. (1963), "A heuristic algorithm and simulation approach to relative allocation of facilities", *Management Science 9(2)*, pp. 294 - 300.

Bäck, T., Hoffmeister, F., Schwefel, H. (1991), "A Survey of Evolution Strategies", *in Belew & Booker (1991)*, pp. 2 - 9.

Bäck, T., (1992), "The Interaction of Mutation Rate, Selection and Self Adaptation within a Genetic Algorithm", *in Parallel Problem Solving from Nature, 2 (1992)*, pp. 85 - 94.

Bagchi, S., Uckun, S., Miyabe, Y., Kawamura, K. (1991), "Exploring Problem-Specific Recombination Operators for Job Shop Scheduling", *in Belew & Booker (1991)*, pp. 10 - 17.

Baker, J.E. (1985), "Adaptive selection methods for Genetic Algorithms", *in Grefenstette (1985)*, pp.100 - 111.

Baker, J.E. (1987), "Reducing Bias and Inefficiency in the Selection Algorithm", *in Grefenstette (1987)*, pp.14 - 21.

Bala, J.W., Wechsler, H. (1993), "Learning to Detect Targets using Scale-Space and Genetic Search", *in Forrest (1993)*, pp. 516 - 522.

Barrett, P.J. (1995), Personal communication, September 1995.

Beasley, D., Bull, D.R., Martin, R.R. (1993), "Reducing Epistasis in Combinatorial Problems by Expansive Coding", *in Forrest (1993)*, pp. 400 - 407.

Belew, R.K., Booker, L.B. [Eds.] (1991), *Proceedings of the 4th International Conference on Genetic Algorithms*, San Mateo, CA: Morgan Kauffman.

Biegel, J.E., Davern, J.J. (1990), "Genetic Algorithms and Job Shop Scheduling", *in Proceedings of the 12th Annual Conference on Computers and Industrial Engineering*, Vol 19, Nos. 1-4, pp. 81 - 91.

Blickle, T., Thiele, L. (1995), "A Mathematical Analysis of Tournament Selection",*in Eshelman (1995)*, pp. 9 - 16.

Bowen, J., Dozier, G. (1995), "Solving Constraint Satisfaction Problems using a Genetic/Systematic Search Hybrid that realises When to Quit.",*in Eshelman (1995)*, pp. 122 - 129.

Bruns, R. (1993), "Direct Chromosome Representation and Advanced Genetic Operators for Production Scheduling", *in Forrest (1993)*, pp. 352 - 359.

Bui, T.N., Moon, B.R. (1995), "On Multi-dimensional Encoding/Crossover",*in Eshelman (1995)*, pp. 49 - 56.

Bull, L., Fogarty, T.C., Snaith, M. (1995), "Evolution in Multi-Agent Systems: Evolving Communicating Classifier Systems for Gait in a Quadrupedal Robot",*in Eshelman (1995)*, pp. 382 - 388.

Burke, E.K., Elliman, D.G., Weare, R.F. (1995), "A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems",*in Eshelman (1995)*, pp. 605 - 610.

Caldwell, C., Johnson, V.S. (1991), "Tracking a Criminal Suspect through "Face-Space" with a Genetic Algorithm", *in Belew & Booker (1991)*, pp. 416 - 421.

Carse, B., Fogarty, T.C., Munro, A. (1995), "Adaptive Distributed Routing using Evolutionary Fuzzy Control",*in Eshelman (1995)*, pp. 389 - 396.

Cartwright, H.M., Mott, G.F. (1991), "Looking Around: Using Clues from the Data Space to Guide Genetic Algorithm Searches", *in Belew & Booker (1991)*, pp. 108 - 114.

Cartwright, H.M., Harris, S.P. (1993), "The Application of the Genetic Algorithm to 2-Dimensional Strings: The Source Apportionment Problem", *in Forrest (1993)*, p. 631.

Chen, R.J., Meyer, R.R., Yeckel, J. (1993), "A Genetic Algorithm for Diversity Minimization and its Parallel Implementation", *in Forrest (1993)*, pp. 163 - 170.

Chorley, R., Buxton, R. (1991), "The Government setting of GIS in the United Kingdom",*in Maguire et al. (1991)*, pp. 67 - 79.

Cohoon, J.P., Martin, W.N., Richards, D.S. (1991), "A Multi-Population Genetic Algorithm for Solving the K-Partition problem on Hypercubes", *in Belew & Booker (1991)*, pp. 244 - 248.

Craighurst, R., Martin, W. (1995), "Enhancing GA Performance through Crossover Prohibitions based on Ancestry",*in Eshelman (1995)*, pp. 130 - 135.

Dasgupta, D. (1994), "Optimisation in Time-Varying Environments using Structured Genetic Algorithms", *Technical Report IKBS-17-93, University of Strathclyde*, April, 1994.

Dasgupta, D., McGregor, D.R. (1991), "A Structured Genetic Algorithm: The model and first results", *Technical Report IKBS-2-91 presented at AISB PG-Workshop*, January, 1992.

Dasgupta, D., McGregor, D.R. (1992), "Designing Neural Networks using the Structured Genetic Algorithm", *Artificial Neural Networks 2*, pp. 263 - 268.

Dasgupta, D., McGregor, D.R. (1992a), "Nonstationary Function Optimisation using the Structured Genetic Algorithm", *Parallel Problem Solving from Nature, Brussels (1992)*, pp. 263 - 268.

Davidor, Y. (1991), "A Naturally Occurring Niche & Species Phenomenon: The Model and First Results", *in Belew & Booker (1991)*, pp. 257 - 263.

Davis, L. (1985), "Job Shop Scheduling with Genetic Algorithms", *in Grefenstette (1985)*, pp. 136 - 140.

Davis, L. (1989), "Adapting Operator Probabilities in Genetic Algorithms", *in Schaffer (1989)*, pp. 61 - 69.

Davis, L. (1990), "Genetic Algorithms and the Year 2000", *Computing beyond 2000 – The Sixth Generation, IBC Technical Services, London, 8 November 1990*, p8.

Davis, L. (1991), *A Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold.

Davis, L. (1991a), "Bit Climbing, Representational Bias and Test Suite Design", *in Belew & Booker (1991)*, pp. 18 - 23.

Davis, L., Orvosh, D., Cox, A., Qiu, Y. (1993), "A Genetic Algorithm for Survivable Network Design", *in Forrest (1993)*, pp. 408 - 415.

Davis, L. (1993a), Personal communication in discussion at ICGA5, University of Illinois, July 1993.

Deb, K. (1990), "A Note on the String Growth in Messy Genetic Algorithms", *TCGA Report No: 90006 The Clearinghouse for Genetic Algorithms*, University of Alabama:Tuscaloosa, AL.

Deb, K. (1991), "Binary and Floating point Function Optimisation using Messy Genetic Algorithms", *IlliGAL Report No: 91004 Illinois Genetic Algorithms Laboratory*, University of Illinois:Urbana-Champaign.

Deb, K., Goldberg, D.E. (1989), "An Investigation of Niche and Species Formation in Genetic Function Optimization", *in Schaffer (1989)*, pp. 42 - 50.

Deb, K., Goldberg, D.E. (1992), "Sufficient Conditions for Deceptive and Easy Binary Functions", *IlliGAL Report No: 92001 Illinois Genetic Algorithms Laboratory*, University of Illinois:Urbana-Champaign.

De Jong, K.A. (1975), *An Analysis of the behaviour of a class of Genetic Adaptive Systems.*, Ph.D. Dissertation, Dissertation Abstracts International, 36 (10), 5140B, University of Michigan: Ann Arbor.

De Jong, K.A. (1985), "Genetic Algorithms: A Ten Year Perspective", *in Grefenstette (1985)*, pp. 169 - 177.

Densham, P.J. (1991), "Spatial Decision Support Systems",*in Maguire et al. (1991)*, pp. 403 - 412.

Department of the Environment (DoE) (1987), *Handling Geographic Information*, London: HMSO.

Easton, F., Mansour, N. (1995), "A Distributed Genetic Algorithm for Employee Staffing and Scheduling Problems", *in Forrest (1995)*, pp. 360 - 367.

Eshelman, L.J. [Ed.] (1995), *Proceedings of the 6th International Conference on Genetic Algorithms*, San Mateo, CA: Morgan Kauffman.

Eshelman, L.J. , Schaffer, J.D. (1991), "Preventing Premature Convergence in Genetic Algorithms by Preventing Incest", *in Belew & Booker (1991)*, pp. 115 - 122.

Fogarty, T.C. (1989), "Varying the Probability of Mutation in the Genetic Algorithm", *in Schaffer (1989)*, pp. 104 - 109.

Fogel, D.B., Atmar, J.W. (1990), "Comparing Genetic Operators with Gaussian Mutations in Simulated Evolutionary Processes using Linear Systems", *Biological Cybernetics, 63* (1990), pp. 111 - 114.

Forrest, S. [Ed.] (1993), *Proceedings of the 5th International Conference on Genetic Algorithms*, San Mateo, CA: Morgan Kauffman.

Forrest, S., Mitchell, M. (1993), "What Makes a Problem Hard for a Genetic Algorithm? Some Anomalous Results and Their Explanation.", *Machine Learning 13*, pp. 285 - 319.

Fujiko, C., Dickinson, J. (1987), "Using the Genetic Algorithm to generate LISP source code to solve the Prisoner's Dilemma", *in Grefenstette (1987)*, pp. 236 - 240.

Gabbert, P.S., Brown, D.E., Huntley, C.L., Markowicz, B.P., Sappington, D.E. (1991), "A System for Learning Routes and Schedules with Genetic Algorithms",*in Belew & Booker (1991)*, pp. 430 - 436.

Geffrion, A.M. (1983), "Can OR/MS evolve fast enough?", *Interfaces 13:* pp. 10 - 25.

Goldberg, D.E. (1983), *Computer aided gas pipeline operation using genetic algorithms and rule learning*, Ph.D. Dissertation, Dissertation Abstracts International, 44 (10), 3174B, University of Michigan: Ann Arbor.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization & Machine Learning*, Reading, MA: Addison-Wesley.

Goldberg, D.E. (1989a), "Zen and the Art of Genetic Algorithms", *in Schaffer (1989)*, pp. 80 - 84.

Goldberg, D.E. (1990), "A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing", *Complex Systems 4*, pp. 445 - 460.

Goldberg, D.E. (1991), "Real-coded Genetic Algorithms, Virtual Alphabets and Blocking", *Complex Systems 5*, pp. 139 - 167

Goldberg, D.E., Richardson, J. (1987), "Genetic Algorithms with Sharing for Multimodal Function Optimisation", *in Grefenstette (1987)*, pp. 41 - 48.

Goldberg, D.E., Segrest, P. (1987), "Finite Markov Chain Analysis of Genetic Algorithms", *in Grefenstette (1987)*, pp. 1 - 8.

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

Goldberg, D.E., Deb, K. (1991), "A Comparative analysis of Selection Schemes used in Genetic Algorithms", *Foundations of Genetic Algorithms*, pp. 69 - 93.

Goldberg, D.E., Deb, K., Korb B. (1991a), "Don't Worry, Be Messy", *in Belew & Booker (1991)*, pp. 24 - 30.

Goldberg, D.E., Deb, K., Horn, J. (1992), "Massive Multimodality, Deception and Genetic Algorithms", *IlliGAL Report No: 92005 Illinois Genetic Algorithms Laboratory*, University of Illinois:Urbana-Champaign.

