DECLARATION CONCERNING THESIS

Author’s full name and year of birth: Grant Cameron Dick
(for cataloguing purposes) 1 December, 1978

Title of thesis: Spatially-Structured Niching Methods for Evolutionary Algorithms

Degree: Doctor of Philosophy

Department: Department of Information Science

Permanent Address: 12 Drivers Road, Maori Hill, Dunedin, NEW ZEALAND

I agree that this thesis may be consulted for research and study purposes and that reasonable quotation may be made from it, provided that proper acknowledgement of its use is made.

I consent to this thesis being copied in part or in whole for:
   i) a library
   ii) an individual
at the discretion of the Librarian of the University of Otago.

Signature: .......................................................  
Date: ..........................................................

Note: This is the standard Library Declaration Form used by the University of Otago for all theses. The conditions set out on the form may be altered only in the most exceptional circumstances. Any restriction on access to a thesis may be permitted only with the approval of:

(i) the appropriate Pro Vice-Chancellor in the case of a Master’s thesis
(ii) the Deputy Vice-Chancellor (Research), in consultation with the appropriate Pro Vice-Chancellor, in the case of a PhD thesis

Where appropriate, the Director of the Enterprise Office may also be consulted.

The form is designed to protect the work of the candidate, by requiring proper acknowledgement of any quotations from it. At the same time the declaration preserves the University philosophy that the purpose of research is to seek the truth and to extend the frontiers of knowledge and that the results of such research which have been written up in thesis form should be made available to others for scrutiny.

The normal protection of copyright law applies to theses.

June 2008
Spatially-Structured Niching
Methods for Evolutionary Algorithms

Grant Dick

a thesis submitted for the degree of
Doctor of Philosophy
at the University of Otago, Dunedin,
New Zealand.

August 14, 2008
Abstract

Traditionally, an evolutionary algorithm (EA) operates on a single population with no restrictions on possible mating pairs. Interesting changes to the behaviour of EAs emerge when the structure of the population is altered so that mating between individuals is restricted. Variants of EAs that use such populations are grouped into the field of spatially-structured EAs (SSEAs).

Previous research into the behaviour of SSEAs has primarily focused on the impact space has on the selection pressure in the system. Selection pressure is usually characterised by takeover times and the ratio between the neighbourhood size and the overall dimension of space. While this research has given indications into where and when the use of an SSEA might be suitable, it does not provide a complete coverage of system behaviour in SSEAs. This thesis presents new research into areas of SEA behaviour that have been left either unexplored or briefly touched upon in current EA literature.

The behaviour of genetic drift in finite panmictic populations is well understood. This thesis attempts to characterise the behaviour of genetic drift in spatially-structured populations. First, an empirical investigation into genetic drift in two commonly encountered topologies, rings and torii, is performed. An observation is made that genetic drift in these two configurations of space is independent of the genetic structure of individuals and additive of the equivalent-sized panmictic population. In addition, localised areas of homogeneity present themselves within the structure purely as a result of drifting. A model based on the theory of random walks to absorbing boundaries is presented which accurately characterises the time to fixation through random genetic drift in ring topologies.

A large volume of research has gone into developing niching methods for solving multimodal problems. Previously, these techniques have used panmictic popu-
lations. This thesis introduces the concept of *localised niching*, where the typically global niching methods are applied to the overlapping demes of a spatially structured population. Two implementations, *local sharing* and *local clearing* are presented and are shown to be frequently faster and more robust to parameter settings, and applicable to more problems than their panmictic counterparts.

Current SSEAs typically use a single fitness function across the entire population. In the context of multimodal problems, this means each location in space attempts to discover all the optima. A preferable situation would be to use the inherent spatial properties of an SSEA to localise optimisation of peaks. This thesis adapts concepts from multiobjective optimisation with environmental gradients and applies them to multimodal problems. In addition to adapting to the fitness landscape, individuals evolve towards their preferred environmental conditions. This has the effect of separating individuals into regions that concentrate on different optima with the global fitness function. The thesis also gives insights into the expected number of individuals occupying each optima in the problem.

The SSEAs and related models developed in this thesis are of interest to both researchers and end-users of evolutionary computation. From the end-user’s perspective, the developed SSEAs require less *a priori* knowledge of a given problem domain in order to operate effectively, so they can be more readily applied to difficult, poorly-defined problems. Also, the theoretical findings of this thesis provides a more complete understanding of evolution within spatially-structured populations, which is of interest not only to evolutionary computation practitioners, but also to researchers in the fields of population genetics and ecology.
Acknowledgements

I never thought I’d actually be doing this…

Completing this thesis would not have been possible without the help of a num-
ber of great, helpful, loving and simply wonderful human beings. The brief
acknowledgement that they receive here is a grave injustice to the debt that I
owe them. However, to acknowledge each person to the level that they deserve
would require a separate volume to accompany this thesis, so this will have to
do!

First and foremost, thank you to my primary supervisor, Peter Whigham. Your
support and guidance over the last nine years that I have known you has been
unbelievable. You have endured more rantings, ravings and general hysterics
than I thought would have been humanly possible. For that, and everything else
you have done, I am eternally grateful. Thank you!

Thank you also to my other supervisor, Colin Aldridge. You insightful remarks,
positive attitude and constant encouragement throughout this journey have been
invaluable.

Thanks to everyone within the Department of Information Science, especially
to Steven Hall-Jones, Heather Harris, Gail Maxwell and the guys in TSG. Also,
thanks to Chris Edwards and Mark George for the interesting conversations and
other distractions while I was waiting for experiments to run their course. Spe-
cial thanks also to my office mate Gary Burrows for putting up with the some-
times erratic and frankly odd goings-on in Room 10.05.

To my fellow postgraduate students within the Information Science department,
I say thank you. In particular, Mustafa Dameh, who always knows how to
brighten a cloudy day.
To my family – Mum, Dad, Shelley and Melanie – thanks. Putting up with me for 29 years is not a task for the faint of heart, but somehow you have.

And of course, thanks to Dr. Melanie Middlemiss; thanks for your understanding, support . . . everything! The past seven years would have been impossible to bear without you . . .

Completing this thesis would not be possible without the generous support of a University of Otago Postgraduate Scholarship (2001) and a FRST Top Achiever Doctoral Scholarship (2002-2004).

Finally, to the two people that sadly, I cannot thank. My Nana, Mavis Reo Cottle, passed away on the 30th of May, 2001, which was the same day that I signed the application form to begin work on my PhD. Also, my cousin, Mark Cottle, who sadly passed away on the 11th of August, 2007. I am touched by the memory of these two people every day, and constantly moved by the impact that they had on the lives of those close to them. For these reasons, I dedicate this thesis to their memory.
# Contents

1 Introduction
   1.1 Motivation ........................................................................... 3
   1.2 Research Objectives and Methodology ................................. 4
   1.3 Contributions ...................................................................... 5
      1.3.1 Publications ................................................................. 6
   1.4 Thesis Structure .................................................................. 6

2 Related Work
   2.1 Multimodal Fitness Landscapes .............................................. 8
      2.1.1 Differences between multimodal and multiobjective optimisation ... 10
   2.2 Niching Evolutionary Algorithms for Multimodal Problems ....... 10
      2.2.1 Crowding methods ...................................................... 11
      2.2.2 Fitness sharing .......................................................... 14
      2.2.3 The implications of adopting fitness sharing ................... 15
      2.2.4 Clearing ................................................................. 19
      2.2.5 Other niching methods ............................................... 22
   2.3 Spatially-Structured Evolutionary Algorithms ....................... 22
      2.3.1 Multiple-deme evolutionary algorithms with migration ...... 23
      2.3.2 Isolation-by-distance population structures .................... 25
      2.3.3 Advantages of SSEAs ............................................... 29
      2.3.4 Alternative models of SSEAs ..................................... 30
      2.3.5 Multimodal problems and SSEAs ............................... 33
      2.3.6 Parapatric speciation with SSEAs ............................... 35
   2.4 Open Issues in Niching Using Spatially-Structured Populations ... 37

3 The Behaviour of Genetic Drift in Spatially-Structured Populations 39
   3.1 Properties of Genetic Drift in a Spatially-Structured Population ... 40
      3.1.1 Effect of population size and deme size on genetic drift .......... 41
      3.1.2 Effect of ploidy on genetic drift .................................. 43
   3.2 A Model for Genetic Drift in Ring Topologies ......................... 46
      3.2.1 Random walks ........................................................... 48
      3.2.2 Genetic drift in structured populations as a random walk .... 50
      3.2.3 Genetic drift in spatially-structured populations with multi-locus indi-
           viduals ................................................................. 51
      3.2.4 Limitations of proposed model .................................... 55
   3.3 Other Properties of Spatially-Structured Genetic Drift ............ 55
3.3.1 Genetic drift in two-dimensional structures .......................... 56
3.3.2 Genetic drift with more than two alleles .......................... 58
3.4 Implications of SSEAs as Multimodal Optimisers .................... 58

4 Environmental Gradients, SSEAs and Parapatric Speciation 63
4.1 A Localised Environment for SSEAs .................................. 64
4.2 Elementary Behaviour .................................................... 68
4.3 Analytical Properties ...................................................... 72
  4.3.1 Equivalence points .................................................. 75
  4.3.2 Parameter sensitivity ................................................ 80
4.4 Extended Behaviour ....................................................... 82
  4.4.1 Manufactured phenotypes, deception and dead zones .......... 87
  4.4.2 Alternative gradient approaches ................................ 90
4.5 Summary ................................................................. 95

5 Local Niching I: Local Sharing ........................................... 97
5.1 Local Sharing ............................................................. 97
  5.1.1 Basic local sharing behaviour .................................... 100
  5.1.2 Elitism in local sharing ............................................ 101
  5.1.3 Deme size in local sharing ........................................ 104
5.2 A Weighted Local Sharing Approach .................................. 107
5.3 Summary ................................................................. 111

6 Local Niching II: Local Clearing and Parameter Robustness 113
6.1 The Local Clearing Method ............................................. 114
  6.1.1 Base performance .................................................. 114
6.2 Parameter Robustness .................................................... 117
6.3 Automatic Identification of $\sigma_{cl}$ ................................ 124
6.4 Summary ................................................................. 130

7 Conclusion ................................................................. 131
7.1 Resolved Issues in Niching Using Spatially-Structured Populations 133
7.2 Recommendations for Real-World Problem Solving ................ 135
  7.2.1 General parameter settings ....................................... 136
7.3 Future Work .............................................................. 137
  7.3.1 Population topologies ............................................. 138
  7.3.2 Gradient-based SSEAs ............................................. 138
  7.3.3 Local niching ....................................................... 138
  7.3.4 General research .................................................. 139

References ................................................................. 140

A Basic Elements of Evolution ............................................. 150
A.1 Mechanisms of Evolution ............................................. 151
A.2 Speciation ............................................................... 151
B Models of Selection and Drift
   B.1 Selection ................................................................. 153
   B.2 Random Genetic Drift in Panmictic Populations .......... 153

C Experimental Framework ............................................. 160
   C.1 Parameter settings ...................................................... 160

D Performance Measures .................................................. 163
   D.1 Species Assignment ..................................................... 163
   D.2 Niche Discovery Performance – Maximum Peak Ratio .... 164
   D.3 Expected Niche Proportions – Chi-Square-Like Performance . 164
   D.4 Species Visualisation over Time ................................. 166

E Test Functions ................................................................. 167
   E.1 Sine Functions: M1 and M4 ........................................ 168
   E.2 Shekel’s Foxholes ...................................................... 169
   E.3 The Shubert Function ............................................... 170
   E.4 The Massively Multimodal, Deceptive Problem (MMDP) .... 171
   E.5 The Minimum Hamming Distance Problem (MINHD) ......... 172
List of Tables

3.1 $R^2$ correlation values for the random walk model of genetic drift on a ring of size $N$ and $d = 1$. ................................................................. 55

4.1 The local fitness values of the optima in the $M1$ and $M4$ problems against each possible ideal phenotype when using a $20 \times 20$ torus. The fittest phenotype for each location has been highlighted. Note that there are no positions in the gradient that favour the fifth optimum of the $M4$ problem. .......... 72

4.2 The equivalence points between peaks for the $M4$ function. ................. 76

6.1 The range of values used for $\sigma_{cl}$ for each of the four test problems ....... 117

6.2 A sample of the species maintained at various locations in the population for the $MMDP$ problem using a value for $\sigma_{cl}$ of 6. Note that although no single deme maintains all 32 optima, collectively every optima in the fitness landscape is being maintained. ................................. 123