Goldberg, D.E., Deb, K., Kargupta, H., Harik, G. (1993), "Rapid, Accurate Optimisation of Difficult Problems using Fast, Messy Genetic Algorithms", *in Forrest (1993)*, pp. 56 - 64.

Greene, D.P., Smith, S.F. (1987), "A Genetic System for Learning Models of Consumer Choice", *in Grefenstette (1987)*, pp. 217 - 223.

Grefenstette, J.J. [Ed.] (1985), *Proceedings of an International Conference on Genetic Algorithms and their Applications*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Grefenstette, J.J. (1986), "Optimization of Control Parameters for Genetic Algorithms", *IEEE Transactions on Systems, Man and Cybernetics, Vol SMC16, No 1. 3.* pp.122 - 128.

Grefenstette, J.J. [Ed.] (1987), *Proceedings of the 2nd International Conference on Genetic Algorithms and their Applications.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Grefenstette, J.J. (1988), "Credit Assignment in Rule Discovery Systems based on Genetic Algorithms", *Machine Learning vol 3., no 2/3.* pp.225 - 246.

Grefenstette, J.J., Baker, J.E. (1989), "How Genetic Algorithms Work: A Critical Look at Implicit Parallelism", *in Schaffer (1989)*, pp.20 - 27.

Grefenstette, J.J., Ramsey, C., Schultz, A. (1990), "Learning sequential decision rules using simulation models and competition", *Machine Learning 5 (4)* pp.355-381.

Handley, S. (1993), "Automated Learning of a Detector for α-helices in Protein Sequences via Genetic Programming", *in Forrest (1993)*, pp. 271 - 278.

Hansen, N., Ostermeier, A., Gawelczyk, A. (1995), "On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation",*in Eshelman (1995)*, pp. 57 - 64.

Harvey, I. (1992a), "The SAGA Cross: The Mechanics of Recombination for Species with Variable Length Genotypes", *in Männer & Manderick*, pp. 269 - 278.

Harvey, I. (1992b), "Species Adaptation Genetic Algorithms: A Basis for a Continuing SAGA", *Proceedings of the First European Conference on Artificial Life (Eds. Varela, F.J, Bourgine, P.)*, MIT Press.

Harvey, I. (1992c), "Evolutionary Robotics and SAGA: the case for Hill Crawling and Tournament Selection.", *Cognitive Science Research Paper CSRP 222*, Brighton, England: University of Sussex.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor, Michigan: University of Michigan Press.

Holland, J.H., Reitman, J.S. (1978), "Cognitive systems based on adaptive algorithms", *in Pattern-Directed Inference Systems (Eds. D. Waterman & F. Hayes-Roth)*, New York: Academic Press.

Holub, A.I. (1992), *C + C++ Programming with Objects in C and C++*, New York: McGraw-Hill.

Homaifar, A., Guan, S., Liepins, G. (1993), "A New Approach on the Travelling Salesman Problem by Genetic Algorithms", *in Forrest (1993)*, pp. 460 - 466.

Horner, A., Goldberg, D.E. (1991), "Genetic Algorithms and Computer Assisted Music Composition", *in Belew & Booker (1991)*, pp. 437 - 441.

Huang, D. (1989), "The Context Array Bucket Brigade algorithm: an enhanced approach to credit-apportionment in Classifier Systems", *in Shaffer (1989)*, pp. 311 - 316.

Hulin, M., (1991), "Analysis of Schema Distributions", *in Belew & Booker (1991)*, pp. 204 - 209.

Janikow, C.Z., Michalewicz, Z. (1991), "An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms", *in Belew & Booker (1991)*, pp. 31 - 36.

Johnston, R.J., Gregory, D., Smith, D.M. [Eds.] (1986), *The Dictionary of Human Geography, 2nd edition*, Oxford: Blackwell.

Jones, G., Brown, R.D., Clark, D.E., Willet, P., Glen, R.C. (1993), "Searching Databases of Two and Three-Dimensional Chemical Structures using Genetic Algorithms", *in Forrest (1993)*, pp. 597 - 602.

Juliff, K. (1993), "A Multi-chromosome Genetic Algorithm for Pallet Loading", *in Forrest (1993)*, pp. 467 - 473.

Julstrom, B.A. (1995), "What Have You Done For Me Lately? Adapting Operator Probabilities in a Steady-State Genetic Algorithm",*in Eshelman (1995)*, pp. 81 - 87.

Kado, K., Ross, P., Corne, D. (1995), "A Study of Genetic Algorithm Hybrids for Facility Layout Problems",*in Eshelman (1995)*, pp. 498 - 505.

Kahng, A.B., Moon, B.R. (1995), "Toward More Powerful Recombinations",*in Eshelman (1995)*, pp. 96 - 103.

Kargupta, H., Deb, K., Goldberg, D.E. (1992), "Ordering Genetic Algorithms and Deception", *IlliGAL Report No: 92006 Illinois Genetic Algorithms Laboratory*, University of Illinois:Urbana-Champaign.

Kelly, J.D., Davis, L. (1991), "Hybridizing the Genetic Algorithm and the K Nearest Neighbours Classification Algorithm", *in Belew & Booker (1991)*, pp. 377 - 383.

Khuri, S., Bäck, T., Heitkötter, J. (1994), "The zero/one Multiple Knapsack Problem and Genetic Algorithms, *in Proceedings of the ACM Symposium of Applied Computation, SAC '94.*, ACM Press.

Kido, T., Kitano, H., Nakanishi, M. (1993), "A Hybrid Search for Genetic Algorithms: Combining Genetic Algorithms, TABU search and Simulated Annealing", *in Forrest (1993)*, pp. 641.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983), "Optimisation by Simulated Annealing", *Science*, Vol 220, May 1993, pp. 671 - 680.

Kitano, H., Smith, S.F., Higuchi, T. (1991), "A Parallel Associative Memory Processor for Rule Learning with Genetic Algorithms", *in Belew & Booker (1991)*, pp. 311 - 317.

Koza, J.R. (1991), "Evolving a Computer Program to Generate Random Numbers Using the Genetic Programming Paradigm", *in Belew & Booker (1991)*, pp. 37 - 44.

Koza, J.R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: The MIT Press.

Kusiak, A., Heragu, S.S. (1987), "The facility layout problem", *European Journal of Operational Research 29*, pp. 229 - 251.

Langdon, W. (1995), "Scheduling Planned Maintenance of the National Grid, Research Note RN/3/95", *submitted to AISB Workshop on Evolutionary Computation, April 1995.*

Lee, M.A., Takagi, H. (1993), "Dynamic Control of Genetic Algorithms using Fuzzy Logic Techniques", *in Forrest (1993)*, pp. 76 - 83.

Levenick, J.R. (1995), "Metabits: Generic endogenous crossover control",*in Eshelman (1995)*, pp. 88 - 95.

Lindgren, K. (1991), "Evolutionary Phenomena in Simple Dynamics",*Artificial Life II*, pp. 295 - 312.

Maguire, D.J., Goodchild, M.F., Rhind, D.W. [Ed.] (1991), *Geographical Information Systems, Vol 1: Principles*, New York, NY: Longman.

Mahfoud, S.W. (1991), "An Analysis of Boltzmann Tournament Selection", *IlliGAL Report No: 91007, Illinois Genetic Algorithms Laboratory*, University of Illinois:Urbana-Champaign.

Mahfoud, S.W. (1992), "Crowding and Preselection revisited.", *IlliGAL Report No: 92004 Illinois Genetic Algorithms Laboratory*, University of Illinois:Urbana-Champaign.

Männer, R., Manderick, B., (1992), *Parallel Problem Solving from Nature 2*, North-Holland.

Mauldin, M.L. (1984), "Maintaining Diversity in Genetic Search", *National Conference on Artificial Intelligence*, pp. 247 - 250.

Miller, J.H., Forrest, S. (1989), "The Dynamical Behaviour of Classifier Systems", *in Shaffer (1989)*, pp. 304 - 310.

Mitchell, M., Holland, J.H., Forrest, S. (1994), "When will a Genetic Algorithm outperform Hill Climbing?", *Advances in Neural Information Processing Systems 6*, San Mateo, CA: Morgan Kaufmann, pp. 51 - 59.

Mühlenbein, H., Schomisch, M., Born, J. (1991), "The Parallel Genetic Algorithm as a Function Optimizer",*in Belew & Booker (1991)*, pp. 271 - 278.

Mühlenbein, H., Schlierkamp-Voosen, D. (1993), "Predictive Models for the Breeder Genetic Algorithm: Continuous Parameter Optimisation", *Evolutionary Computation*, 1:25 - 49.

Mühlenbein, H., Schlierkamp-Voosen, D. (1994), "The science of breeding and its application to the Breeder Genetic Algorithm", *Evolutionary Computation*, 1: 335 - 360.

Murnion, S. (1995), "Spatial Analysis using an Unsupervised Neural Network", *submitted April 1995 to Special Neural Networks Edition: Journal of Computers and Geosciences.*

Murnion, S. (1995a), Personal Communication, October 1995.

Murray, R.B. (1993), *C++ Strategies and Tactics*, Reading, MA: Addison-Wesley.

Nakano, R., Yamada, T. (1991), "Conventional Genetic Algorithm for Job Shop Problems",*in Belew & Booker (1991)*, pp. 474 - 479.

Openshaw, S. (1991), "Developing appropriate Spatial Analysis methods for GIS",*in Maguire et al. (1991)*, pp. 389 - 402.

Ordnance Survey (O.S.) (1995), *Ordnance Survey Catalogue 1995*, Crown Copyright ©1994: HMSO.

Orwell, G. (1945), *Animal Farm*, Longman: London, p.92.

Parodi, A., Bonelli, P. (1993), "A New Approach to Fuzzy Classifier Systems", *in Forrest (1993)*, pp. 223 - 230.

Pettey, C.C., Leuze, M.R. (1989), "A Theoretical Investigation of a Parallel Genetic Algorithm", *in Shaffer (1989)*, pp. 398 - 405.

Potter, M.A., De Jong, K.A., Grefenstette, J.J. (1995), "A Coevolutionary Approach to Learning Sequential Decision Rules",*in Eshelman (1995)*, pp. 366 - 372.

Powell, D.J., Tong, S.S., Skolnick, M.M. (1989), "EnGENEous: domain independent, machine learning for design optimisation", *in Shaffer (1989)*, pp. 151 - 159.

Radcliffe, N.J. (1991), "Forma Analysis and Random Respectful Recombination",*in Belew & Booker (1991)*, pp. 222 - 229.

Radcliffe, N.J. (1992), "Genetic Set Recombination",*in Whitley (1992)*.

Rechenberg, I. (1973), *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart, 1973.

Reeves, C.R. (1993), "Using Genetic Algorithms with Small Populations", *in Forrest (1993)*, pp. 92 - 99

Schaffer, J.D. [Ed.] (1989), *Proceedings of the 3rd International Conference on Genetic Algorithms*, San Mateo, CA: Morgan Kauffman.

Schaffer, J.D., Caruana, R.A., Eshelman, L.J., Das, R. (1989a), "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimisation", *in Schaffer (1989)*, pp. 52 - 60.

Schaffer, J.D., Eshelman, L.J. (1991), "On Crossover as an Evolutionary Viable Strategy", *in Belew & Booker (1991)*, pp. 61 - 68.

Schwefel, H.P. (1975), *Evolutionsstrategie und numerische Optimierung.* Dissertation, Technische Universität Berlin, Berlin, May 1975.

Sedbrook, T.A., Wright, H., Wright R. (1991), "Application of a Genetic Classifier for Patient Triage", *in Belew & Booker (1991)*, pp. 334 - 338.

Shaefer, C.G. (1987), "The ARGOT Strategy: Adaptive Representation Genetic Optimizer Technique", *in Grefenstette (1987)*, pp. 50 - 54.

Shaffer, J.D., Morishima, A. (1987), "An Adaptive Crossover Distribution Mechanism for Genetic Algorithms", *in Grefenstette (1987)*, pp. 36 - 40.

Shu, L., Schaeffer, J. (1991), "HCS: Adding Hierarchies to Classifier Systems", *in Belew & Booker (1991)*, pp. 339 - 345.

Sirag, D.J., Weisser, P.T. (1987), "Toward a Unified Thermodynamic Genetic Operator", *in Grefenstette (1987)*, pp. 116 - 122.

Smith, S. F. (1980), *A learning system based on Genetic Algorithms,* Ph.D. Dissertation, University of Pittsburgh.

Smith, S.F. (1983), "Flexible Learning of Problem Solving Heuristics through Adaptive Search", *in Proceedings of 8th International Joint Conference on AI,* Karlsrühe, W. Germany, August 1983, pp. 422 - 425.

Smith, R., Goldberg, D.E. (1990), "Reinforcement Learning with Classifier Systems", *in Proceedings of AI, Simulation and Planning in High Autonomy Systems,* Los Alamitos, CA: IEEE Press, pp. 184 - 192.

Spears, W.M. (1994), "Simple Subpopulation Schemes", *in Proceedings of the 3rd Annual Conference on Evolutionary Programming,* San Diego, CA: World Scientific, pp. 296 - 307.

Spiessens, P., Manderick, B. (1991), "A Massively Parallel Genetic Algorithm: Implementation and First Analysis", *in Belew & Booker (1991)*, pp. 279 - 287.

Srinivas, M., Patnaik, L.M., (1994), "Adaptive Probabilities of Crossver and Mutation in Genetic Algorithms", *IEEE Transactions on Systems, Man and Cybernetics, Vol 24 (4)*, pp. 656 - 667.

Starkweather, T., McDaniel, S., Mathias, K., Whitley, D., Whitley, C. (1991), "A Comparison of Genetic Sequencing Operators", *in Belew & Booker (1991)*, pp. 69 - 76.

Starkweather, T., Whitley, D., Mathias, K. (1992), "Optimization using Distributed Genetic Algorithms", *in Männer & Manderick (1992)*, pp. 176 - 183.

Styblinski, M.A., Tang, T.S. (1990), "Experiments in Nonconvex Optimization: Stochastic Approximation with Function Smoothing and Simulated Annealing", *Neural Networks 3*, pp. 467 - 483.

Suzuki, J. (1993), "A Markov Chain Analysis on a Genetic Algorithm", *in Forrest (1993)*, pp. 146 - 153.

Syswerda, G. (1989), "Uniform Crossover in Genetic Algorithms", *in Schaffer (1989)*, pp. 2 - 9.

Tackett, W.A. (1993), "Genetic Programming for Feature Discovery and Image Discrimination", *in Forrest (1993)*, pp. 303 - 309.

Tam, K.Y. (1992), "Genetic Algorithms, function optimization and facility layout design", *European Journal of Operational Research 63*, pp. 322 - 346.

Tate, D.M., Smith, A.E. (1993), "Expected Allele Coverage and the Role of Mutation in Genetic Algorithms", *in Forrest (1993)*, pp. 31 - 37.

Tate, D.M., Smith, A.E. (1993b), "Genetic Optimization using a Dynamic Penalty Function", *in Forrest (1993)*, pp. 499 - 505.

Thangiah, S.R. (1995), "An Adaptive Clustering Method using a Geometric Shape for Vehicle Routing Problems with Time Windows",*in Eshelman (1995)*, pp. 536 - 544.

Varanelli, J.M., Cohoon, J.P. (1995), "Population-Oriented Simulated Annealing: A Genetic/Thermodynamic Hybrid Approach to Optimisation",*in Eshelman (1995)*, pp. 174 - 182.

von Laszewski, G. (1991), "Intelligent Structural Operators for the k-way Graph Partitioning Problem", *in Belew & Booker (1991)*, pp. 45 - 52.

Watabe, H., Okino, N. (1993), "A Study on Genetic Shape Design", *in Forrest (1993)*, pp. 445 - 450.

Westerdale, T.H. (1989), "A Defense of the Bucket Brigade", *in Shaffer (1989)*, pp. 282 - 290.

White, T., Oppacher, F., (1994), "Adaptive Crossover using Automata", *in Parallel Problem Solving from Nature 3 (1994)*, pp. 229 - 238.

Whitley, D. (1987), "Using Reproductive Evaluation to Improve Genetic Search and Heuristic Discovery", *in Grefenstette (1987)*, pp. 108 - 115.

Whitley, D. (1989), "The GENITOR algorithm and Selection Pressure: Why Rank Based Allocation of Reproductive Trials is best.", *in Shaffer (1989)*, pp. 116 - 121.

Whitley, D. [Ed.] (1992), *Foundations of Genetic Algorithms 2.* San Mateo, CA: Morgan Kauffman.

Whitley, D. (1993), "Genetic Algorithm Theory" *in Tutorial 2: Advanced Genetic Algorithms Topics presented at the 5th International Conference on Genetic Algorithms*, Technical Report tr-103., Fort Collins, CO.: Colorado State University.

Whitley, D., Starkweather, T., Fuquay, D. (1989), "Scheduling Problems and Travelling Salesmen: The Genetic Edge Recombination Operator", *in Schaffer (1989)*, pp. 133 - 140.

Whitley, D., Mathias, K., Rana, S., Dzubera, J. (1995), "Building Better Test Functions",*in Eshelman (1995)*, pp. 239 - 246.

Williams, B.V., Bostock, R., Harget, A.J., Bounds, D. (1994), "A Normalised Chromosome Representation for a Genetic Bumptree Classifier", *Proceedings of IEEE Conference on Evolutionary Computation.*

Wilson, S.W. (1985), "Knowledge growth in an artificial animal", *in Grefenstette (1985)*, pp. 16-23.

Wilson, S.W., Goldberg D.E. (1989), "A Critical Review of Classifier Systems", *in Shaffer (1989)*, pp. 244 - 255.

Appendix A: Network Placement Data

A.1 Introduction

This appendix presents the detailed location data for the network machine placement problems outlined in chapter 6, along with graphical output from Matlab illustrating this data within the hypothetical building.

A.2 The 8 machine problem data

The legal ranges of the 8 machines within the building were as follows:

Machine	*z min.*	*z max.*	*x min.*	*x max.*	*y min.*	*y max.*
1	0	12	5000.0	15000.0	0.0	5000.0
2	20	40	0.0	4000.0	0.0	30000.0
3	80	120	7500.0	12500.0	10000.0	17500.0
4	100	105	0.0	20000.0	7500.0	24500.0
5	68	120	17000.0	20000.0	0.0	20000.0
6	45	80	15000.0	20000.0	22000.0	30000.0
7	15	38	8500.0	12200.0	7500.0	16500.0
8	0	12	5000.0	15000.0	25000.0	30000.0

Table A.1: Legal co-ordinate ranges of the 8 machine network problem.

A.3 The 15 machine problem data

The legal ranges of the 15 machines within the building were as follows:

Machine	*z min.*	*z max.*	*x min.*	*x max.*	*y min.*	*y max.*
1	0	120	0.0	20000.0	0.0	30000.0
2	86	86	0.0	5000.0	0.0	10000.0
3	86	86	5000.0	10000.0	10000.0	18000.0
4	86	86	0.0	20000.0	20000.0	25000.0
5	86	86	15000.0	20000.0	0.0	10000.0
6	0	60	0.0	2500.0	10000.0	25000.0
7	45	67	7500.0	12500.0	12800.0	27000.0
8	0	10	0.0	10000.0	5000.0	25000.0
9	33	90	0.0	16500.0	2000.0	2005.0
10	110	120	5000.0	10000.0	10000.0	23000.0
11	75	100	10000.0	15000.0	0.0	7000.0
12	10	70	10000.0	15000.0	10000.0	15000.0
13	60	84	17000.0	20000.0	12000.0	18000.0
14	60	120	0.0	8000.0	27000.0	30000.0
15	0	60	12000.0	16000.0	27000.0	30000.0

Table A.2: Legal co-ordinate ranges of the 15 machine network problem.

A graphical interpretation of the 15 machine values is given in figure A.1 below:

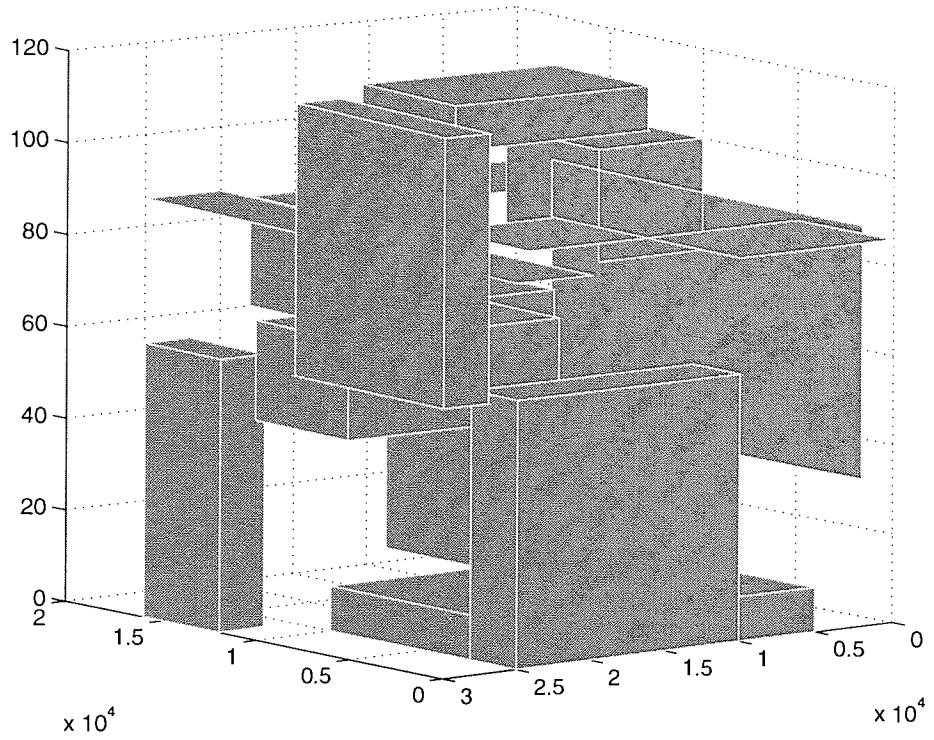


Figure A.1: Graphical display of the machine ranges in the 15 machine network placement problem.

A.4 Graphical display of the 8 machine problem results

The positions of the machines selected as the fittest solution by the best ever Multi-GA run are more clearly seen by examination of those positions in 2D, across x/y and x/z axes, given in figures A2 and A3 respectively.