6.3 A sample of the species maintained at various locations in the population for the $MMDP$ problem using a value for $\sigma_{cl}$ of 15. Note that although no single deme maintains all 32 optima, collectively every optima in the fitness landscape is being maintained. ................................. 124

D.1 The expected distribution of individuals over the fitness landscape of $M1$. .... 165

D.2 The expected distribution of individuals over the fitness landscape of $M4$. .... 165
## List of Figures

2.1 Examples of unimodal and multimodal fitness landscapes in comparison to a problem with multiple objectives. ........................................ 9
2.2 Process of an EA using De Jong-style crowding. ......................... 12
2.3 The effect of sharing on a fitness landscape. The solid line depicts the actual fitness landscape. The individuals of the population are represented as points on the fitness landscape. The dashed line depicts the fitness landscape after sharing has been applied to the population. ......................... 15
2.4 Example population topologies commonly used in SSEAs. The circles represent the individuals of the population, while the lines represent the connections between individuals and are used to establish demes. ............................... 28
2.5 Deme construction for a ring topology. The grey element represents the central individual of the deme. The black elements represent individuals that will take part in the deme. .............................................. 28
2.6 Example population topologies commonly used in SSEAs. ........... 28
3.1 Simulation of genetic drift on a ring with respect to increasing population size. ......... 41
3.2 Simulation of genetic drift on a ring with respect to increasing deme size (population size $N = 100$). The population behaviour approaches panmixia as $d$ increases. The panmictic fixation time was calculated via a diffusion model approximation of genetic drift (Equation (B.9)). ......................... 42
3.3 Comparison of decay of heterozygosity in a ring structure and a panmictic population. Small values of $d$ result in a rapid elimination of heterozygotes from the ring. .............................................. 44
3.4 Evolution of a ring-structured population over time for haploids and diploids. Green points are individuals that are heterozygous, while yellow and blue represent the two possible homozygous genotypes. The majority of heterozygotes in diploid population are quickly eliminated and the remnants exist solely between the boundaries of homozygotes. Despite the differences in genotypes, the overall dynamics of the population are very similar. .... 45
3.5 The effect of genetic drift on different levels of ploidy. The overall effect of genetic drift on a ring appears to be additive and independent of the underlying ploidy of the individuals. .............................................. 46
3.6 Comparison between simulation and extended Wright-Fisher model for genetic drift on a ring. .............................................. 48
3.7 A particle in one-dimensional space with absorbing boundaries. ........ 49
3.8 Comparison between simulation of genetic drift and the proposed random walk model. .............................................. 52
3.9 The change in fixation time due to increased number of loci in the genotype. The increase in time is approximately linear with respect to the logarithm of the number of loci. ................................. 53
3.10 Comparison between simulation and model for mean generations to fixation via genetic drift on a ring with multiple loci. ................................. 54
3.11 The effect of genetic drift within a torus on different levels of ploidy. The behaviour with respect to ploidy has a similar two component property as that of a ring structure. ................................. 56
3.12 Evolution of a torus-structured population over time for haploids and diploids. Homozygotes are presented in blue or yellow, with heterozygotes highlighted in green. As with the ring structure, the majority of heterozygotes in diploid population are quickly eliminated and the remnants exist largely between the boundaries of homozygotes. Large homogeneous subpopulations also quickly appear within the population. ................................. 57
3.13 Comparison of genetic drift on a ring with multiple alleles and with two alleles for a given initial allele frequency. ................................. 59
3.14 The discovery and maintenance of optima via a panmictic and spatially-structured EA under selection and drift. Termination of an EA was via a halting window technique (Beasley, Bull and Martin, 1993) ................................. 60
3.15 Behaviour of SSEA on two test problems with the reintroduction of selection. ................................. 61

4.1 The generation of an environmental gradient for the GBSSEA. Note that all possible combinations of phenotypic trait (within the granularity of space) are produced. ................................. 65
4.2 The effect of the $\sigma$ parameter for the function that determines an individual’s response to a given environment. Small values for $\sigma$ encourage strong adaptation to the local environment. ................................. 66
4.3 The effect that response to a local environment has on the overall fitness landscape using the $M4$ problem. Note that, when a local environment response is introduced, the fitness landscape is modified so that each optima maintains a region in the environment in which it is locally the fittest solution. ................................. 68
4.4 A simple definition of a gradient for $M1$ and $M4$ problems using a $20 \times 20$ torus. ................................. 69
4.5 The maximum peak ratio performance on the $M1$ and $M4$ problems for a simple SSEA and an SSEA that incorporates an environmental gradient. ................................. 70
4.6 The chi-square-like performance on the $M1$ and $M4$ problems for a simple SSEA and an SSEA that incorporates an environmental gradient. ................................. 71
4.7 Emergence of species over time for the $M1$ problem using a simple SSEA. ................................. 73
4.8 Emergence of species over time for the $M4$ problem using a simple SSEA. ................................. 73
4.9 Emergence of species over time for the $M1$ problem using a GBSSEA. ................................. 74
4.10 Emergence of species over time for the $M4$ problem using a GBSSEA. ................................. 74
4.11 The phenotype space that each optimum occupies as the locally fittest solution. Each block of colour represents the area that a given optimum is dominant. ................................. 75
4.12 The maximum peak ratio and chi-square-like performance of the gradient-based SSEA approach using a $52 \times 52$ torus. ................................. 77
4.13 Emergence of species over time for the $M4$ problem using a Gradient-Based SSEA and a $52 \times 52$ torus. The gradient includes a single ideal phenotype that promotes the fifth optimum as the locally fittest solution.

4.14 A ‘piecewise’ tunable three-peak function. The parameters $\alpha_1$, $\alpha_2$ and $\alpha_3$ control the relative heights of the peaks.

4.15 Emergence of niches over time with the three-peak problem ($\alpha_2 = 0.300$).

4.16 Emergence of niches over time with the three-peak problem ($\alpha_2 = 0.400$).

4.17 The effect of $\sigma$ on the equivalence points between species in the $M1$ and $M4$ fitness landscapes. The equivalence points in $M1$ are particularly robust with regards to different values of $\sigma$. However, the equivalence points in $M4$ are heavily dependent on the value of $\sigma$.

4.18 The effect of $\sigma$ on the maximum peak ratio and chi-square-like performance of the gradient-based SSEA approach.

4.19 The maximum peak ratio and chi-square-like performance of the gradient-based SSEA approach on the Shekel’s Foxholes problem.

4.20 Emergence of species over time for the Shekel’s Foxholes problem using a Gradient-Based SSEA.

4.21 The maximum peak ratio performance of the GBSSEA on the $M1$ problem with increasing dimensionality of the problem space.

4.22 The maximum peak ratio and chi-square-like performance of the gradient-based SSEA approach on the MMDP problem.

4.23 Emergence of species over time for the MMDP problem using a Gradient-Based SSEA.

4.24 Emergence of regions in the population that are occupied by deceptive solutions in the MMDP problem. In the most extreme case, every cell in the $10 \times 10$ torus is occupied by a deceptive solution.

4.25 An alternative ‘rugged’ approach to gradient implementation. The ideal phenotype for each location is generated randomly.


4.27 Distribution of species using a rugged gradient.

4.28 The effect of gradient implementation on maximum peak ratio performance of the GBSSEA on the $M1^m$. At higher problem dimensions, the rugged gradient implementation is able to maintain more optima within the population than both the simple gradient and SSEA (no gradient) approaches.

4.29 The difference in performance between the naive rugged gradient implementation and one that ensures a 1:1 mapping between genotype and phenotype.

5.1 The performance of the local sharing method on the Foxholes problem. The performance of a standard SSEA and a fitness sharing EA are included as a point of reference.

5.2 The performance of the local sharing method on the MMDP problem. The performance of a standard SSEA and a fitness sharing EA are included as a point of reference.

5.3 The performance of the local sharing method on the Foxholes problem. Elitist replacement was removed from both spatially-structured EAs.
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>The performance of the local sharing method on the MMDP problem. Elitist</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>replacement was removed from both spatially-structured EAs.</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>The performance of the local sharing method on the Foxholes problem. Elitist</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>replacement by a probabilistic tournament was used for both spatially-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>structured EAs.</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>The performance of the local sharing method on the MMDP problem. Elitist</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>replacement by a probabilistic tournament was used for both spatially-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>structured EAs.</td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>The performance of the local sharing method on the Foxholes problem. A</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>larger ‘D25’ deme size was used. Performance of a standard SSEA and a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fitness sharing EA are included as a point of reference.</td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>The performance of the local sharing method on the MMDP problem. A</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>larger ‘D25’ deme size was used. Performance of a standard SSEA and a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fitness sharing EA are included as a point of reference.</td>
<td></td>
</tr>
<tr>
<td>5.9</td>
<td>The effect of increasing deme size on maximum peak ratio performance of the</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>Foxholes problem.</td>
<td></td>
</tr>
<tr>
<td>5.10</td>
<td>The effect of increasing deme size on maximum peak ratio performance of the</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>MMDP problem.</td>
<td></td>
</tr>
<tr>
<td>5.11</td>
<td>A potential function for determining the weights applied to the sharing</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>function in the Local Sharing SSEA.</td>
<td></td>
</tr>
<tr>
<td>5.12</td>
<td>The performance of the weighted local sharing method on the Foxholes problem.</td>
<td>110</td>
</tr>
<tr>
<td>5.13</td>
<td>The performance of the weighted local sharing method on the MMDP problem.</td>
<td>111</td>
</tr>
<tr>
<td>6.1</td>
<td>The maximum peak ratio performance of the local clearing approach compared</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>to an EA using simple clearing.</td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>The effect of $\sigma_{cl}$ of the maximum peak ratio performance for</td>
<td>118</td>
</tr>
<tr>
<td></td>
<td>standard and local clearing on the Foxholes problem.</td>
<td></td>
</tr>
<tr>
<td>6.3</td>
<td>The effect of $\sigma_{cl}$ of the maximum peak ratio performance for</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>standard and local clearing on the Shubert problem.</td>
<td></td>
</tr>
<tr>
<td>6.4</td>
<td>The effect of $\sigma_{cl}$ of the maximum peak ratio performance for</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>standard and local clearing on the MMDP problem.</td>
<td></td>
</tr>
<tr>
<td>6.5</td>
<td>The effect of $\sigma_{cl}$ of the maximum peak ratio performance for</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>standard and local clearing on the MINHD problem.</td>
<td></td>
</tr>
<tr>
<td>6.6</td>
<td>The effect of deme size on the ability of the local clearing method to</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>maintain solutions in the presence of an incorrect value for $\sigma_{cl}$.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Shubert problem was being searched with a value for $\sigma_{cl}$ of 1.00.</td>
<td></td>
</tr>
<tr>
<td>6.7</td>
<td>The maximum peak ratio performance of the local clearing approach compared</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>to an EA using simple clearing.</td>
<td></td>
</tr>
<tr>
<td>6.8</td>
<td>The evolution of $\sigma_{cl}$ over time for the four test problems.</td>
<td>129</td>
</tr>
<tr>
<td>B.1</td>
<td>Three examples of change in allele frequency via genetic drift. Unlike</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>selection, random genetic drift evolves the population in no specific</td>
<td></td>
</tr>
<tr>
<td></td>
<td>direction.</td>
<td></td>
</tr>
<tr>
<td>B.2</td>
<td>The Wright-Fisher model of genetic drift as compared to experimental data</td>
<td>158</td>
</tr>
<tr>
<td>D.1</td>
<td>Possible scenarios for species discovery and maintenance.</td>
<td>166</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>E.1</td>
<td>The two sine functions.</td>
<td>168</td>
</tr>
<tr>
<td>E.2</td>
<td>The Shekel’s <em>Foxholes</em> function.</td>
<td>169</td>
</tr>
<tr>
<td>E.3</td>
<td>The <em>Shubert</em> function.</td>
<td>170</td>
</tr>
<tr>
<td>E.4</td>
<td>The massively multimodal deceptive problem is the result of concatenating</td>
<td>171</td>
</tr>
<tr>
<td></td>
<td>five bimodal, deceptive problems</td>
<td></td>
</tr>
<tr>
<td>E.5</td>
<td>The minimum Hamming distance function, <em>MINHD</em></td>
<td>172</td>
</tr>
</tbody>
</table>
List of Algorithms

2.1 The deterministic crowding algorithm ........................................... 13
2.2 An evolutionary algorithm that incorporates fitness sharing. .......... 17
2.3 The clearing algorithm ............................................................... 20
2.4 Breeding a generation using the clearing method. ......................... 21
2.5 The general sequence for a spatially-structured evolutionary algorithm. . 27
4.1 The general sequence for a spatially-structured evolutionary algorithm that incorporates an environmental gradient. .................. 67
5.1 The local sharing method. .............................................................. 98
5.2 The general sequence for a spatially-structured evolutionary algorithm with local sharing. ......................................................... 99
6.1 The general sequence for a spatially-structured evolutionary algorithm with local clearing. ...................................................... 115
6.2 The general sequence for adapting $\sigma_{cl}$ for clearing using information garnered from the current population. ......................... 127
Chapter 1

Introduction

Evolutionary algorithms (EAs) are a family of general purpose, robust search mechanisms inspired by the processes of biological evolution. Through an iterative process of ‘generate and test’, EAs select points (individuals) that occupy promising regions of a search space and generate new individuals from those selected through recombination and/or variation operators. EAs are an interesting search heuristic as they are applicable to cases in which a priori knowledge of the problem is sparse. Being population based, EAs are global optimisers; communication (through recombination) between individuals allows the population to rapidly identify regions of the fitness landscape with high fitness and hence locate a satisfactory solution to a given problem.

Problems with difficult fitness landscapes typically require the use of large populations in order for an EA to perform an effective search. Naturally, the use of large populations requires a large amount of computation for the EA. The need for large population sizes has led researchers in evolutionary computation to develop mechanisms that make EAs more suitable to execution via parallel computation methods. One such way is to partition the population into smaller units and process each unit separately on multiple processors. In order to allow each processor to act independently (and hence in parallel), the communication between each processor must be kept to a minimum. Therefore, restrictions are placed on the parts of the population that can interact. Individuals are mapped to a topological space and the implied distances between population members are used to restrict mating to individuals that are close to each other. These spatial interactions lead to the formation of subpopulations that operate in relative isolation. In addition to providing a logical organisation with which to implement parallel processing, the population structures within these spatially-structured EAs (SSEAs) demonstrate markedly different behaviour to that of their unstructured counterparts. It did not take long for researchers to realise that such population structures mimicked
those described in population genetics literature. As such, the focus of research into SSEAs moved away from computational efficiency and toward the behavioural aspects of restricted mating through the introduction of population structure.

Evolutionary algorithms are often used to search problem spaces in order to find a single global optimum. However, many fitness landscapes encountered in real-world problems are multimodal and as such may possess numerous desirable optima. In such cases, it would be desirable for an EA to return a set of solutions to a user. For example, in a job scheduling problem there might be many possible configurations of jobs that lead to the shortest completion time; ideally, an optimisation method would return multiple “optimal” schedules to the user and, with external information that could not be factored into the search method, let them make the final choice. A traditional EA, designed to converge upon a single solution, will typically fail at this task. One can attempt to discover multiple optima of a given problem by running a traditional EA many times in the hope that each run converges to a different region in the fitness landscape. This concept can be extended slightly by using knowledge gained from earlier runs to alter the fitness landscape and increase the likelihood that the future runs converge upon novel optima. However, this iterative approach is highly dependent on the stochastic nature of EAs and typically requires excessively large numbers of runs in comparison to the number of optima present in the fitness landscape.

A more sophisticated alternative to relying on multiple runs of an EA and chance convergence is to use the analogy between the process of discovering multiple solutions to a given problem and the processes the populations undergo during the formation of new species. From this analogy, concepts such as non-random mating through population structure, assortative mating and frequency-dependent selection have all been adopted from population genetics for use in evolutionary computation. These resultant niching EAs differ from the iterative approach in that they attempt to discover multiple optima within a given fitness landscape in the course of a single run. The discovery and maintenance of multiple optima in a single population offer niching EAs many benefits over their traditional counterparts, including:

- **Search efficiency**: a niching EA requires fewer iterations to discover multiple optima as it proactively seeks out novel areas of the fitness landscape.

- **Diversity maintenance**: through the course of a single run, a niching EA will try to form subpopulations around optima in the fitness landscape. A niching EA can then use the optima it has already discovered as building blocks for discovering further optima.
• Post-processing of optima by ‘experts’: a niching EA may not factor all the parameters of a given problem into the evolutionary search. This could be for computational reasons, or to simplify the fitness landscape. In such circumstances, a niching EA can identify several good candidate solutions for the end-user, who can then pick the best solution through further analysis.

1.1 Motivation

Spatially-structured evolutionary algorithms have been subject to considerable attention recently. However, most theoretical research into SSEAs has concentrated on the concept of takeover times, that is the time required for an individual to become the sole solution in the population. Research into takeover times is of little use in the context of multimodal optimisation, as the convergence of the population onto a single location in the fitness landscape is undesirable. What research there is into the use of SSEAs as multimodal optimisers is inconclusive. While some claim that SSEAs are capable of cover optimisation, others suggest that the presence of multiple species in the population is temporary and that an SSEA will ultimately converge upon a single entity. At best, this research suggests that the abilities of SSEAs to locate and maintain multiple solutions is heavily problem dependent. Models from population genetics suggest that SSEAs should be effective multimodal optimisers, so what exactly is causing them to be unreliable?

SSEAs bear some resemblance to speciation models found in population genetics. More precisely, SSEAs have some properties analogous to the parapatric model of speciation. One of the central concepts underlying parapatric speciation is that different regions in space will present different challenges to its inhabitants. Selection pressures are therefore not constant throughout space, and the resultant localised selection promotes genetic divergence. However, a typical SSEA applies the same fitness function to each individual in the population. Therefore, each location in the population presents the same selection pressures. Clearly, there is a discrepancy between the nature of parapatric speciation and the implementation of a typical SSEA.

Although far more mature with respect to multimodal optimisation, niching EAs are not without faults. Many niching EAs require the use of a similarity measure. Although the effort required to measure the distance between individuals is often less than the effort required to determine an individual’s fitness, many distance measures must be taken in the course of a single generation. In the worst cases, each individual must be compared with every other population member. Given that large population sizes are typically needed for difficult problems, the time complexity of niching algorithms cannot always be ignored.
Another, perhaps more serious, problem with many niching EAs is the requirement of a problem-specific parameter, the *niching radius*. This is the parameter used to determine if two individuals belong to the same species and must be set to reasonable values if the niching EA is to work properly; too large a value will force the niching EA to merge close species, while too small a value will stagnate the evolutionary search. Unfortunately, this parameter is difficult to set without *a priori* knowledge of the fitness landscape of the problem being searched.

The goal of this thesis is to identify the attributes of spatially-structured EAs that are advantageous, or alternatively disadvantageous, in the context of discovering multiple desirable solutions in a complex fitness landscape. Once these properties are suitably identified, the thesis will develop simple extensions to existing SSEAs that will make them more reliable in the context of multimodal optimisation. The final aim of this thesis is to take the resultant SSEAs and compare their efficiency and robustness with existing niching EA methods.

### 1.2 Research Objectives and Methodology

The objectives of this thesis are two-fold. First, the thesis will identify the shortcomings of SSEAs with respect to multimodal optimisation and introduce simple extensions that significantly alter the behaviour of SSEAs on multimodal problems. Second, we will examine how the incorporation of spatially-structured populations into existing niching techniques can bestow these algorithms with new properties that extend their applicability to new problems and improve their robustness to existing problem domains.

The first part of this study identifies the behavioural components of SSEAs that are useful in the context of multimodal optimisation, namely genetic drift and the resultant local divergence of genotypes. Through a combination of formal and empirical methods, we show that drift in SSEAs is both quantitatively and qualitatively different to a unstructured EA. First, genetic drift in a one-dimensional SSEA will be characterised through extensive simulations. Then a formal analysis based upon random walk theory will be performed to accurately model drift time. Finally, the applicability of the model to other SSEA configurations will be examined through simulation.

Subsequently in this thesis parapatric speciation is introduced into SSEAs through the use of an environmental gradient. The introduction of an environmental gradient ties the selection process to the population structure. Selection is no longer constant throughout space as an individual’s *local fitness* depends not only on its genetic make-up, but also on the conditions presented to it by the environment at its current location. It therefore becomes
less likely that an individual will be maximally fit at all locations in space. Therefore, our goal is to understand and characterise the behaviour of gradient-based SSEAs under different environmental configurations on several benchmark multimodal optimisation problems. We then formally analyse these findings to develop theoretical models to describe the types of problems that are amenable to searching via a gradient-based SSEA.

Incorporating a gradient into the population structure is one potential solution to creating an SSEA capable of multimodal optimisation. Hybridising sympatric methods, such as sharing, into SSEAs has been proposed, yet no actual implementations exist in the evolutionary computation literature. Two methods, local sharing and local clearing, are developed. Our goal here is to characterise the behaviour of these two algorithms with respect to parameter setting, computational complexity and problem applicability through an extensive empirical analysis.

1.3 Contributions

- We have developed a quantitative model that describes the mean time to fixation of alleles in a one-dimensional SSEA under genetic drift. We have also shown that the model should be extendable to higher dimension population structures. The developed model has benefits not only to the evolutionary computation community; since the model uses a problem-independent representation consisting of alleles, loci and ploidy, it should also be of interest to researchers in population genetics and complex systems.

- We have taken a model of speciation along environmental gradients and adapted it for use in multimodal optimisation via SSEAs. We have also identified the types of problems that will be searchable through this method.

- We have developed two SSEAs that incorporate sympatric niching methods into the local mating process. We have shown that these methods incorporate a level of redundancy in the speciation process that gives them a high level of robustness with respect to parameter setting. This should give them an advantage over existing niching methods on problems where a priori knowledge of the fitness landscape is sparse.
1.3.1 Publications

Several sections of this thesis have been published in peer-reviewed conference proceedings or journals:

**URL:** http://dx.doi.org/10.1109/CEC.2005.1554913

**URL:** http://dx.doi.org/10.1007/11903697_58

**URL:** http://dx.doi.org/10.1007/11903697_64

**URL:** http://dx.doi.org/10.1007/s11390-008-9110-6

1.4 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 gives an overview of niching evolutionary algorithms and other research relevant to this thesis. Chapter 3 presents a detailed analysis of genetic drift within the confines of spatially-structured populations typically used in SSEAs. Chapter 4 introduces the use of environmental gradients and SSEAs
to effectively perform multimodal optimisation and describes the concept of local fitness; the fitness of an individual with respect to its location in space. Chapter 5 presents the local niching concept and analyses the behaviour of one particular implementation of local niching – the local sharing concept. Chapter 6 uses another implementation of local niching – the local clearing method – to investigate the robustness of local niching with respect to parameterisation. Finally, Chapter 7 summarises the findings of this thesis and outlines the paths that future work in this field should take.

In addition to the main body of the thesis, three appendices have been included to help the reader understand some of the peripheral concepts touched upon in this work: Appendix A introduces some of the basic population genetics terminology; Appendix B provides a brief discussion of the theoretical aspects of selection and genetic drift found in population genetics and basic models of evolution; Appendix C discusses the experimental framework for the thesis and the common parameters used across all experiments; Appendix D describes the measures used for comparative analysis in this thesis; and Appendix E documents the test functions used to analyse the evolutionary algorithms produced in the course of this work.
Chapter 2

Related Work

Evolution in artificial systems follows the same basic principles as those of natural populations. Each individual possesses a coded solution to a given problem; the *genotype*. The genotype is decoded into a *phenotype*, which is a description of an individual’s response to a given problem. The suitability of the phenotype of a given individual to the problem determines its fitness. Fitter individuals tend to survive and reproduce more often than less fit individuals and, when individuals reproduce, they pass on some of their traits to their offspring. Occasionally, new genetic information will be introduced into a population through *mutation*. Evolutionary algorithms (EAs) are therefore a form of stochastic search that utilise selection and inheritance to discover near-optimal solutions to arbitrary problems.

This thesis investigates the use of spatially-structured evolutionary algorithms for searching multimodal fitness landscapes. In order to determine what is required from such methods, a review of the current state of niching methods for EAs is made. This chapter critically reviews related material and highlights some areas that previous work has not addressed. Some prior knowledge of evolution and evolutionary algorithms on the part of the reader is assumed. Basic explanations and common terminology of the fields are presented as appendices at the end of this thesis.