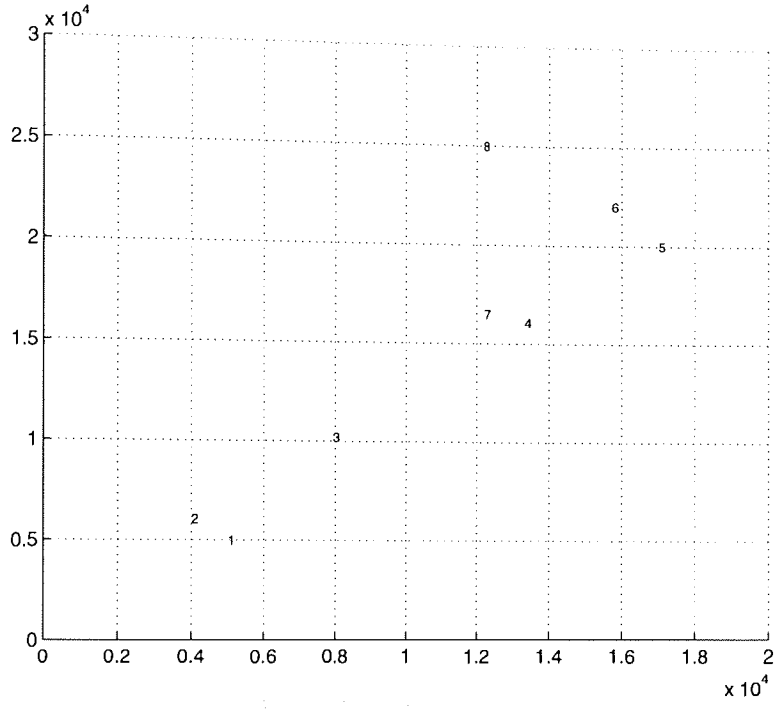


Figure A.2: x/y co-ordinate view of machine positions in the 8 machine problem.

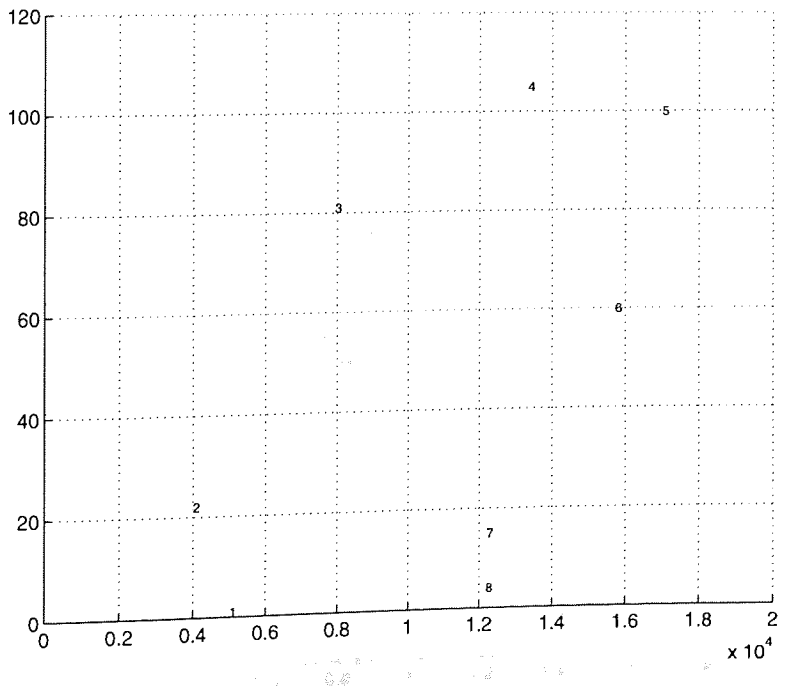


Figure A.3: x/z co-ordinate view of machine positions in the 8 machine problem.

A.5 Graphical display of the 15 machine problem results

The positions of the machines selected as the fittest solution by the best ever Multi-GA run are more clearly seen by examination of those positions in 2D, across x/y and x/z axes, given in figures A4 and A5 respectively.

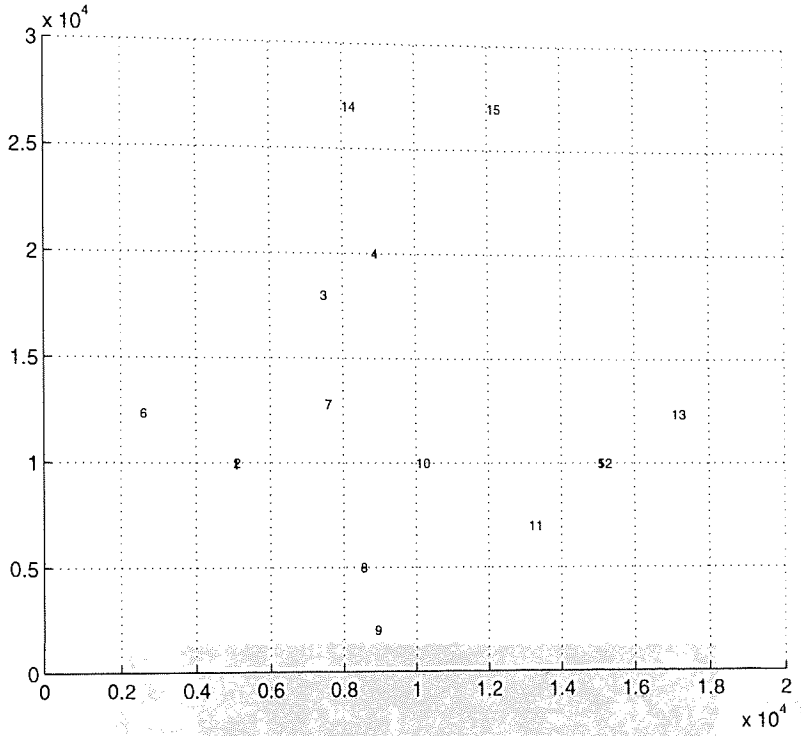


Figure A.4: x/y co-ordinate view of machine positions in the 15 machine problem.

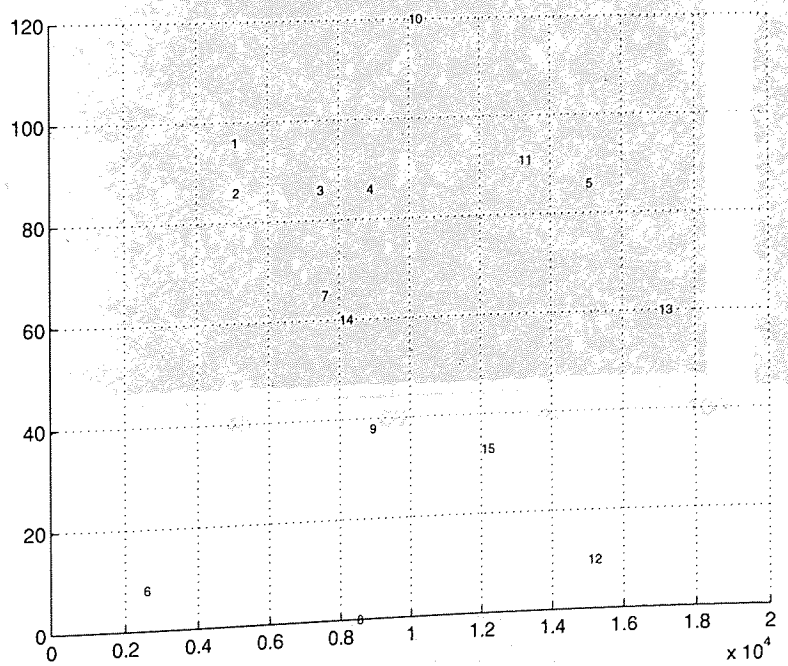


Figure A.5: x/z co-ordinate view of machine positions in the 15 machine problem.

Appendix B: Site Selection Results

Introduction

This appendix contains a number of figures referred to in chapter 6, illustrating the results presented in graphical form, superimposed on the appropriate population grids. The results present only profitable sites found by the best individual at the terminating generation, indicated by a yellow X. Due to the plotting method used by Matlab, the centre point of the X is located at the bottom left point of the square to which the chromosome refers.

Fig. B.1 10 Chromosome addition, engaged at 100% on raw data

This figure shows the results achieved by the best ever result presented in table 6.10 superimposed onto the raw population grid containing all sites.

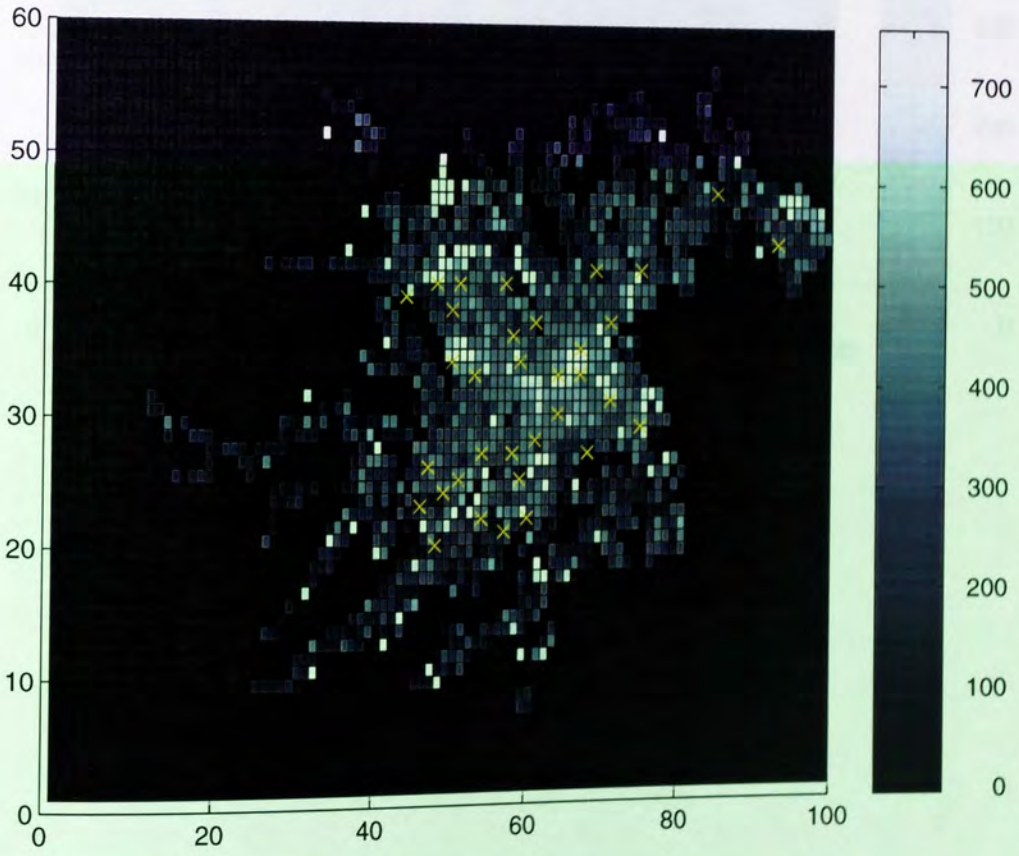


Fig. B.2 10 Chromosome addition, engaged at 75% on raw data

This figure shows the results achieved by the best ever result presented in table 6.12 superimposed onto the raw population grid containing all sites.