2.1 Multimodal Fitness Landscapes

Evolutionary algorithms are applicable to a wide range of problems. The fitness landscapes that these problems present are also varied; some may present a landscape similar to that of Figure 2.1(a) in which there is one global optimum and the path to that optimum is relatively smooth. These are generally referred to as *unimodal* fitness landscapes. A more challenging class of fitness landscape, such as the one depicted in Figure 2.1(b), will have many local...
optima. The optima in these multimodal landscapes may differ in value and their distribution within the landscape may be uneven. Simple EAs have previously been shown to be successful in finding a single global optimum in multimodal fitness landscapes (De Jong, 1975). However, in a multimodal fitness landscape, there may no longer be just one global optimum, but rather several global optima distributed throughout the search space. Alternatively, there might be a single global optimum, but throughout the fitness landscape there are several local optima similar in fitness to the global optimum. Collectively, the set of global and near-global optima are referred to as the desired optima of a fitness landscape. It is the goal of multimodal optimisation, then, to identify all the desired optima in a given fitness landscape (Mahfoud, 1995b).

Typically, a single solution is returned to the user when using a simple EA to search a fitness landscape. This occurs regardless of the number of optima in the landscape and their relative size. A complication arises when a simple EA is used to attempt to locate all possible optima in a fitness landscape simultaneously. If the desired optima do not have the same fitness, selection pressure within the EA will drive the population toward the optimum with the greatest fitness. A problem still exists even if all the optima in the fitness landscape are of equal fitness; in this case, genetic drift will remove individuals from other peaks until the population resides on a single point in the fitness landscape. Modifications are required to the framework of EAs in order to prevent either of these eventualities.

Multimodal optimisation is an important area of research; several real-world problem domains exhibit highly multimodal fitness landscapes. In job shop scheduling problems, for example, it is likely that the objective of reducing the time required to complete tasks can be solved though several equivalent plans (Perez, Herrera and Hernandez, 2003). In a more general sense, all combinatorial problems have the potential to possess multimodal fitness
landsces (Yang, 1998). Additionally, multimodal optimisation can play an important role in training neural network ensembles Yao and Liu (2004).

2.1.1 Differences between multimodal and multiobjective optimisation

There has been considerable work investigating the use of EAs for multiobjective optimisation and many successful methods have been developed. These multiobjective evolutionary algorithms (MOEAs) are able to support many different solutions within a single population (Deb, 2001). There are obvious connections between multimodal and multiobjective optimisation and some methods initially designed for multimodal problem solving have been successful in multiobjective optimisation (Horn, Nafpliotis and Goldberg, 1994). Despite these similarities, a distinction must be made between MOEAs and EAs for multimodal problem solving (Mahfoud, 1995b). In a multiobjective optimisation, one searches for solutions that present a compromise between two or more potentially conflicting objectives (see Figure 2.1(c)). Ideally, the solutions discovered in a multiobjective optimisation will strictly dominate all other points in the search space; they will be strictly better on at least one objective in the problem, and strictly no worse on the remaining criteria. The set of points in the search space that are not strictly dominated by any other points are said to lie on the Pareto-optimal front (Deb, 2001). Although solutions that lie on the Pareto-optimal front (the blue dots in Figure 2.1(c)) may differ in fitness between objectives, they are essentially equal in fitness when all objectives are considered simultaneously.

Like multiobjective optimisation, the goal in a multimodal search is to find many unique and desirable solutions within a given fitness landscape. However, unlike multiobjective optimisation, the solutions discovered in a multimodal search method will present optimal (or near-optimal) configurations to satisfy a single objective.

2.2 Niching Evolutionary Algorithms for Multimodal Problems

The problems presented to EAs by multimodal problems have long been recognised in the literature, and potential solutions aimed at maintaining population diversity appeared early on in the history of evolutionary computation (De Jong, 1975). Methods such as crowding use generation gaps and restricted replacement based around similarity between individuals. Later techniques use the concept of finite resources and of rewarding individuals for searching novel locations in the fitness landscape (Goldberg and Richardson, 1987). These
techniques, collectively referred to as *niching methods*, all used the concept of similarity between individuals as the basis for maintaining multiple optima in the population (Mahfoud, 1995b).

### 2.2.1 Crowding methods

De Jong (1975) initially proposed the crowding method as a way to maintain diversity within a population in order to prevent premature convergence. The crowding method used overlapping generations; instead of breeding an entire population in every generation, a proportion of offspring were created as per a normal EA with a generation gap. Instead of performing uniform sampling to determine the individual to replace, De Jong used the following method: for each offspring, select a sample from the population and determine the ‘distance’ between the current offspring and each individual in the sample. The closest matching individual is removed from the population and the offspring takes its place. The general process of crowding is shown graphically in Figure 2.2.

Crowding introduces a new parameter, the crowding factor $CF$, which determines the number of individuals drawn from the population to determine the replaced individual. When $CF = 1$, no crowding is performed and the algorithm behaves like a standard EA with a generation gap. When $CF$ equals the population size, the entire population is scanned to determine the individual that will be replaced.

Crowding relies on a distance measure between individuals in order to determine the replaced individuals. In De Jong’s study, individuals were encoded as binary strings and the Hamming distance between individuals was used as the measure of similarity between population members (De Jong, 1975). Later studies on crowding suggested that instead of using Hamming distance (i.e. genotypic) measures, similarity between individuals should be measured using phenotypic distances (Mahfoud, 1992, 1995b).

The test suite used by De Jong contained one multimodal problem. Although De Jong was only interested in finding one optimum, he noted that performance on this problem was improved with the introduction of crowding. De Jong conjectured that this was a result of the increased diversity that crowding promoted.

The crowding method was further analysed by Mahfoud who directly examined its ability to discover and maintain multiple peaks in a fitness landscape (Mahfoud, 1992). Mahfoud was able to show that crowding, as implemented by De Jong, can only guarantee the maintenance of two optima in any problem. This led Mahfoud to perform extensive experiments on two test problems in an attempt to develop an improved crowding method. The first problem he observed was that basic crowding made many replacement errors, meaning that
Figure 2.2: Process of an EA using De Jong-style crowding.
it frequently replaced individuals from one niche with offspring from another. To reduce the likelihood of replacement errors, Mahfoud set the crowding factor to equal the population size. At this setting, Mahfoud made an interesting observation; 99.9% of the time an offspring replaced one of their parents. This led to the removal of the crowding factor altogether and the establishment of a crowding method in which an individual replaced the closer of its two parents. This introduced a new problem; since all parents were involved in mating, the method did not apply any selection pressure on the population. In order to create a selection pressure in the system, Mahfoud incorporated Cavicchio’s preselection algorithm (Cavicchio, 1970); selection pressure toward fitter individuals was created by insisting that offspring replaced their closest parent so long as their fitness exceeded that of the parent. The final algorithm, which Mahfoud named deterministic crowding is shown in Algorithm 2.1.

Deterministic crowding was shown to be able to successfully search difficult multimodal problem domains, although the algorithm typically requires more function evaluations than other niching methods (Mahfoud, 1995a; Pétrowski, 1996). However, deterministic crowding has problems maintaining multiple solutions in multi-dimensional real-parameter fitness landscapes (Mahfoud, 1995b; Watson, 1999).

2.2.2 Fitness sharing

An environment in a natural evolutionary system will typically provide a limited number of resources, which restricts the number of organisms that the environment can sustain. In ecological terms, this is known as the carrying capacity of the environment. Simple EAs do not model this; instead they assume an infinite number of resources and allocate them equally to each individual, hence the individual receives the full reward from the fitness function. This has several implications; first, in a multimodal environment with unequal peaks, all individuals will seek out the tallest peak at the expense of the lesser optima. Second, the infinite resource model does not consider competition for resources, meaning that all individuals can comfortably coexist on the same peak and receive the same fitness that they would have if they were the only individual on that peak.

The concept of sharing finite resources in an evolutionary algorithm was first proposed by Holland (1975), but fitness sharing (Goldberg and Richardson, 1987) was the first successful attempt at modelling resource contention within a simple EA. With sharing, each optimum in the fitness landscape is allocated a finite number of resources relative to its size. These resources must be shared among individuals that exploit a given optimum. This introduces two benefits over a simple EA:
**Algorithm 2.1: The deterministic crowding algorithm**

**input**: A population of individuals of size $N$

**output**: The same population with reproduced individuals

1. $\text{population} \leftarrow \text{generateInitialPopulation};$

2. while not done do
   3. Shuffle $\text{population};$
   4. for $i \leftarrow 1$ to $(N - 1)$ step 2 do
      5. $p_1 \leftarrow \text{population}[i];$
      6. $p_2 \leftarrow \text{population}[i + 1];$
      7. $(c_1, c_2) \leftarrow \text{reproduce}(p_1, p_2);$
      8. if $[\text{dist}(p_1, c_1) + \text{dist}(p_2, c_2)] \leq [\text{dist}(p_1, c_2) + \text{dist}(p_2, c_1)]$ then
         9. if fitness($c_1$) > fitness($p_1$) then $p_1 \leftarrow c_1;$
         10. if fitness($c_2$) > fitness($p_2$) then $p_2 \leftarrow c_2;$
      else
         12. if fitness($c_2$) > fitness($p_1$) then $p_1 \leftarrow c_2;$
         13. if fitness($c_1$) > fitness($p_2$) then $p_2 \leftarrow c_1;$
      end
   14. end
   15. end

16. return $\text{population};$
1. Rather than attempt to crowd around a single peak, individuals within the population will actively seek out less populated optima. This ensures that selection pressure is always present to counteract the effects of genetic drift.

2. The effective value of a peak in space is reduced as the number of individuals exploiting it increases. This in turn makes the lesser-valued optima in the search space more attractive and selection will drive individuals toward them. This reduces the likelihood of losing the lesser peaks from the population.

In the crowding methods shown in the previous section, the fitness landscape did not change over time. In fitness sharing, the fitness landscape changes from generation-to-generation in response to the distribution of individuals. This implicitly creates a dynamic fitness landscape in which the optima become moving targets. Since the magnitude of optima are dynamic, individuals are constantly subject to selection pressure. An example of how sharing changes the fitness landscape is shown in Figure 2.3. The dashed line represents the distribution of fitnesses within the population sharing has been applied. As can be seen, the effect of sharing is to equalise the optima in the landscape. The largest optimum in the fitness landscape is more heavily populated and therefore individuals occupying this peak are subject to a greater reduction in fitness than those occupying the remaining peaks. This equalising effect of sharing helps to ensure that peaks in a fitness landscape are allocated individuals in proportion to their value and creates a stabilising force that reduces the likelihood of losing lesser-valued optima.

### 2.2.3 The implications of adopting fitness sharing

In order for sharing to work, some assumptions need to be made; it is assumed that the optima in the problem occupy similar-sized areas and that they are evenly distributed throughout the fitness landscape. This is due to the mechanism sharing uses to determine which resources are being consumed by each individual. Sharing uses a radius measure, called the sharing radius, to determine the amount of similarity between individuals and hence the niche count of an individual. Given a sharing radius $\sigma_{sh}$ and a distance between two individuals $d$, the niche count of an individual is defined as:

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{sh}}\right)^{\alpha}, & \text{if } d < \sigma_{sh}; \\ 0, & \text{otherwise}. \end{cases} \quad (2.1)$$

In the above function, $\alpha$ is a value that alters the shape of the function (typically, this is set to 1). The distance measure, $d$, is preferably in the phenotype space. In other words, distance should be measured as closely to the problem space as possible (Deb and Goldberg, 1989).
Figure 2.3: The effect of sharing on a fitness landscape. The solid line depicts the actual fitness landscape. The individuals of the population are represented as points on the fitness landscape. The dashed line depicts the fitness landscape after sharing has been applied to the population.
As shown in Algorithm 2.2, implementing fitness sharing is a simple extension to standard EAs. Its sole purpose is to dynamically alter the fitness of individuals after evaluation; in terms of implementation it plays no direct part in the fitness function. The other operators in an EA, such as selection and recombination, are equally unaffected.

```
input : A population of individuals of size \(N\)
output: The same population with reproduced individuals

1  \(\text{population} \leftarrow \text{generateInitialPopulation;}\)
2  \(\text{while not done do}\)
3    \(\text{foreach individual } i \in \text{population do}\)
4      \(m \leftarrow 0;\)
5      \(\text{foreach individual } j \in \text{population do}\)
6        \(d \leftarrow \text{dist}\,(i, j);\)
7        \(m \leftarrow m + \text{sh}\,(d);\)
8      \(\text{end}\)
9      \(i.\text{fitness} \leftarrow \text{evaluate}\,(i) / m;\)
10   \(\text{end}\)
11   \(\text{parents} \leftarrow \text{select}\,(\text{population});\)
12   \(\text{generation} \leftarrow \{\};\)
13   \(\text{for } i \leftarrow 1 \text{ to } (|\text{parents}| - 1) \text{ step 2 do}\)
14     \(p_1 \leftarrow \text{parents}[i];\)
15     \(p_2 \leftarrow \text{parents}[i + 1];\)
16     \((c_1, c_2) \leftarrow \text{reproduce}\,(p_1, p_2);\)
17     \(\text{generation} \leftarrow \{\text{generation}, c_1, c_2\};\)
18   \(\text{end}\)
19   \(\text{population} \leftarrow \text{generation};\)
20 \(\text{return } \text{population};\)
```

Algorithm 2.2: An evolutionary algorithm that incorporates fitness sharing.

Fitness sharing can be used with any selection method. However, some selection methods have been shown to affect the stability of sharing with respect to peak maintenance. Proportional selection schemes, such as stochastic universal sampling or stochastic remainder selection (Baker, 1987) are commonly used with sharing. Other selection methods, such
as tournament selection, can be used so long as the shared fitness of individuals is calculated as the new generation is being created, rather than before any offspring are produced (as for the normal sharing method) (Oei, Goldberg and Chang, 1991).

Criticisms of the sharing method

Fitness sharing is a well-known and extensively researched niching method (Mahfoud, 1995b). It has been shown to be effective for complex multimodal optimisation problems (Goldberg, Deb and Horn, 1992; Horn et al., 1994). Despite this success, it is often subject to criticisms. To use sharing, some assumptions about the given fitness landscape must be made; the ideal value for $\sigma_{sh}$ assumes a priori knowledge of the problem (Mahfoud, 1994; Darwen and Yao, 1995). Usually this information is unknown and $\sigma_{sh}$ must be estimated. In theory, an incorrect value for $\sigma_{sh}$ will result in sharing failing to locate and maintain all the optima in the problem. However, previous work testing the empirical properties of sharing suggests that it is not as sensitive to the value of the sharing radius as the theoretical models would suggest (Deb and Goldberg, 1989; Watson, 1999). Additionally, theoretical models exist that show that sharing is fairly reliable in the presence of niche overlap (Horn, 1997).

Another potential drawback to the use of sharing is the complexity of the algorithm; as sharing must compare all individuals to each other, its algorithmic complexity is $O(N^2)$. However, the complexity of sharing only forms part of the complete system; as noted by Mahfoud (1995b), the overhead of any niching methods is likely to be much less than that of the effort required to evaluate individuals for fitness.

Sharing has been shown to have difficulty in searching landscapes in which the number of undesirable optima far outweighs the number of desired peaks (Goldberg et al., 1992; Mahfoud, 1995a). This is because the undesirable optima interact with the larger peaks in a way that the fitness of any individual on a desired peak is reduced before the individual has had a chance to breed. Elitist selection methods are required to solve such problems, however elitism is not compatible with the standard sharing method (Mahfoud, 1995b; Pétrowski, 1996).

The issues of complexity and difficulty in searching certain fitness landscapes limit the applicability of sharing. In Chapter 5, these issues are addressed and a modified sharing algorithm is presented that has both lower complexity and improved search capabilities.
2.2.4 Clearing

Petrowski (1996) identified some of the above problems with sharing and devised an alternative method that would efficiently search for multiple optima in these difficult fitness landscapes.\(^1\) As with sharing, clearing is based around the concept of a carrying capacity of a niche. However, unlike sharing, which divides the resources among the individuals exploiting a niche, clearing rewards only the fittest individual; the weaker individuals occupying a niche in the fitness landscape have their fitness reduced to zero. This effectively removes the weaker individuals from the breeding pool and the end result of clearing is a diverse set of individuals, called the *winners*, containing the fittest member of each niche.\(^2\)

The clearing method uses distance measures (typically in the phenotype or space) to group individuals with similar characteristics into distinct subpopulations. Once the subpopulations have been established, clearing produces a diverse mating pool containing the fittest individuals from each subpopulation. This diverse set can be used to clearing’s advantage; as the mating pool contains only one representative of each subpopulation, it can be used to preserve good solutions without severely disrupting the distribution of individuals within the fitness landscape. This makes the clearing method significantly faster and more successful at searching these ‘sharing-hard’ problems (Pétrowski, 1996; Pétrowski, 1997b). There are several methods for performing elitism within clearing; one is simply to copy all winners into the next generation. This method can sometimes be too restrictive, so another method is to copy winners into the next generation under the condition that they are of above average fitness prior to clearing. This second method is the favoured strategy for most studies of clearing (Pétrowski, 1996).

The overall clearing algorithm is shown in Algorithm 2.3. The sorting of the population (line 3 in the algorithm) is important as it ensures that the fittest individual of each niche is dealt with first and hence preserved in the mating pool. Like sharing, the clearing method requires a radius of similarity to determine if two individuals belong to the same niche. This value, \(\sigma_{cl}\), takes on the same value for a problem as it would in the case of sharing.

The algorithm to breed a new generation using clearing is shown in Algorithm 2.4. This particular algorithm uses the ‘above average’ elitism method described above. In another similarity with sharing, the selection process when using clearing (line 7 of the algorithm)

---
\(^1\)A later study by Li *et al.* (2002) produced a very similar algorithm to that proposed by Petrowski. This study will not be discussed in this thesis and Petrowski’s work will form the representative clearing method.

\(^2\)Petrowski actually defined a parameter \(\kappa\), which controls the number of individuals kept for each niche. In most studies \(\kappa = 1\) and hence only one individual per niche remains after clearing. For the sake of simplicity, the parameter \(\kappa\) is removed from the discussion of clearing.
is often proportional to fitness. Typically stochastic universal sampling (Baker, 1987) is used, as this form of proportionate selection exhibits the least amount of drift (Pétrowski, 1996; Sareni and Krähenbuhl, 1998). However, because of the inherent bias toward fitter individuals as part of the niching process, clearing as a selection method in itself has also been proposed (Pétrowski, 1997b).

Clearing has been shown to outperform sharing in terms of number of optima maintained on a number of problems of varying difficulty (Pétrowski, 1996; Sareni and Krähenbuhl, 1998). Petrowski (1996) also states that clearing is around an order of magnitude faster than other niching methods at finding all the optima in a fitness landscape. This comparison is somewhat misleading however, as the numbers that Petrowski based his comparison on (taken from Mahfoud (1995a)) were for time to total population convergence, not the time required to locate all optima.

**Criticisms of the clearing method**

The clearing method is a powerful tool for searching for multiple solutions in a fitness landscape. However, like most niching techniques it has some drawbacks. Like sharing, clearing relies on a radius of similarity and studies using the clearing operator have shown that a correct value for this parameter is crucial if clearing is to locate and maintain all optima in a fitness landscape (Li et al., 2002; Parrott and Li, 2006). Alternative implementations of

---

**Algorithm 2.3: The clearing algorithm**

```plaintext
input : A set of individuals (the population) and the clearing radius $\sigma_{cl}$
output: A set of the dominant individuals in the population

1. $matingPool \leftarrow population$
2. Sort $matingPool$ in descending order of fitness;
3. for $i \leftarrow 1$ to $|matingPool|$ do
   4. $m_i \leftarrow matingPool[i]$;
   5. for $j \leftarrow i + 1$ to $|matingPool|$ do
      6. $m_j \leftarrow matingPool[j]$;
      7. if dist$(m_i, m_j) < \sigma_{cl}$ then remove the $j^{th}$ element from $matingPool$;
   8. end
4. end
10. return $matingPool$;
```

---
**input**: A set of individuals (the population) and the clearing radius $\sigma_{cl}$

**output**: A new generation of individuals

1. $matingPool \leftarrow clearing(population, \sigma_{cl})$
2. $\textbf{foreach} \ winner \ in \ matingPool \ \textbf{do}$
3. 3.1 $\textbf{if} \ fitness(current) \ \geq \ meanFitness(population) \ \textbf{then}$
4. 3.1.1 $generation \leftarrow \{generation, \ winner\}$
5. 3.1.2 $\textbf{end}$
6. $\textbf{end}$
7. $parents \leftarrow select(matingPool)$
8. $\textbf{for} \ i \leftarrow 1 \ \textbf{to} \ (|parents| - 1) \ \textbf{step} \ 2 \ \textbf{do}$
9. 9.1 $p_1 \leftarrow parents[i]$;
10. 9.2 $p_2 \leftarrow parents[i + 1]$;
11. 9.3 $(c_1, c_2) \leftarrow reproduce(p_1, p_2)$;
12. 9.4 $generation \leftarrow \{generation, c_1, c_2\}$;
13. $\textbf{end}$
14. $\textbf{return} \ generation$;

**Algorithm 2.4**: Breeding a generation using the clearing method.
clearing do away with the clearing radius in favour of a classification tree process (Pétrowski, 1997a; Pétrowski and Genet, 1999). However, these methods have requirements that, while outside the scope of this thesis, may present other problems to niching via clearing. First, the classification tree version of clearing requires random individuals to be created at each generation. This is to generate new points in the fitness landscape so that ‘valleys’ of lower fitness are established between peaks. The randomly-generated individuals will prevent an EA using clearing from converging, and hence may present difficulties in determining when to halt the EA. Second, the classification tree variant of clearing splits the fitness landscape into geometric intervals (Pétrowski, 1997a; Pétrowski and Genet, 1999). These intervals must be “small enough to differentiate close peaks” (Pétrowski, 1997a). This therefore reintroduces the problem of requiring some problem-specific knowledge.

2.2.5 Other niching methods

Crowding, sharing and clearing methods are the most well known and frequently encountered niching techniques. Other niching methods have been proposed (for example, Beasley et al., 1993; Ursem, 1999), but perhaps the most interesting of these is the ‘simple subpopulation schemes’ introduced by Spears (1994). This uses tag bits appended to the individuals to determine species. The number of tag bits determines the number of ‘species’ within the global population; for \( n \) tag bits there will be a maximum of \( 2^n \) species supported. Niching was promoted through the use of a sharing-like operator which was applied to the tag bits rather than the genotypic or phenotypic distances between individuals. The fitness of individuals was reduced by the number of individuals that shared the same tag. As an additional diversity-maintaining force, pairs of individuals could only mate if their species tags matched. Spears tested this method on four test problems and noted that the algorithm was able to find and maintain at least four out of five optima in each of the tested fitness landscapes.

2.3 Spatially-Structured Evolutionary Algorithms

The previous section discussed several methods of maintaining genetic diversity within the population of an EA. In all but one example, individuals were permitted to breed with any other member that survived the selection process. In essence, these EAs used random mating, or panmictic, populations. Species tagging was an exception in that it introduced assortative mating so that individuals may only mate with other members of the population with a similar species label. This introduced the concept of sympatric speciation into an EA (Mayr, 1970).
Species tagging is only one method in which an EA can incorporate speciation. Another method is to impose a spatial structure on the population so that there is a concept of geographic distance between individuals. Geographically close individuals form localised subpopulations, or *demes*, within the global population. Mating is confined to demes and offspring are placed within the vicinity of their parents. The basic mechanisms of evolution (mutation, selection and genetic drift) within each deme act in relative independence from the rest of the population. Thus each deme is encouraged to explore different regions of the fitness landscape. Through subsequent generations of local mating, the genotypes found in distant demes will sufficiently differ to the point that they could be considered different species.

Evolutionary algorithms that impose a geography on a population typically draw inspiration from two models from population genetics. The *island model* divides the global population into several smaller demes, which operate in isolation save for infrequent migration of individuals between demes. Alternatively, the *isolation-by-distance* model places the entire population onto a surface and establishes numerous, overlapping demes of individuals. Respectively, the two models are examples of *allopatric* and *parapatric* speciation (Mayr, 1970).

### 2.3.1 Multiple-deme evolutionary algorithms with migration

One approach to introduce spatial structure into an EA is to replace the single panmictic population with several smaller subpopulations, or islands. Each subpopulation (deme) is essentially an EA running in isolation. As each deme has a unique genetic makeup, there is a high probability that evolution will converge the subpopulations on different parts of the fitness landscape. Examples using multiple isolated demes are often referred to as *partitioned* EAs (Tanese, 1989).