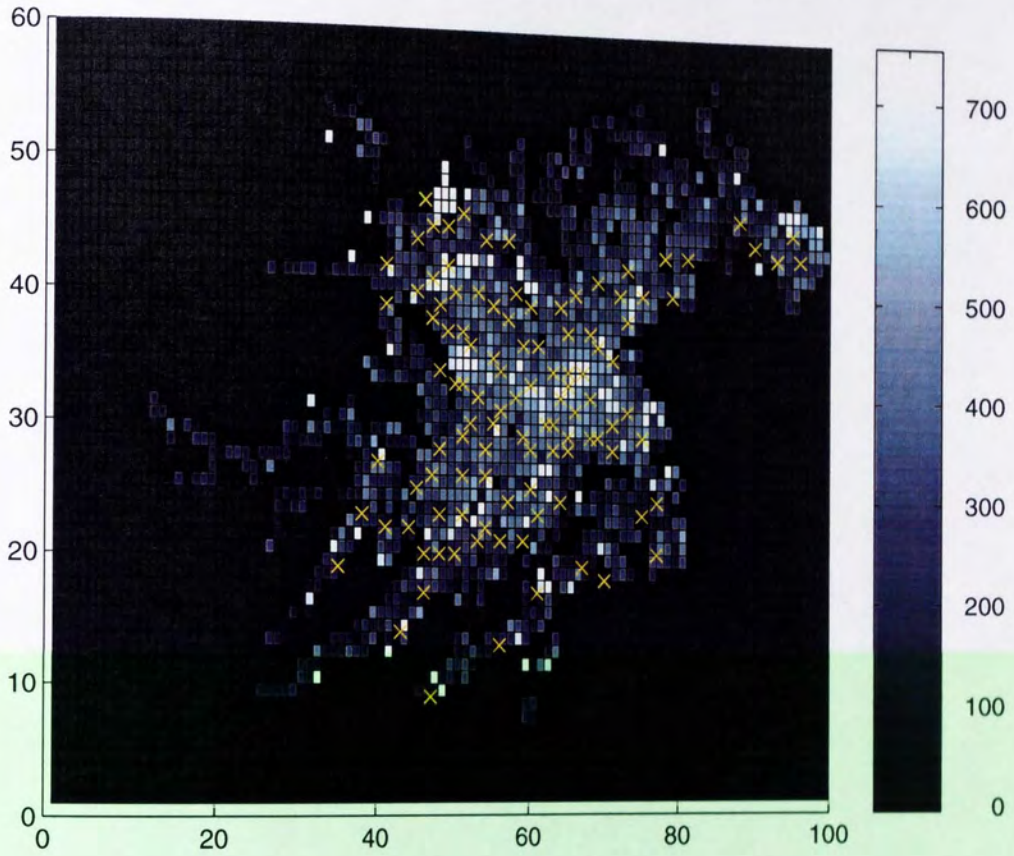


Fig. B.3 10 Chromosome addition, engaged at 50% on raw data

This figure shows the results achieved by the best ever result presented in table 6.12 superimposed onto the raw population grid containing all sites.

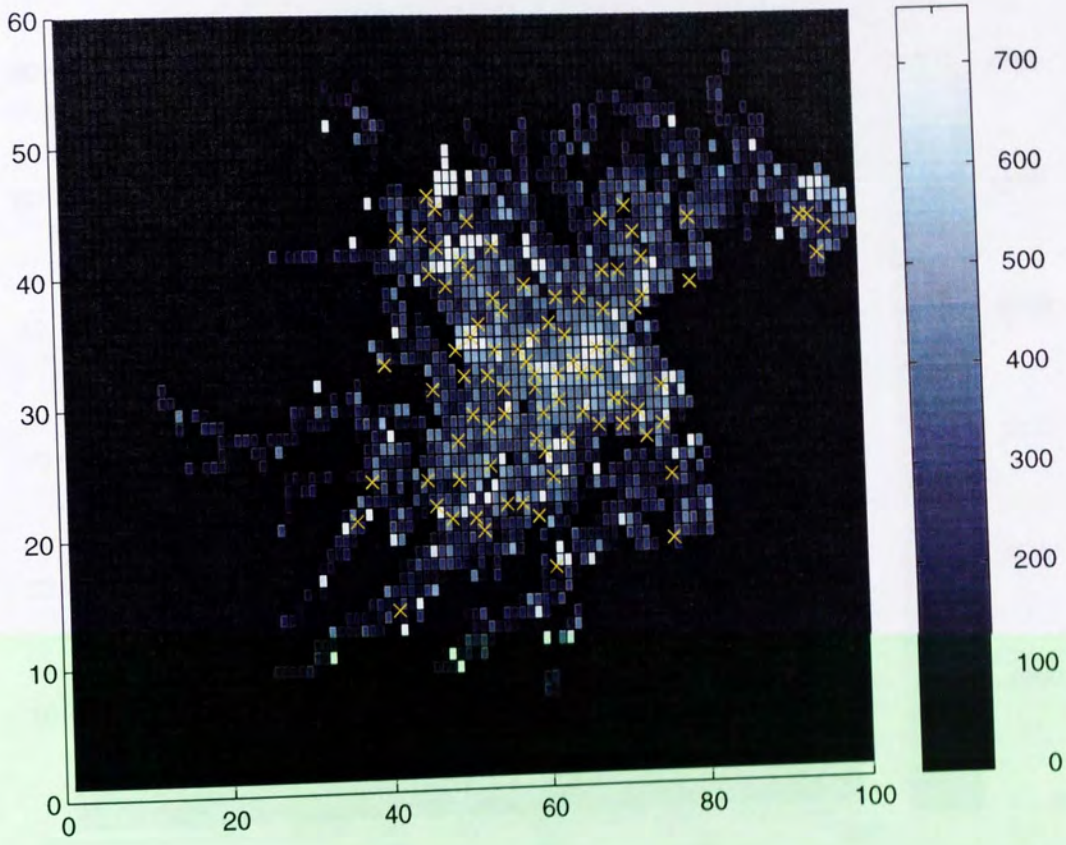
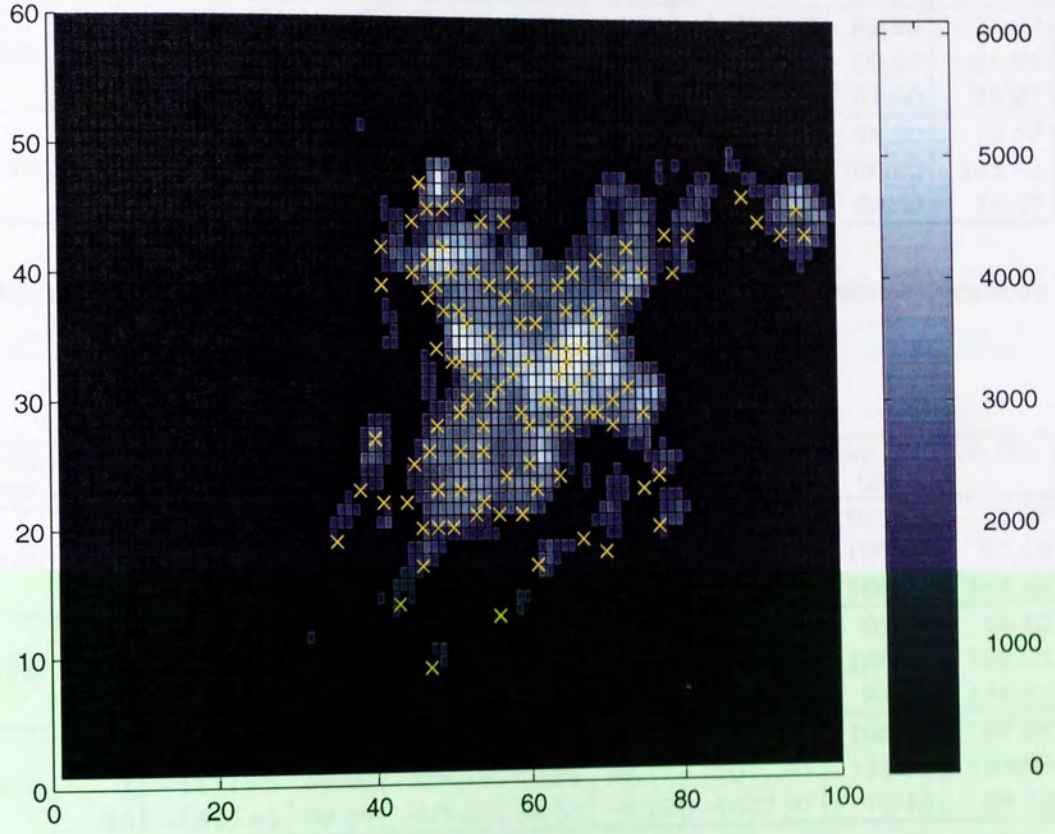


Fig. B.4 10 Chromosome addition, engaged at 50% on smoothed data

This figure shows the results achieved by the best ever result presented in table 6.12 superimposed onto the smoothed population grid containing indicative locations meeting the 2000 site fitness criterion.



Appendix C: Chapter 5 Full Results Tables

C.1. Dynamic Crossover

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-1676.01	1.92	-89.23	71.55	-2.04	-1677.97	82.65	62.30
	1	-1676.86	1.19	-98.51	86.12	-2.04	-1677.97	80.60	93.74
	10	-1676.94	1.29	-67.04	67.17	-2.04	-1677.97	84.95	108.84
75%	0	-1676.61	1.81	-73.70	70.96	-2.04	-1677.97	99.85	77.59
	1	-1676.96	1.38	-100.73	92.15	-2.04	-1677.97	60.55	28.92
	10	-1677.19	3.74	-146.36	70.90	-2.04	-1677.97	51.80	28.27
90%	0	-1676.81	1.94	-76.89	73.29	-2.04	-1677.97	88.60	79.52
	1	-1677.21	1.28	-87.53	84.52	-2.04	-1677.97	100.80	123.75
	10	-1677.52	0.22	-99.26	89.86	-2.04	-1677.97	65.80	53.57

Table C.1: 2 variable F7 dynamic crossover results for direct fitness distance calculation with 75% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-1654.26	47.20	-86.19	73.50	-2.30	-1677.97	105.95	73.36
	1	-1633.08	59.02	-89.71	76.21	-2.17	-1677.97	100.80	97.50
	10	-1625.79	62.26	-150.41	97.70	-2.04	-1677.97	160.20	143.42
75%	0	-1618.64	73.22	-111.43	71.90	-2.17	-1677.97	98.65	59.42
	1	-1608.60	70.67	-98.69	74.22	-4.16	-1677.97	120.50	129.93
	10	-1649.75	46.60	-119.20	88.46	-4.02	-1677.97	98.10	116.26
90%	0	-1626.16	59.35	-146.59	88.17	-2.04	-1677.97	109.15	87.83
	1	-1611.88	58.99	-128.28	80.96	-2.17	-1677.97	134.55	135.25
	10	-1625.40	62.46	-93.35	90.51	-3.20	-1677.97	78.45	65.52

Table C.2: 2 variable F7 dynamic crossover results for direct fitness distance calculation with 0% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-1677.44	1.07	-88.39	73.34	-2.04	-1677.97	48.35	22.61
	1	-1676.86	3.39	-87.04	72.61	-2.04	-1677.97	42.15	33.80
	10	-1677.88	0.11	-97.53	82.50	-2.04	-1677.97	43.15	46.94
75%	0	-1677.61	0.24	-108.34	64.11	-2.04	-1677.97	46.55	23.03
	1	-1677.64	0.90	-98.36	64.10	-2.05	-1677.97	30.60	12.00
	10	-1677.09	1.08	-83.01	63.17	-2.04	-1677.97	26.00	9.34
90%	0	-1677.62	0.00	-101.83	60.31	-2.04	-1677.97	34.90	16.23
	1	-1677.48	1.17	-113.01	91.07	-2.04	-1677.97	30.20	9.80
	10	-1677.58	1.17	-84.18	59.24	-2.04	-1677.97	24.30	8.42