The smaller demes of the partitioned population tend to evolve at a faster rate than if all the individuals were grouped into one panmictic population (Wright, 1932; Grosso, 1985). This property was used to good effect in previous work investigating Walsh polynomial functions (Tanese, 1989). However, on other problems the subpopulations also tend to converge at a lower level of fitness. To combat this premature convergence, EAs with multiple demes perform infrequent *migration* of individuals from one deme to another. Migration helps to increase diversity within each deme and allows evolution to continue when an partitioned EA may have converged. The rate at which migration occurs determines the isolation level of the demes; most researchers agree that a low rate of migration promotes the most diverse populations (Wright, 1932; Grosso, 1985).
The subpopulations within a multiple-deme EA tend to be panmictic. The topology of the population is defined by the migration paths between demes. The topology of these subdivided populations tends toward two models; full migration and migration between ‘neighbouring’ demes. A full-migration model occurs when each deme has a migration path to all other demes. This topology is analogous to the Wright’s island model from population genetics (Wright, 1932). The other possible topology is to impose some sort of geographical order on the demes and restrict migration to between neighbouring demes. This second approach draws inspiration from Kimura’s stepping-stone model (Kimura, 1953; Kimura and Weiss, 1964). The two types of topology are rarely distinguished in evolutionary computation and the blanket term island model EAs is usually used to describe all EAs that use multiple demes (Cantú-Paz, 2001).

Grosso (1985) performed an extensive study of island model EAs under various migration rates. His model was unique in that, in conjunction with a subdivided population, individuals were diploid and the fitness function used heterozygote advantage (refer to Appendix A). Grosso observed that the population as a whole maintained the most diversity when small, infrequent migrations were performed. Given a migration rate of one individual into a deme every four generations, Grosso was able to reliably converge each deme to a different peak in the fitness landscape. This phenomenon did not occur when no migration (i.e. a partitioned EA) was used and also disappeared when larger, more frequent migrations were used. Despite the evidence that migration played a pivotal role in maintaining overall population diversity, some researchers have argued that the role population structure played in the maintenance of multiple peaks is questionable. Instead it is argued that the diploid individuals in conjunction with heterozygote advantage may have provided most of the diversity maintenance (Mahfoud, 1995b). However, a second study into island model EAs also noted that overall diversity (compared to a panmictic population) was increased through the use of demes and migration (Tanese, 1989). This study used a more traditional, haploid bitstring representation of individuals, so heterozygote advantage could not play a part in diversity maintenance.

A more recent study of island model EAs (Skolicki, 2007) also examined overall genetic diversity, both at the intra-island level and at the global inter-island level. The study concluded that, while intra-island convergence is more rapid than a panmictic system (i.e. diversity within an island is lost quite rapidly), the overall inter-island diversity typically exceeded that of an unstructured EA. In addition to the insights that it provides on diversity within island model EAs, Skolicki’s work is also of interest because it introduces the concept of heterogeneous islands, where the representation of various aspects of the evolutionary
algorithm differs from island to island. In particular, the thesis studied the concept of heterogeneity with respect to problem encoding. By allowing each island to encode the problem parameters differently, overall genetic diversity was increased, which would tend to improve the overall search characteristics of the approach. Interestingly, this approach would appear to lend itself well to multimodal optimisation, as each island could be configured to target a unique portion of the fitness landscape, and therefore each island would be encouraged to converge upon different local optima. However, this concept was not explored in the thesis.

Another very recent study into multiobjective optimisation using island model EAs has produced a similar framework to that of Skolicki’s heterogeneous islands. In explicit niching (Bui, 2007), each island is encouraged to explore a different part of the search space via a set of local constraints specific to that island. Additionally, a migration policy inspired by the update mechanisms of particle swarm optimisation algorithms is incorporated into the framework. This intelligent migration policy is used to communicate global knowledge of the search status to each island, which in turn can use this information to update its local search strategy. Collectively, the approach demonstrated good convergence characteristics on a number of benchmark multiobjective problems.

Island model EAs have received a lot of attention in evolutionary computation literature. Until recently, the most in-depth analysis of these methods was performed by Cantú-Paz, who developed models for ideal migration rates, deme sizes and the migration topology between demes (Cantú-Paz, 2001). Additionally, the recent work by Skolicki (2007) offers a comprehensive analysis of the behaviour of island model EAs from various viewpoints. Together, the operations-centric approach of Cantú-Paz, and the more behavioural-oriented approach of Skolicki present a comprehensive review of island-based evolutionary algorithms and the reader is referred to this work for more information. The remainder of this thesis discusses population structures that impose geographical restrictions on selection and mating at an individual level.

2.3.2 Isolation-by-distance population structures

Island model EAs take inspiration from models developed in population genetics. The island and stepping-stone models are not the only population structures proposed in population genetics. A third model, the isolation-by-distance model, was proposed by Wright (1940; 1943). This model assumes that a population resides within a continuous, unbounded geometry that exists in one or two dimensions. Demes were defined by the geographical distances between individuals. Migration is not required in the isolation-by-distance model. Instead an individual can belong to multiple demes. The isolation-by-distance model therefore relies
on numerous overlapping demes within the population. The large distances between individuals in the population dramatically reduces gene flow and therefore different demes tend to evolve in different directions.

Evolutionary algorithms using the isolation-by-distance method go by various names: Manderick and Spiessens refer to their implementation initially as a fine-grained parallel genetic algorithm (Manderick and Spiessens, 1989) and later a massively parallel genetic algorithm (Spiessens and Manderick, 1991); Mühlenbein and Gorges-Schleuter called their method ASPARAGOS (Mühlenbein, 1989; Gorges-Schleuter, 1989); and Tomassini (1993) and Whitley (1993) both refer to isolation-by-distance EAs as cellular EAs. This thesis will refer to all of these variants with the term spatially-structured evolutionary algorithms (SSEAs).

An SSEA is essentially an implementation of the parapatric speciation concept; there are no complete boundaries within the population to restrict gene flow. Instead, SSEAs use the relatively vast distances between some individuals to slow the rate of exchange of genetic material to a sufficiently low level to promote local divergence of genotypes (Wright, 1943; Mayr, 1970). Though each SSEA implements a slightly different algorithm, the overall method for using a spatially-structured population within an EA is shown in Algorithm 2.5. Essentially, there are three basic steps to an SSEA. First, for a given location, construct the deme. After this, select the parents from the deme. Finally, insert the newly created offspring into the deme using some predefined criteria.

Spatially-structured EAs typically implement space as a one or two-dimensional lattice of locations, each of which contains a single individual. Typically the ends of the array are wrapped to form a ring topology or, in the case of a two-dimensional array, a torus. Examples of these topologies are shown in Figure 2.4. The structure of the population plays an important role in the behaviour of the algorithm; two-dimensional population structures tend to show less local divergence than a one-dimensional population (Wright, 1940). However, two-dimensional population structures tend to allow an EA to search a fitness landscape more rapidly than an equivalent-sized one-dimensional population. Recent research has investigated how the shape of the population can be changed dynamically to best accommodate the desirable properties of one and two-dimensional structures into a single SSEA (Alba and Dorronsoro, 2005).

The shape of the global population also plays an important role in determining the possible deme structures for each location in space (line 9 in Algorithm 2.5). In a one-dimensional topology, there is only one possible method of deme construction; demes are constructed by collecting the neighbours to the left and right of the current location (Figure 2.5). The num-
**Algorithm 2.5**: The general sequence for a spatially-structured evolutionary algorithm.

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>population ← {};</code></td>
</tr>
<tr>
<td>2</td>
<td><strong>foreach</strong> location in space <strong>do</strong></td>
</tr>
<tr>
<td>3</td>
<td><code>population[location] ← newRandomIndividual();</code></td>
</tr>
<tr>
<td>4</td>
<td><code>evaluate(population[location]);</code></td>
</tr>
<tr>
<td>5</td>
<td><strong>end</strong></td>
</tr>
<tr>
<td>6</td>
<td><strong>while</strong> not done <strong>do</strong></td>
</tr>
<tr>
<td>7</td>
<td><code>generation ← {};</code></td>
</tr>
<tr>
<td>8</td>
<td><strong>foreach</strong> location in space <strong>do</strong></td>
</tr>
<tr>
<td>9</td>
<td><code>deme ← constructDeme(location);</code></td>
</tr>
<tr>
<td>10</td>
<td><code>parents ← select(deme);</code></td>
</tr>
<tr>
<td>11</td>
<td><code>offspring ← reproduce(parents);</code></td>
</tr>
<tr>
<td>12</td>
<td><code>evaluate(offspring);</code></td>
</tr>
<tr>
<td>13</td>
<td><code>generation[location] ← pickSurvivor(offspring, deme);</code></td>
</tr>
<tr>
<td>14</td>
<td><strong>end</strong></td>
</tr>
<tr>
<td>15</td>
<td><code>population ← generation;</code></td>
</tr>
<tr>
<td>16</td>
<td><strong>end</strong></td>
</tr>
<tr>
<td>17</td>
<td><strong>return</strong> <code>population;</code></td>
</tr>
</tbody>
</table>
The circles represent the individuals of the population, while the lines represent the connections between individuals and are used to establish demes.

Figure 2.5: Deme construction for a ring topology. The grey element represents the central individual of the deme. The black elements represent individuals that will take part in the deme.

The number of individuals to collect in each direction is controlled by a parameter $d$, and every deme consists of $2d + 1$ individuals. Adding an extra dimension to the population allows for more possible deme shapes, the most common of which are the linear, compact and diamond topologies shown in Figure 2.6. Finally, some SSEAs take a more ad hoc approach to deme construction through the use of short random walks from a given location in space (Collins and Jefferson, 1991).

After the deme is constructed, two parents are selected for breeding (line 10 in Algorithm 2.5). The actual methods of selection vary from implementation to implementation; some SSEAs select one parent from the deme via fitness proportionate selection and a second parent at random from the remaining deme members (Manderick and Spiessens, 1989). Others use the current individual at a location as the first parent and select the second parent from the remaining deme members based on fitness (Mühlenbein, 1989; Gorges-Schleuter, 1989).
Another possibility is to pick the best individuals discovered through a short random walk starting from the location in which the offspring are to be created (Collins and Jefferson, 1991). Despite the differences in implementation, all selection methods for SSEAs share the common trait that selection of parents is confined to localised regions of the spatial structure.

Once parents have been selected, the next step is to create offspring (line 11 in Algorithm 2.5). Breeding in an SSEA is similar to that of a panmictic EA with one exception; usually, reproduction involving two parents in a simple EA will create two offspring. Most SSEAs however, only produce one offspring as a result of breeding. After the offspring is produced and evaluated, it is inserted into the deme from which it was created (line 13 in Algorithm 2.5). Some SSEAs choose to replace the current individual at the centre of the deme at every generation. Others choose to replace this individual under the condition that the new individual is better (elitist replacement). Other methods for replacement exist but, as for selection, the important concept is that offspring are placed within the population in a manner that preserves the geographic relationship with their parents.

2.3.3 Advantages of SSEAs

The incorporation of a population topology changes the behaviour of an evolutionary algorithm in many ways. These behavioural changes can be exploited in ways that make SSEAs more capable search methods than their panmictic counterparts. SSEAs are capable of being implemented on parallel machines without the use of a centralised controller (Gorges-Schleuter, 1989). Most early work into SSEAs was implemented on massively parallel machines like the Connection Machine (Hillis, 1986). These machines are quite rare now, however one can implement SSEAs on a network of computers using the message passing interface.
face standard (Dongarra, 1994; Tomassini, 2005). This highly parallel nature allows SSEAs to use very large populations, which in turn increases the search capabilities of SSEAs.

Spatially-structured populations exhibit an increased level of inbreeding within the population when compared with equivalent-sized panmictic populations (Wright, 1943; Gorges-Schleuter, 1992). This in turn promotes local differentiation of genotypes and encourages demes to explore disparate regions of the fitness landscape. In addition, each deme consists of a far smaller number of individuals than the total population. It is known that the rate of evolution of a population is inversely proportional to its size (Kimura and Ohta, 1969). Therefore, not only should the numerous demes of an SSEA explore different parts of the fitness landscape, but they should also do so at a faster rate than an equivalent panmictic population. The ability of SSEAs to locate and maintain multiple optima of a fitness landscape will be discussed in §2.3.5 and further examined in Chapter 3.

Each deme in a spatially-structured population is largely autonomous. The operations of selection, reproduction and evaluation are conducted in relative isolation and do not require significant amounts of information from the global population. Each location is essentially an evolutionary algorithm that works with a single individual (Sarma, 1998). Because of this, SSEAs lend themselves well to hybridisation with local search techniques which, in turn, allows an SSEA to use problem-specific knowledge when searching a fitness landscape. In Mühlenbein’s SSEA, all individuals undergo a local hillclimb before the selection and reproduction phases of the algorithm (Mühlenbein, 1989). Similarly, the ASPARAGOS method uses a problem-specific local search operator when searching the fitness landscape of a travelling salesman problem (Gorges-Schleuter, 1989).

The majority of research into SSEAs has concentrated on the emergent selection pressure of the population. In particular, the time required for a single fitter mutant to take over the entire population is used to determine the selection intensity for a given SSEA topology (Rudolph, 2000; Giacobini and Tomassini, 2003; Giacobini, Alba and Tomassini, 2003; Giacobini, Tomassini, Tettamanzi and Alba, 2005; Tomassini, 2005). Other investigations into selection have taken into account the variance of the selection process (De Jong and Sarma, 1995) and the effect the ratio between deme size and population size has on selection intensity (Sarma and De Jong, 1997). The behaviour of selection in an SSEA is often compared to that of a panmictic EA with the conclusion that selection in spatially-structured populations is qualitatively similar, but quantitatively less than for the same selection method when used in a panmictic EA (Sarma, 1998). The ‘qualitative’ component of selection relates to the probability that a given mutation will fix in a population, and work investigating selection in SSEAs has concluded that the probability of a mutant allele fixing in the pop-
ulation is no different than that of a panmictic EA. This conclusion is in agreement with previous studies from population genetics, which show that the probability of fixation of a mutant allele in a structured population is the same as that for an equivalent-sized panmictic population, so long as the fitness of the mutation is constant throughout space (Maruyama, 1974). When the fitness of a mutation depends on its location in space, the probability of fixation is no longer equivalent to a panmictic population. The effects that localised fitness have on fixation probabilities are discussed further in §2.3.6 and used in detail in Chapter 4 to create an effective niching SSEA.

2.3.4 Alternative models of SSEAs

Spatially-structured EAs have many possible implementations. The earliest SSEAs implemented a generational breeding policy; every location in the population was updated simultaneously. However, it is also possible to implement an SSEA in which only a few locations in any generation are updated. These SSEAs are often referred to as asynchronous algorithms (Gorges-Schleuter, 1989; Tomassini, 2005) because they resemble cellular automata with asynchronous update policies. Asynchronous update policies often demonstrate a greater selection intensity than the equivalent generational SSEAs (Giacobini et al., 2005). Asynchronous update policies have been shown to improve the performance of SSEAs on some problems (Tomassini, 2005).

The ECOlogical GA is an SSEA that implements the Uniform Choice update policy (Davidor, 1991). The ECO GA uses a two-dimensional torus. In each time step a location in the torus is chosen at random. Two parents are selected from the deme centred around this location and are used to breed two offspring. Two individuals are picked from the deme at random and each is paired up with one of the offspring. A probabilistic tournament is then held for each pair to determine if the current individual is to be replaced by the new one.

The ECOlogical GA was extended and used in a second study searching for solutions to a job scheduling problem (Davidor, Yamada and Nakano, 1993). This version implemented several significant changes over the first EGO GA. As before, a random location was picked at each time step. The individual at this location becomes the first parent for all offspring created in the reproduction cycle. The number of offspring to create is determined by the ratio between the fitness of the first parent and the mean fitness of the deme. For example, if this ratio is 1.4, 1 offspring is guaranteed to be created and there is a 40% chance of breeding a second individual. The second parent for each offspring was picked via fitness-proportional selection from the deme, excluding the first parent. Finally, for each offspring produced, an individual from the deme was picked and replaced if the offspring was fitter. This was tested against a set of job scheduling problems and was shown to outperform a panmictic EA.
Most SSEAs use a fixed array of discrete locations for individual placement. Wright’s initial ideas for the isolation-by-distance model assumed a continuous population structure in which parents were sampled via a normal distribution. The spatially-dispersed genetic algorithm (sdGA) extended normal SSEAs to use a continuous population surface (Dick and Whigham, 2002; Whigham and Dick, 2002; Dick, 2003b, c). Unlike standard SSEAs, which use a discrete cutoff function for deme participation, the sdGA used a correlation function which added a temporal element to deme participation (Dick, 2004). The intention is that clusters of individuals will form around fitter population members and that the sdGA will adapt subpopulation sizes to the problem at hand. The sdGA has been tested on some elementary problems and has shown some ability to outperform a panmictic population.

The communication between individuals within an SSEA is easily represented as a graph in which individuals are represented as vertices and the edges between these vertices represent a common deme participation between individuals. The concept of an SSEA was generalised into the form of Graph-Based Genetic Algorithms (GGBA) (Ashlock, Smucker and Walker, 1999; Bryden, Ashlock, Corns and Willson, 2006). GBGAs use an asynchronous update policy similar to the Uniform Choice method (Tomassini, 2005). In each time step, a vertex in the graph is picked at random to become the first parent. A deme is constructed by selecting all the vertices that share an edge with the randomly chosen vertex. An individual is selected from this deme using roulette wheel selection and mated with the first parent to produce a single offspring. This new individual then replaces the first parent if it is of at least equal fitness (local elitism). GBGAs offer a range of topologies that are not frequently encountered in traditional SSEAs, such as tree-like and fractal graphs. The GBGA method has been used to search for suitable solutions to the problem of placing baffles in a wood burning stove intended for use in Nicaragua (Bryden, Ashlock and McCorkle, 2004). Later studies have taken the GBGA approach, and in particular its ability to generate complex network structures, and applied it successfully to multiobjective optimisation tasks (Kirley and Stewart, 2007a, b).

The population size in an SSEA is usually determined at the start of a run and remains constant throughout the evolutionary search. The population is fully connected because there are no vacant locations in space. Kirley (2002) implemented the concept of disasters into an SSEA to create patchy disconnected groups of genetically distinct individuals in the population. This algorithm, called the cellular genetic algorithm with disturbances (CGAD), uses a torus as its population structure. Locations in the CGAD have three possible states: active sites were populated with an individual, vacant sites have the potential to locate an individual but are currently empty and inactive cells are unable to support an individual. Occasionally,
the CGAD picks a location in space and simulates a ‘disaster’ in the landscape by changing the sites surrounding the location to the inactive state. This disrupts the connectivity in the population and given enough disasters, will create smaller, disconnected subpopulations in the population structure. Eventually, the inactive cells are transferred from the inactive state to the vacant state. At this point, neighbouring locations can begin to repopulate the vacant cells. Kirley tested this SSEA on a range of multimodal, multiobjective and non-stationary problems and observed that the CGAD, under suitable parameter settings, was able to outperform a standard SSEA.

### 2.3.5 Multimodal problems and SSEAs

Many have noted the ability of SSEAs to maintain multiple niches for numerous generations. Mühlenbein, as part of the ASPARAGOS algorithm, placed individuals onto a thin two-dimensional torus that he termed a ‘ladder’. Mühlenbein noted that such a topology increased the amount of time required for genetic information to propagate throughout the population. For example, propagation times in a torus with $P$ individuals are $O(\sqrt{P})$, while the propagation time in an equivalent-sized ladder topology is $O(P)$. The longer diffusion times matched Wright’s isolation-by-distance model in that long, narrow population structures produce more divergent patches at a greater rate than square topologies (Wright, 1940). The ladder topology of the ASPARAGOS algorithm is shown to dramatically constrain the convergence of a population toward one peak, but it cannot promote stable subpopulations.

Speissens and Manderick (1991) examined the local differentiation properties of their massively-parallel genetic algorithm. They observed that small areas of homogeneity quickly appeared in the population. Through subsequent generations, some of these clusters would expand at the expense of other clusters. The rate that these clusters would expand was shown to follow quadratic growth. From this, Spiessens and Manderick concluded that although a spatially-structured population discovers multiple solutions during the course of a run, it will eventually converge to a single solution.

Davidor (1991) tested his ECOlogical GA on an unequal-peak multimodal problem taken from Goldberg and Richardson’s earlier study on fitness sharing (Goldberg and Richardson, 1987). Although not explicitly stated, the population size was in the thousands, which is much larger than the population size typically used for the test problem. Davidor noted that the population was able to support multiple solutions for over 5000 evaluations but admitted that “the ECO GA, as presented, does not claim to maintain niche & species ad infinitum” (Davidor, 1991, p. 262). Davidor’s work is frequently cited as an indicative example of an SSEA that can maintain multiple solutions for long periods of time (Mahfoud,
1995b; Deb, 2001). However, these studies fail to take into account the large population size and confuse function evaluations with full generations (the ECOlogical GA creates only two individuals per generation). Taking these two facts into account, the ECOlogical GA may actually perform worse than other SSEAs at maintaining multiple solutions within a single population.

Collins and Jefferson (1991) noted that their proposed local selection algorithm was able to regularly discover both solutions to their test function and maintain them within the population for at least 1000 generations. Collins and Jefferson observed the presence of patches of each solution in space in near equal proportions. They noted however, that their one-dimensional population structure was more stable than their equivalent two-dimensional topology with regards to maintaining both solutions. Between subpopulations of different solutions there were boundaries of low-fitness hybrids, or lethal individuals. Given that the solutions to their test problem were bit-wise compliments, the presence of these hybrids is not particularly surprising and may explain why Collins and Jefferson were successful at maintaining multiple solutions where others using the local random walk selection model failed (McInerney, 1992).

Spears, as part of his simple subpopulation schemes concept, used a ring topology in conjunction with a species tagging fitness sharing method (Spears, 1994). The sharing component operated once per generation and considered the population as a whole. After this step, selection and reproduction continued as for a typical SSEA. Spears noted that this scheme was more effective at maintaining multiple peaks in a fitness landscape than a simple subpopulation scheme that did not impose a population structure. It is not entirely clear that the increase in performance was a consequence of the combination of a ring and the sharing method, or if the improvement was purely attributable to the adoption of spatial constraints in the population. The issue is further complicated by the fact that the sharing component of the algorithm and the local mating did not interact; the ring topology played no part in determining the shared fitness of the population members. Spears himself noted this peculiarity and suggested a third subpopulation scheme which used the topology of the population within the sharing process (Spears, 1994). However, this new method has not appeared in any subsequent work. The combination of a spatially-structured population with a genetics-based niching method demonstrated several desirable behaviours. An investigation into the properties of these local niching techniques is presented in Chapters 5 and 6.

Dick (2003a) noted that the sdGA showed some ability to maintain multiple solutions to some simple multimodal problems by introducing very large distances between subpopulations. However, what was most likely happening in these tests was something akin to the
founder effect; a small number of individuals became separated from the rest of the population and formed new, reproductively isolated ‘colonies’ of genetically distinct individuals. To date the sdGA has not been tested on more difficult multimodal problems.

Kirley (2002) included some multimodal problems in his test suite for his CGAD. While the objective of the experiments in this study was to find the single global optimum in the problem, Kirley noted that the introduced disasters created patchiness in the space, with localised, genetically-distinct subpopulations emerging. A multiobjective problem was also included in Kirley’s test suite, and he observed that the CGAD could maintain multiple solutions from the Pareto front.

2.3.6 Parapatric speciation with SSEAs

Spatially-structured population are an attempt to introduce parapatric speciation into EAs. This form of speciation relies on very large distances between individuals to work (Mayr, 1970). Demes that are geographically far apart will tend to evolve in different directions for two reasons. In one case, genetic drift may affect gene frequencies differently for each deme. This is because genetic drift is direction-less; many events that are independent of selection may occur during mating and in one deme they may lead to the extinction of an allele and to its fixation in another. In addition to varying effects of drift, each deme may be exposed to different environmental conditions, hence selection will favour different adaptations.

An SSEA traditionally uses the same fitness function for each location in space; the same environment exists across the entire spatial-structure, so SSEAs typically implement only half the requirements for parapatric speciation. This might help to explain why most researchers using SSEAs observe that the population demonstrates temporary diversity and regions of differentiation that ultimately converge to a single genotype. What is needed for true parapatric speciation in an SSEA is a fitness function that varies depending on the location in space in which it is used. One solution may be to place a gradient on the surface of the population structure (Doebeli and Dieckmann, 2003). This will produce a series of different environmental conditions at various locations throughout space. The intention is that each environment will be best exploited by specific phenotypes. This encourages different parts of the population to seek out locally fit phenotypes which in turn should force the population as a whole to investigate multiple parts of the fitness landscape.

An example of using environmental gradients within an evolving spatially-structured population is described in an empirical investigation into genetic trade-off (Whigham and Green, 2004). A gradient consisting of 16 equal-sized regions was applied to a $100 \times 100$ grid. Each band in the gradient represented a different environmental condition, for example
the salinity of seawater in a salt marsh. Two experiments were conducted using this population configuration. The first experiment involved a trade-off between an individual’s ability to survive numerous gradients at the expense of lower fitness. A second experiment used individuals of equal fitness in all conditions, however, individuals that could survive in more environments took longer to reach reproductive maturity. Both experiments selected and reproduced individuals in a manner similar to that of a typical SSEA. The authors demonstrated in both cases that the distribution of individuals showed a clear differentiation of phenotypes and the simple mechanism of genetic trade-off was sufficient to produce genetic isolation that could eventually lead to speciation.