Table C.3: 2 variable F7 dynamic crossover results for inverse fitness distance calculation with 75% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-1671.37	14.57	-119.37	91.40	-2.04	-1677.97	46.00	33.95
	1	-1635.71	67.36	-114.81	74.31	-2.17	-1677.97	43.20	32.25
	10	-1635.07	55.93	-123.95	83.81	-2.30	-1677.97	50.30	95.50
75%	0	-1622.00	77.29	-88.28	85.40	-2.04	-1677.97	46.25	37.25
	1	-1659.19	41.92	-145.82	97.14	-2.04	-1677.97	34.25	10.09
	10	-1636.23	56.67	-118.73	88.69	-2.04	-1677.97	64.75	102.88
90%	0	-1653.52	47.41	-109.48	68.81	-8.97	-1677.97	76.45	114.25
	1	-1671.07	25.62	-92.01	53.32	-2.07	-1677.97	54.20	65.76
	10	-1665.61	35.43	-120.48	66.21	-2.04	-1677.97	45.85	86.84

Table C.4: 2 variable F7 dynamic crossover results for inverse fitness distance calculation with 0% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-180.39	0.09	-67.03	8.03	-56.87	-180.50	98.95	102.94
	1	-180.40	0.06	-66.49	5.41	-55.85	-180.49	77.55	121.38
	10	-180.35	0.14	-65.12	5.07	-56.01	-180.50	77.50	122.92
75%	0	-180.37	0.12	-68.91	7.09	-55.79	-180.50	96.85	104.92
	1	-180.37	0.08	-67.41	5.21	-58.44	-180.49	88.65	125.28
	10	-180.38	0.17	-69.88	5.18	-59.24	-180.49	85.50	130.42
90%	0	-180.46	0.06	-65.53	1.71	-59.70	-180.49	67.15	30.93
	1	-180.42	0.10	-66.27	0.57	-66.10	-180.43	147.75	84.00
	10	-180.35	0.00	-66.21	1.70	-60.91	-180.43	35.30	14.93

Table C.5: 2 variable F8 dynamic crossover results for direct fitness distance calculation with 75% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-180.31	0.16	-69.13	8.27	-56.21	-180.49	93.00	69.57
	1	-180.25	0.19	-69.93	6.42	-56.21	-180.50	68.90	57.65
	10	-180.30	0.15	-68.90	8.54	-56.21	-180.49	55.25	44.67
75%	0	-180.28	0.14	-69.20	6.34	-55.35	-180.47	175.60	148.36
	1	-180.32	0.19	-70.87	7.31	-55.63	-180.49	158.30	129.79
	10	-180.29	0.20	-71.49	5.34	-63.66	-180.49	133.85	141.07
90%	0	-180.35	0.00	-73.15	0.00	-73.15	-180.35	87.00	0.00
	1	-180.49	0.05	-73.15	0.00	-73.15	-180.49	41.00	0.00
	10	-180.41	0.00	-73.15	0.00	-73.15	-180.41	50.00	0.00

Table C.6: 2 variable F8 dynamic crossover results for direct fitness distance calculation with 0% individual crossover.

Chrom Xover	El'tm	Average Best	St. Dev.	Average Worst	St. Dev.	Worst Ever	Best Ever	Avg. Best Gen.	St. Dev.
60%	0	-180.47	0.00	-66.14	7.03	-57.30	-180.50	40.90	21.17
	1	-180.48	0.12	-66.45	6.81	-55.41	-180.50	41.05	28.70
	10	-180.44	0.07	-67.42	5.46	-57.05	-180.50	26.20	15.57
75%	0	-180.46	0.10	-67.43	4.79	-58.43	-180.50	50.65	27.68
	1	-180.48	0.12	-67.52	6.83	-56.16	-180.50	47.05	32.35
	10	-180.48	0.06	-67.13	4.58	-57.27	-180.50	53.60	80.82
90%	0	-180.48	0.11	-66.30	5.21	-54.93	-180.50	65.70	88.09
	1	-180.48	0.07	-66.33	5.48	-58.08	-180.50	83.70	123.71
	10	-180.46	0.11	-67.61	6.24	-58.01	-180.49	81.95	133.97

Table C.7: 2 variable F8 dynamic crossover results for inverse fitness distance calculation with 75% individual crossover.

Chrom Xover	El'tm	Average Best	St. Dev.	Average Worst	St. Dev.	Worst Ever	Best Ever	Avg. Best Gen.	St. Dev.
60%	0	-180.47	0.00	-69.60	9.67	-56.20	-180.50	93.50	119.35
	1	-180.47	0.10	-67.66	6.62	-57.29	-180.50	47.15	27.96
	10	-180.44	0.13	-67.83	7.91	-56.14	-180.50	32.50	40.54
75%	0	-180.45	0.07	-68.73	6.77	-56.76	-180.49	67.65	101.53
	1	-180.46	0.13	-66.25	7.00	-54.18	-180.50	38.90	17.04
	10	-180.46	0.03	-65.49	6.06	-57.32	-180.50	25.60	9.87
90%	0	-180.43	0.12	-72.33	3.53	-57.27	-180.50	302.40	91.87
	1	-180.49	0.14	-72.03	3.07	-60.62	-180.50	36.55	4.81
	10	-180.43	0.13	-73.11	1.67	-68.68	-180.49	30.40	2.04

Table C.8: 2 variable F8 dynamic crossover results for inverse fitness distance calculation with 0% individual crossover.

Chrom Xover	El'tm	Average Best	St. Dev.	Average Worst	St. Dev.	Worst Ever	Best Ever	Avg. Best Gen.	St. Dev.
60%	0	-4703.22	149.57	676.09	191.58	1101.87	-5014.50	407.70	94.33
	1	-4719.32	168.38	753.01	253.73	1376.31	-5023.92	336.95	67.93
	10	-4743.73	133.10	768.63	220.14	1105.55	-5017.69	344.50	67.93
75%	0	-4708.18	173.92	684.24	209.24	1089.45	-4933.06	430.00	64.48
	1	-4702.17	167.48	708.11	183.14	1178.31	-5017.82	335.35	59.15
	10	-4704.02	228.28	810.51	315.87	1619.49	-5016.96	309.40	63.14
90%	0	-4620.68	159.24	736.93	191.56	1306.81	-4892.87	402.95	68.85
	1	-4730.17	175.73	679.69	146.01	957.11	-5018.98	352.55	49.72
	10	-4691.05	167.27	680.56	190.73	1223.40	-4929.98	316.95	45.20

Table C.9: 10 variable F7 dynamic crossover results for direct fitness distance calculation with 75% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-4080.17	357.49	777.90	278.17	1479.91	-4833.53	475.65	41.01
	1	-4079.80	259.53	629.52	287.43	1450.07	-4485.72	484.05	16.10
	10	-4153.58	204.96	646.64	193.43	1113.82	-4501.45	464.75	40.00
75%	0	-4036.30	303.11	728.88	253.83	1231.39	-4573.20	482.20	31.25
	1	-4165.12	269.31	647.81	153.17	1103.69	-4641.08	484.40	11.97
	10	-4093.35	272.14	737.89	273.97	1378.84	-4476.53	461.15	34.05
90%	0	-4123.00	333.71	803.04	273.93	1293.58	-4840.40	470.90	34.24
	1	-4050.74	346.97	792.57	315.05	1599.92	-5017.19	464.65	32.93
	10	-4049.48	296.00	707.29	219.18	1058.87	-4707.77	439.95	55.94

Table C.10: 10 variable F7 dynamic crossover results for direct fitness distance calculation with 0% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-4737.47	180.12	848.29	186.25	1245.07	-5022.30	362.65	113.69
	1	-4742.36	169.84	724.17	216.22	1168.32	-5018.98	306.90	42.55
	10	-4669.78	188.81	723.66	237.41	1231.38	-4903.45	271.65	70.81
75%	0	-4680.43	146.40	746.79	104.51	939.73	-4908.89	363.65	86.63
	1	-4754.75	186.88	734.29	281.55	1792.65	-5018.85	286.20	83.89
	10	-4711.92	137.17	825.03	324.21	1730.76	-5020.87	241.95	67.91
90%	0	-4794.70	133.26	632.34	196.55	943.84	-5012.48	357.55	83.35
	1	-4719.03	138.00	716.57	233.41	1189.57	-4906.64	256.80	46.78
	10	-4771.21	120.93	630.58	204.52	983.72	-4937.23	278.00	99.86

Table C.11: 10 variable F7 dynamic crossover results for inverse fitness distance calculation with 75% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-4182.23	288.84	736.30	245.63	1361.00	-4673.25	400.80	84.21
	1	-4253.49	294.08	823.82	320.39	1418.85	-4695.11	319.30	75.15
	10	-4181.14	304.47	698.90	143.44	1081.50	-4718.93	310.50	80.66
75%	0	-4205.79	220.92	721.12	197.18	1270.00	-4566.98	410.25	71.02
	1	-4225.30	236.97	671.84	185.12	1071.75	-4869.00	345.40	89.12
	10	-4083.70	261.65	789.90	295.34	1469.65	-4640.13	284.65	78.12
90%	0	-4339.10	339.95	752.00	158.29	1188.22	-4897.17	403.20	75.28
	1	-4173.48	299.52	685.10	161.77	1030.34	-4627.38	296.80	47.38
	10	-4112.71	238.47	776.00	234.42	1409.85	-4539.89	238.45	89.00

Table C.12: 10 variable F7 dynamic crossover results for inverse fitness distance calculation with 0% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-180.41	0.05	-118.37	1.13	-117.02	-180.49	359.30	63.89
	1	-180.42	0.08	-117.71	0.59	-116.99	-180.50	253.55	69.40
	10	-180.44	0.00	-118.06	0.92	-116.99	-180.50	239.70	70.41
75%	0	-180.41	0.04	-117.85	1.02	-116.97	-180.50	374.85	55.84
	1	-180.45	0.00	-117.89	1.47	-116.99	-180.50	216.90	34.48
	10	-180.41	0.00	-118.03	0.77	-117.01	-180.48	188.00	51.10
90%	0	-180.43	0.06	-117.91	1.20	-116.99	-180.49	375.55	52.24
	1	-180.43	0.00	-118.01	1.06	-116.99	-180.50	257.15	53.92
	10	-180.44	0.00	-118.06	1.32	-116.99	-180.50	200.75	45.76

Table C.13: 10 variable F8 dynamic crossover results for direct fitness distance calculation with 75% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-180.34	0.18	-118.28	1.23	-117.00	-180.49	438.05	44.22
	1	-180.41	0.05	-118.57	1.43	-116.98	-180.50	320.10	64.26
	10	-180.35	0.08	-117.97	0.79	-117.02	-180.47	304.20	90.27
75%	0	-180.38	0.10	-117.92	1.08	-116.91	-180.48	415.80	60.24
	1	-180.39	0.07	-117.82	0.95	-116.93	-180.49	341.55	68.79
	10	-180.34	0.09	-118.00	1.07	-116.99	-180.49	300.35	58.12
90%	0	-180.36	0.09	-117.72	0.84	-116.95	-180.49	408.00	49.67
	1	-180.36	0.11	-117.82	0.98	-116.98	-180.49	305.45	74.12
	10	-180.37	0.16	-118.25	1.17	-117.01	-180.47	290.40	63.92

Table C.14: 10 variable F8 dynamic crossover results for direct fitness distance calculation with 0% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-180.41	0.13	-117.70	1.17	-116.97	-180.49	435.35	46.99
	1	-180.47	0.04	-117.90	1.09	-117.00	-180.50	292.50	68.48
	10	-180.47	0.10	-118.85	1.24	-116.96	-180.50	184.65	64.91
75%	0	-180.43	0.04	-117.69	0.97	-116.90	-180.49	417.30	80.61
	1	-180.47	0.04	-117.98	1.52	-116.91	-180.50	310.20	74.17
	10	-180.46	0.08	-117.84	0.90	-116.98	-180.49	208.30	57.38
90%	0	-180.42	0.00	-117.61	0.77	-116.99	-180.49	429.10	55.74
	1	-180.45	0.06	-118.04	1.33	-116.99	-180.50	324.80	72.83
	10	-180.48	0.00	-118.13	1.29	-116.99	-180.49	172.15	43.86

Table C.15: 10 variable F8 dynamic crossover results for inverse fitness distance calculation with 75% individual crossover.

Chrom Xover	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	0	-180.42	0.11	-117.81	0.80	-117.00	-180.50	334.95	117.63
	1	-180.46	0.08	-118.12	1.34	-116.99	-180.50	260.80	70.04
	10	-180.47	0.06	-118.03	1.06	-116.97	-180.50	167.65	49.72
75%	0	-180.41	0.02	-117.73	1.11	-116.90	-180.48	382.05	59.98
	1	-180.46	0.02	-117.73	0.95	-116.96	-180.49	283.15	83.36
	10	-180.45	0.03	-117.81	0.80	-116.99	-180.50	196.00	85.70
90%	0	-180.42	0.00	-117.66	1.01	-116.97	-180.48	406.90	55.05
	1	-180.47	0.00	-118.01	1.04	-116.93	-180.49	288.35	79.02
	10	-180.46	0.05	-118.04	1.34	-116.98	-180.49	187.30	59.48

Table C.16: 10 variable F8 dynamic crossover results for inverse fitness distance calculation with 0% individual crossover.

C.2. Dynamic Mutation

Chrom Xover	*Max Mtn.*	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
60%	10%	0	-1677.54	0.85	-26.90	48.27	-2.04	-1677.97	45.00	18.64
		1	-1677.96	0.00	-27.02	48.97	-2.04	-1677.97	54.20	96.44
		10	-1677.80	0.46	-27.13	50.75	-2.04	-1677.97	40.50	44.83
	30%	0	-1677.97	0.00	-2.38	1.00	-2.04	-1677.97	140.00	104.43
		1	-1677.97	0.00	-2.04	0.01	-2.04	-1677.97	44.55	13.79
		10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	45.55	80.37
	50%	0	-1677.97	0.00	-2.25	0.23	-2.05	-1677.97	283.80	129.73
		1	-1677.97	0.00	-2.04	0.02	-2.04	-1677.97	67.45	22.76
		10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	28.80	6.67
75%	10%	0	-1677.02	4.61	-9.58	28.03	-2.04	-1677.97	120.35	129.93
		1	-1677.81	0.00	-5.03	12.33	-2.04	-1677.97	29.90	8.23
		10	-1677.64	1.03	-15.69	34.69	-2.04	-1677.97	24.55	6.38
	30%	0	-1677.94	0.00	-2.54	1.19	-2.04	-1677.97	110.05	96.22
		1	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	41.25	14.88
		10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	45.75	80.69
	50%	0	-1677.97	0.00	-2.33	0.24	-2.04	-1677.97	222.20	115.50
		1	-1677.97	0.00	-2.05	0.02	-2.04	-1677.97	71.75	17.80
		10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	29.00	7.31
90%	10%	0	-1677.65	0.98	-2.36	0.95	-2.04	-1677.97	72.75	99.31
		1	-1677.81	0.00	-8.69	28.29	-2.04	-1677.97	35.45	9.37
		10	-1677.96	0.00	-9.85	28.55	-2.04	-1677.97	67.10	130.43
	30%	0	-1677.96	0.00	-2.21	0.73	-2.04	-1677.97	113.45	58.19
		1	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	43.55	12.30
		10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	24.55	5.62
	50%	0	-1677.97	0.00	-2.25	0.24	-2.04	-1677.97	239.95	152.58
		1	-1677.97	0.00	-2.05	0.02	-2.04	-1677.97	70.00	19.99
		10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	28.80	6.62

Table C.17: 2 variable F7 dynamic mutation results for direct rank based calculation.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-1677.50	0.91	-30.22	49.44	-2.04	-1677.97	109.75	135.94
	1	-1677.81	0.00	-8.73	15.79	-2.04	-1677.97	39.15	30.06
	10	-1677.94	0.20	-16.76	38.82	-2.04	-1677.97	24.60	6.51
30%	0	-1677.96	0.00	-2.35	0.96	-2.04	-1677.97	62.60	72.45
	1	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	31.90	9.31
	10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	32.60	21.88
50%	0	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	68.50	63.71
	1	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	35.70	9.70
	10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	41.70	63.24

Table C.18: 2 variable F7 dynamic mutation results for inverse rank based calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-1671.64	25.75	-97.53	74.94	-5.35	-1677.97	31.85	11.31
	1	-1677.44	1.15	-123.02	71.39	-14.85	-1677.97	28.20	8.00
	10	-1677.68	0.00	-128.36	88.83	-3.29	-1677.97	21.15	5.43
30%	0	-1671.96	22.59	-117.72	81.35	-6.19	-1677.97	27.95	14.35
	1	-1671.49	25.71	-99.18	72.58	-5.11	-1677.97	30.30	5.06
	10	-1677.60	0.82	-165.58	77.60	-10.73	-1677.97	21.40	3.63
50%	0	-1677.51	1.01	-100.37	69.69	-14.07	-1677.97	33.50	11.75
	1	-1677.75	0.00	-111.22	59.19	-4.17	-1677.97	25.55	5.71
	10	-1677.55	0.65	-129.06	65.41	-19.72	-1677.97	19.25	4.77

Table C.19: 2 variable F7 dynamic mutation results for direct fitness distance calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-1677.58	0.84	-2.67	1.62	-2.04	-1677.97	78.30	77.43
	1	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	36.70	9.80
	10	-1677.65	0.98	-2.35	0.96	-2.04	-1677.97	30.10	21.80
30%	0	-1677.97	0.38	-2.11	0.10	-2.04	-1677.97	154.30	65.12
	1	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	60.75	16.82
	10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	31.35	17.49
50%	0	-1677.96	0.00	-2.57	0.72	-2.06	-1677.97	217.15	144.43
	1	-1677.97	0.00	-2.05	0.03	-2.04	-1677.97	121.90	58.14
	10	-1677.97	0.00	-2.04	0.00	-2.04	-1677.97	34.20	9.49

Table C.20: 2 variable F7 dynamic mutation results for inverse fitness distance calculation at 75% chromosome crossover.

Chrom Xover	Max Mtn.	El'tn	Average Best	St. Dev.	Average Worst	St. Dev.	Worst Ever	Best Ever	Avg. Best Gen.	St. Dev.
60%	10%	0	-180.48	0.06	-66.30	5.25	-57.09	-180.50	69.20	81.85
		1	-180.49	0.06	-66.07	5.65	-54.56	-180.50	71.05	96.35
		10	-180.48	0.10	-65.91	4.83	-57.09	-180.50	46.65	56.10
	30%	0	-180.49	0.11	-65.04	6.11	-54.96	-180.50	187.40	130.44
		1	-180.50	0.08	-65.13	5.76	-54.96	-180.50	93.25	75.48
		10	-180.49	0.05	-65.94	6.16	-54.96	-180.50	86.75	127.50
	50%	0	-180.49	0.03	-65.11	6.35	-57.04	-180.50	246.00	149.06
		1	-180.50	0.07	-64.96	6.53	-56.63	-180.50	93.60	62.12
		10	-180.50	0.08	-65.87	6.32	-55.82	-180.50	77.05	93.38
75%	10%	0	-180.48	0.10	-65.96	6.26	-54.96	-180.50	60.00	73.66
		1	-180.49	0.07	-65.96	6.26	-54.96	-180.50	40.85	20.05
		10	-180.48	0.13	-66.79	6.01	-54.96	-180.50	44.15	70.26
	30%	0	-180.50	0.04	-65.65	5.59	-57.09	-180.50	196.10	130.94
		1	-180.50	0.12	-65.61	5.55	-57.09	-180.50	94.45	91.61
		10	-180.50	0.10	-65.73	5.94	-57.09	-180.50	89.60	118.62
	50%	0	-180.50	0.00	-65.79	6.47	-56.50	-180.50	294.70	121.32
		1	-180.50	0.05	-65.34	5.83	-57.03	-180.50	117.25	61.43
		10	-180.50	0.09	-66.11	6.04	-57.03	-180.50	61.25	77.21
90%	10%	0	-180.48	0.09	-64.73	5.40	-56.77	-180.50	81.00	106.00
		1	-180.49	0.05	-64.23	5.32	-56.77	-180.50	68.85	92.98
		10	-180.49	0.10	-65.87	5.77	-57.09	-180.50	56.60	103.02
	30%	0	-180.50	0.08	-66.10	6.32	-56.22	-180.50	220.70	127.04
		1	-180.50	0.07	-66.10	6.32	-56.22	-180.50	82.05	58.60
		10	-180.50	0.12	-66.75	6.04	-57.09	-180.50	77.45	106.68
	50%	0	-180.50	0.00	-65.60	6.12	-57.03	-180.50	296.40	121.10
		1	-180.50	0.06	-65.60	6.12	-57.03	-180.50	160.30	82.81
		10	-180.50	0.13	-65.76	6.29	-57.03	-180.50	56.00	83.87

Table C.21: 2 variable F8 dynamic mutation results for direct rank based calculation.

Max Mtn.	El'tn	Average Best	St. Dev.	Average Worst	St. Dev.	Worst Ever	Best Ever	Avg. Best Gen.	St. Dev.
10%	0	-180.48	0.09	-65.52	4.32	-56.40	-180.50	64.45	61.14
	1	-180.49	0.04	-66.34	6.01	-58.53	-180.50	55.65	94.24
	10	-180.49	0.06	-70.42	7.71	-56.25	-180.50	46.65	74.46
30%	0	-180.49	0.03	-66.38	6.56	-56.92	-180.50	131.90	106.61
	1	-180.49	0.11	-66.25	5.80	-57.68	-180.50	122.65	134.05
	10	-180.49	0.11	-65.24	4.79	-58.30	-180.50	188.50	161.89
50%	0	-180.50	0.09	-66.51	5.30	-54.54	-180.50	142.75	115.02
	1	-180.50	0.11	-64.19	5.90	-55.48	-180.50	96.50	94.63
	10	-180.50	0.09	-68.18	6.29	-58.42	-180.50	91.40	105.53

Table C.22: 2 variable F8 dynamic mutation results for inverse rank based calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-180.45	0.00	-67.45	5.22	-56.54	-180.49	48.10	81.08
	1	-180.47	0.07	-66.62	6.34	-55.95	-180.50	33.00	5.98
	10	-180.47	0.10	-66.18	5.58	-57.56	-180.50	31.40	35.81
30%	0	-180.45	0.08	-66.69	8.00	-54.61	-180.50	35.80	28.72
	1	-180.48	0.11	-66.39	7.32	-54.89	-180.50	38.05	16.83
	10	-180.47	0.09	-67.75	6.36	-57.16	-180.50	26.55	19.72
50%	0	-180.47	0.04	-65.58	5.47	-55.12	-180.50	32.60	9.91
	1	-180.47	0.10	-65.31	5.54	-56.46	-180.50	30.90	11.24
	10	-180.46	0.00	-67.98	5.62	-56.30	-180.50	27.80	23.29

Table C.23: 2 variable F8 dynamic mutation results for direct fitness distance calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-180.48	0.13	-68.29	6.92	-55.04	-180.50	137.85	117.49
	1	-180.49	0.10	-64.93	5.89	-54.38	-180.50	82.30	90.57
	10	-180.48	0.06	-66.28	7.21	-55.95	-180.50	54.15	90.53
30%	0	-180.50	0.00	-67.26	5.76	-58.13	-180.50	231.70	133.51
	1	-180.50	0.05	-66.19	6.52	-54.18	-180.50	157.20	117.82
	10	-180.50	0.08	-66.16	5.37	-56.36	-180.50	77.85	55.89
50%	0	-180.49	0.06	-66.46	7.07	-54.61	-180.50	268.30	129.07
	1	-180.50	0.12	-64.47	5.00	-54.48	-180.50	214.45	120.62
	10	-180.50	0.07	-65.01	6.46	-55.28	-180.50	90.10	94.28

Table C.24: 2 variable F8 dynamic mutation results for inverse fitness distance calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-4854.05	117.22	761.48	248.23	1229.99	-5022.12	362.30	101.72
	1	-4921.26	101.97	776.29	245.51	1229.99	-5029.65	459.20	31.50
	10	-4924.08	98.44	765.09	219.55	1229.99	-5029.83	400.50	90.46
30%	0	-4855.29	55.50	865.04	228.88	1229.99	-4983.87	384.05	88.70
	1	-4950.10	68.18	760.87	233.54	1229.99	-5019.45	429.80	52.45
	10	-5012.51	35.94	787.72	225.89	1229.99	-5029.77	485.30	13.19
50%	0	-4381.06	109.04	1127.44	184.58	1579.17	-4610.29	225.70	112.32
	1	-4762.79	155.32	988.75	145.38	1236.62	-4968.24	398.40	73.52
	10	-5008.80	34.81	778.99	225.97	1229.99	-5028.88	475.45	20.68

Table C.25: 10 variable F7 dynamic mutation results for direct rank calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-4900.97	79.27	806.06	204.16	1229.99	-5025.33	455.15	70.62
	1	-4891.56	131.14	867.38	243.16	1383.19	-5028.53	421.60	67.45
	10	-4930.90	104.66	806.09	220.44	1291.00	-5028.67	377.35	101.54
30%	0	-4973.31	66.67	829.28	199.17	1361.00	-5027.06	393.75	75.26
	1	-5025.21	5.54	844.88	147.49	1184.48	-5029.80	470.50	23.55
	10	-5009.52	40.53	764.11	148.71	982.25	-5029.83	343.10	106.28
50%	0	-4975.80	38.88	882.18	159.95	1444.46	-5006.29	146.10	100.82
	1	-5025.66	0.00	899.11	91.52	1032.81	-5029.19	461.80	16.82
	10	-5026.18	3.66	847.85	117.47	1123.49	-5026.64	448.75	40.91

Table C.26: 10 variable F7 dynamic mutation results for inverse rank calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-4615.54	194.00	793.22	228.41	1178.05	-4909.79	95.70	23.56
	1	-4577.87	177.80	747.88	221.66	1255.98	-4908.47	91.55	7.69
	10	-4612.34	205.98	741.33	249.68	1269.04	-5005.33	65.70	4.65
30%	0	-4674.82	166.18	786.84	222.56	1203.18	-5024.28	109.40	20.02
	1	-4682.59	159.06	843.80	258.35	1454.99	-5016.68	96.15	6.95
	10	-4591.07	202.09	765.52	270.62	1679.76	-4940.04	64.60	5.96
50%	0	-4693.79	178.51	832.02	187.38	1309.60	-4983.30	107.70	24.60
	1	-4665.67	168.22	777.41	138.84	1101.46	-5017.42	100.30	8.57
	10	-4632.70	206.42	753.91	181.80	1323.10	-5017.20	67.15	8.55

Table C.27: 10 variable F7 dynamic mutation results for direct fitness distance calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-4810.95	132.47	865.01	209.95	1330.81	-5016.77	364.50	109.18
	1	-5020.59	11.54	902.21	286.00	1599.81	-5029.17	473.50	29.42
	10	-4993.54	61.41	798.12	224.48	1301.27	-5029.83	408.70	83.55
30%	0	-4596.38	70.22	1205.34	213.53	1771.65	-4702.81	325.70	97.01
	1	-4950.59	50.57	1189.29	211.25	1659.94	-5020.32	446.20	58.67
	10	-5017.67	25.57	942.65	255.53	1531.67	-5029.59	481.80	17.25
50%	0	-4117.96	160.46	1373.48	186.43	1770.61	-4652.17	299.25	121.77
	1	-4789.00	112.14	1505.34	143.18	1785.32	-4959.91	432.40	55.48
	10	-4985.61	52.84	1472.53	210.06	1780.37	-5025.99	467.45	30.97

Table C.28: 10 variable F7 dynamic mutation results for inverse fitness distance calculation at 75% chromosome crossover.

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-180.17	0.05	-117.60	1.06	-116.98	-180.37	270.90	142.70
	1	-180.39	0.00	-117.60	1.06	-116.98	-180.47	401.65	83.01
	10	-180.49	0.00	-117.31	0.40	-116.98	-180.50	332.15	104.75
30%	0	-180.15	0.09	-117.64	1.13	-116.98	-180.24	281.25	128.99
	1	-180.22	0.09	-117.74	1.12	-116.92	-180.36	370.85	99.81
	10	-180.47	0.11	-117.65	1.14	-116.98	-180.49	474.80	21.36
50%	0	-180.11	0.11	-117.29	0.56	-116.90	-180.25	270.60	139.70
	1	-180.13	0.11	-117.24	0.53	-116.90	-180.25	281.45	127.50
	10	-180.39	0.00	-117.56	0.67	-116.98	-180.45	395.60	84.36

Table C.29: **10 variable F8 dynamic mutation results for direct rank based calculation at 75% chromosome crossover.**

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-180.15	0.09	-117.54	0.68	-116.98	-180.23	238.55	148.65
	1	-180.45	0.00	-117.53	0.76	-116.99	-180.49	469.00	34.85
	10	-180.48	0.00	-117.65	0.89	-116.96	-180.50	245.50	114.24
30%	0	-180.17	0.05	-117.48	1.13	-116.90	-180.28	295.30	124.73
	1	-180.37	0.11	-117.25	0.32	-116.97	-180.46	387.35	74.11
	10	-180.49	0.00	-117.37	0.56	-116.92	-180.50	313.90	92.90
50%	0	-180.16	0.06	-117.59	1.33	-116.98	-180.29	241.55	121.35
	1	-180.31	0.07	-117.64	1.53	-116.96	-180.40	388.45	88.30
	10	-180.49	0.01	-117.65	0.68	-116.99	-180.50	355.55	97.95

Table C.30: **10 variable F8 dynamic mutation results for inverse rank based calculation at 75% chromosome crossover.**

Max Mtn.	*El'tm*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. Best Gen.*	*St. Dev.*
10%	0	-180.07	0.10	-117.79	1.18	-116.97	-180.22	128.35	75.14
	1	-180.37	0.03	-117.75	1.50	-116.99	-180.46	168.35	27.59
	10	-180.41	0.04	-118.32	1.97	-116.97	-180.48	74.50	20.47
30%	0	-180.12	0.14	-117.63	1.20	-116.91	-180.26	187.15	120.28
	1	-180.38	0.10	-117.77	1.18	-116.91	-180.46	210.55	30.36
	10	-180.41	0.03	-117.98	1.30	-116.96	-180.47	73.30	10.85
50%	0	-180.17	0.12	-117.95	1.37	-116.96	-180.33	192.30	146.00
	1	-180.37	0.09	-117.43	0.60	-116.98	-180.47	225.85	34.84
	10	-180.41	0.07	-117.64	0.78	-116.90	-180.49	108.45	104.89

Table C.31: **10 variable F8 dynamic mutation results for direct fitness distance calculation at 75% chromosome crossover.**

Max Mtn.	*El'tn*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. best gen.*	*St. Dev.*
10%	0	-180.16	0.04	-117.41	0.89	-116.96	-180.36	241.45	108.18
	1	-180.35	0.02	-117.33	0.49	-116.99	-180.44	417.00	55.90
	10	-180.49	0.00	-117.77	1.12	-116.99	-180.50	358.90	105.89
30%	0	-180.08	0.00	-117.40	0.68	-116.95	-180.22	245.90	145.05
	1	-180.23	0.08	-117.53	0.70	-116.96	-180.34	320.65	133.26
	10	-180.47	0.00	-117.64	0.98	-116.95	-180.49	465.80	20.58
50%	0	-180.08	0.01	-117.31	0.64	-116.89	-180.21	236.70	138.07
	1	-180.12	0.00	-117.38	0.59	-116.95	-180.28	223.80	158.92
	10	-180.37	0.08	-117.94	1.17	-116.99	-180.45	394.55	104.97

Table C.32: **10 variable F8 dynamic mutation results for inverse fitness distance calculation at 75% chromosome crossover.**

C.3. Alternative alphabets

Surface	*Xover*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. best gen.*	*St. Dev.*
F7:2	Quot.	-1677.97	0.00	-142.46	72.05	-6.118	-1677.97	306.30	114.31
	Avg.	-1586.70	77.29	-131.35	74.26	-4.457	-1674.79	70.95	114.55
F7:10	Quot.	-4204.52	170.28	699.97	204.05	1212.29	-4541.30	443.50	86.41
	Avg.	-2982.22	388.15	794.31	176.20	1191.77	-3288.49	56.90	91.88
F8:2	Quot.	-180.50	0.03	-57.89	6.52	-54.61	-180.50	70.95	123.02
	Avg.	-180.50	0.06	-70.41	8.66	-56.04	-180.50	106.70	90.86
F8:10	Quot.	-180.50	0.00	-116.64	0.09	-116.50	-180.50	427.05	86.02
	Avg.	-180.49	0.00	-118.59	1.92	-117.05	-180.50	469.05	32.06

Table C.33: **Real valued results for experiments performed on F7 and F8 surfaces with mutation fixed at 4%.**

Surface	*Xover*	*Average Best*	*St. Dev.*	*Average Worst*	*St. Dev.*	*Worst Ever*	*Best Ever*	*Avg. best gen.*	*St. Dev.*
F7:2	Quot.	-1672.05	25.80	-149.95	56.14	-36.915	-1677.97	348.45	89.19
	Avg.	-1536.61	63.84	-146.95	73.47	-5.827	-1657.34	96.70	149.22
F7:10	Quot.	-4272.91	170.89	844.53	240.84	1536.80	-4595.01	410.75	86.29
	Avg.	-2446.69	228.81	701.05	261.69	1457.81	-3053.80	91.85	156.68
F8:2	Quot.	-180.50	0.00	-54.61	0.02	-54.61	-180.50	15.45	7.79
	Avg.	-180.50	0.08	-69.63	8.41	-57.38	-180.50	206.15	129.58
F8:10	Quot.	-180.50	0.07	-116.62	0.10	-116.49	-180.50	416.45	81.41
	Avg.	-180.49	0.00	-117.96	0.92	-116.99	-180.50	464.05	42.42

Table C.34: **Real valued results for experiments performed on F7 and F8 surfaces with dynamic mutation up to 10%.**