The work of Whigham and Green was not aimed at solving a specific problem. Rather, it investigated the use of genetic trade-off in an abstract framework. There have been relatively few examples of using environmental gradients for problem solving within the evolutionary computation community. The terrain-based GA (Gordon, Pirie, Wachter and Sharp, 1999) draped a gradient over a spatially-structured population in order to form combinations of parameter settings (crossover-points and mutation rates) for use within the evolutionary search for a single optimum. Here the gradient was used to find good parameters for the EA rather than to modify the fitness landscape. Those that have used a gradient for the purpose of altering the fitness landscape did so to search for solutions along the Pareto front of multiobjective fitness landscapes (Murata, Ishibuchi and Gen, 2000; Kirley, 2001). In each of these cases, the gradient was a vector that applied a weighting to each objective in the problem. This encouraged some locations to focus on separate objectives. Other locations in space placed an equal emphasis on objectives and their role was to effectively concatenate the good solutions found by other locations that focused on specific objectives. Each of these studies was able to find numerous solutions along the Pareto front of several benchmark two-dimensional multiobjective problems. For problems with more objectives, the authors of one of these studies proposed a dimensionality of space equal to the number of objectives in the problem, so a 10-objective problem would require a 10-dimensional population structure (Murata et al., 2000).

As mentioned in §2.1, although they share a number of similarities, multimodal and multiobjective problems are not equivalent; one cannot guarantee that an efficient algorithm for multiobjective problems will perform as well at searching a multimodal fitness landscape. To this end, the abilities of SSEAs with environmental gradients on multimodal problems is unexplored by the current literature on evolutionary computation. The performance of such methods on multimodal problems, the modifications required to their design to accommodate these problems and initial analysis of the behaviour of these algorithms are presented in Chapter 4.
2.4 Open Issues in Niching Using Spatially-Structured Populations

Spatially-structured evolutionary algorithms have been subject to considerable research. In particular, the behaviour of SSEAs with regards to selection intensity has received considerable attention (De Jong and Sarma, 1995; Sarma and De Jong, 1997; Giacobini et al., 2003; Giacobini and Tomassini, 2003; Tomassini, 2005). However, despite the effort put into researching SSEAs, we still do not have a thorough insight into the behaviour of these algorithms in the context of multimodal optimisation. Spatially-structured populations, while a conceptually simple extension to evolutionary search, dramatically alter the behaviour of the core operators of an EA. Also, SSEAs require additional parameters, such as topology and deme size, over that of a panmictic EA. It is therefore important that these aspects of SSEAs are considered in the context of multimodal problems. This chapter has surveyed the current literature pertaining to multimodal optimisation via EAs incorporating niching and spatially-structured populations. In the course of this review, several unresolved issues were identified which will be addressed in subsequent chapters:

Issue 1: Can genetic drift, as it manifests in SSEAs, be an effective tool in locating multiple optima in a multimodal landscape? Genetic drift in panmictic populations is well understood. Panmictic drift in natural and artificial populations is accurately modelled by the diffusion equation methods proposed in population genetics literature (Kimura and Ohta, 1969). Similarly useful equations model genetic drift within panmictic EAs (Asoh and Mühlenbein, 1994). Genetic drift in subdivided populations using the island or stepping-stone models is also well understood (Kimura and Weiss, 1964). Understanding of genetic drift in SSEAs resembling the isolation-by-distance model is limited; it is known that local mating in SSEAs increases the rate of inbreeding which, in turn, promotes local differentiation of genotypes (McInerney, 1992; Gorges-Schleuter, 1992; Sarma, 1998). Clearly, the behaviour of genetic drift in such models will be different to that of any other population structures, but to what degree? A more enlightened understanding of genetic drift in SSEAs is vital if a successful niching SSEA is to be developed.

Issue 2: Is there a class of multimodal problem in which current SSEAs, without modification, can discover multiple optima within a single run? Previous work reports different findings with respect to the success of SSEAs on multimodal functions. One study was able to support both solutions to a bimodal problem within
the same population over a sustained length of time (Collins and Jefferson, 1991). Other studies, however, are less decisive and suggest that SSEAs will typically converge upon a single solution (Davidor, 1991; Spiessens and Manderick, 1991; Sarma, 1998). Therefore, it would be useful to determine the classes of problem on which SSEAs can be successfully used to locate multiple optima. Properly characterising the behaviour of genetic drift within SSEAs (Issue 1) will be of use here.

**Issue 3: What are the implications of implementing true parapatric speciation within an SSEA?**

The parapatric speciation concept is based upon the theory that population structure produces two mechanisms that will bring about significant divergence of genotypes. Population structure will restrict mating interactions to geographically close individuals, promoting local divergence. In addition, each location in space presents different challenges/resources that will be best overcome/exploited by individuals with different characteristics. Virtually all SSEAs in previous studies provide one part of the parapatric speciation concept, namely localised mating. However, the fitness function applied to each individual at every location in space is identical. Therefore, a typical SSEA does not present novel conditions or environments that would promote local adaptation.

SSEAs with environmental gradients have shown some promise in multiobjective problems (Murata et al., 2000; Kirley, 2001). However, their suitability for use in multimodal fitness landscapes is untested. How should a gradient be created in order to support multiple peaks of multimodal problems? What about multidimensional fitness landscapes; how should the gradient be created to accommodate these problems? Typical niching methods allocate individuals to peaks in the fitness landscape depending on the relative fitness of the optima. Would this trend continue with a gradient-based SSEA? If not, how could one predict the size of each subpopulation that is allocated to each peak in the fitness landscape?

**Issue 4: Are SSEAs suited to hybridisation with existing niching methods?**

Spears (1994) proposed a model in which a sharing method is combined with a spatially-structured population. However this model has not been subsequently discussed in the evolutionary computation literature. A hybrid niching SSEA would be relatively straightforward to implement, but what would be the implications of using such a system? There are some problems that are known to be difficult to solve through standard niching methods such as sharing (Goldberg et al., 1992; Horn and Goldberg, 1995). The introduction of spatially-structured populations may improve the search characteristics of niching methods on these problems. Additionally, the impact that population structure has on the robustness of the niching method with respect to parameterisation must be investigated.
Chapter 3

The Behaviour of Genetic Drift in Spatially-Structured Populations

An evolving population is subject to two primary forces of evolution. The first, selection, acts to alter gene frequencies in response to relative fitnesses to a particular environment. The dynamics of selection in panmictic populations is well understood, and accurate models for selection exist in both the population genetics and evolutionary algorithm domains (Crow and Kimura, 1970; Robertson, 1962; Blickle and Thiele, 1996). Selection pressure in SSEAs has also been subject to a significant amount of research (De Jong and Sarma, 1995; Sarma and De Jong, 1997; Sarma, 1998; Giacobini et al., 2003; Giacobini, Alba, Tettamanzi and Tomassini, 2004). The conclusion of this research is that selection in SSEAs is qualitatively similar but quantitatively less than selection within an equivalent panmictic population (Sarma, 1998).

In addition to selection, any evolving population is subject to the force of genetic drift. Genetic drift results from the stochastic events that occur during reproduction in a finite population (Wright, 1931). Unlike selection, drift does not act on gene frequencies in any direction; it is selectively neutral. In the context of an evolutionary algorithm, an understanding of genetic drift is important; as a mechanism of evolution and reducer of variation in a population it has several implications, particularly in the context of multimodal problem solving (Mahfoud, 1994, 1995c).

The behaviour of genetic drift within the confines of a panmictic population is well understood. Genetic drift in the context of divided populations, such as the stepping stone model, has also been extensively studied (Kimura and Weiss, 1964). However, the nature of genetic drift in a spatially-structured population with overlapping demes has not been thoroughly explored. As a result, knowledge of how SSEAs behave on multimodal problems is incomplete.
This chapter analyses the behaviour of genetic drift in spatially structured populations. It begins with an empirical analysis of genetic drift in population structures commonly used in SSEAs: rings and tori (see Figure 2.4). These simulations demonstrate that, unlike selection, drift in a spatially-structured population is neither qualitatively nor quantitatively similar to an equivalent panmictic population. This leads to the formulation of models which accurately predict the fixation time of an allele within a ring-structured population.

The final part of this chapter examines the behaviour of a simple SSEA on two multi-modal problems with identical genotypes but markedly different phenotypes. As the genotypes of these problems are identical, we would expect the effects of genetic drift on them to be the same. However, the behaviour of an SSEA on these two problems is significantly different and this observation leads to two conclusions: SSEAs are capable of searching for multiple solutions to a problem when the relative fitness of peaks are equal; and any modifications to SSEAs to make them more general-purpose multimodal search methods should focus on modifying selection behaviour.

### 3.1 Properties of Genetic Drift in a Spatially-Structured Population

The goal of this chapter is to gain a better understanding of the dynamics of drift within a structured population with overlapping demes. Empirical observations must first be made if a theoretical model is to be formulated. To this end, several experiments were designed to establish the fixation time for ring-structured populations under different conditions. Although the model established in this chapter will only consider ring-structured populations, it will be shown in §3.3 that the observations made in this section apply to other population structures commonly encountered within evolutionary computation. Therefore, the model proposed here should be applicable to other SSEAs with only minor modification.

A common measure for the scale of genetic drift within a system is the time required for all the individuals within a population to assume the same genotype in the absence of selection and mutation. This measure, the *fixation time*, is adopted for this study. Unless otherwise stated, the initial condition for each experiment in this chapter is a randomly-generated population sampled from a uniform distribution.
3.1.1 Effect of population size and deme size on genetic drift

A well known property of genetic drift within panmictic populations is that the mean time for a population to fixate on a single allele value is linear with respect to the population size (Kimura and Ohta, 1969). Figure 3.1 shows a plot for the mean time to fixation for a ring structure as the population size increases. In all cases, a deme was constructed by taking the individual at a given location, along with both of its nearest neighbours. These results indicate that fixation times in a spatially-structured population grow quadratically with respect to population size, and a model is presented in §3.2.2 that appears to confirm this.

We have shown that ring-structured populations take longer on average than unstructured populations to converge to homogeneity. This was for a fixed deme size, where \( d = 1 \). Obviously, the value of \( d \), and the corresponding deme size, will also play a part. Indeed, given the value of \( d = \frac{N}{2} \), the behaviour of a ring structure should converge to panmixia, as every individual takes part in every deme. Figure 3.2 shows the mean time to fixation for a given ring-structured population as \( d \) increases. As the value of \( d \) approaches \( \frac{N}{2} \), the system converges to panmixia.
Figure 3.2: Simulation of genetic drift on a ring with respect to increasing deme size (population size $N = 100$). The population behaviour approaches panmixia as $d$ increases. The panmictic fixation time was calculated via a diffusion model approximation of genetic drift (Equation (B.9)).
3.1.2 Effect of ploidy on genetic drift

Evolutionary algorithms tend to operate on haploid individuals, where each locus of a chromosome contains a single allele value. However, diploid individuals, who possess two alleles at each locus, frequently occur in nature. A diploid individual is said to be heterozygous at a locus if the locus contains two different allele values. When the alleles at a given locus are equal, an individual is considered to be homozygous.

An interesting aspect of diploidy is the concept of heterozygote advantage. Typically, homozygotes and heterozygotes possess a relative fitness. If the heterozygote fitness is greater than that of the homozygotes, then heterozygotes will be favoured for mating. This should tend towards promoting diversity within the population. Indeed, given suitable conditions, selection for heterozygotes will delay the fixation of a population by several orders of magnitude (Robertson, 1962). Grosso was also able to maintain a diverse population in a system using heterozygote advantage in conjunction with an island model population structure with migration (Grosso, 1985). It is unclear which component of Grosso’s system was primarily responsible for the diversity maintenance, although others have argued a case for heterozygote advantage (Mahfoud, 1995b).

Heterozygote advantage has the potential to promote diversity within a population. However, such a method cannot be used within a spatially-structured population without first analysing the effect that drift in such a population has on heterozygosity. The effect of spatially-structured genetic drift on heterozygotes is not directly known. Previous studies have used an inbreeding coefficient to measure the diversity of a population (Gorges-Schleuter, 1992; McInerney, 1992; Sarma, 1998). In each case, haploid individuals were used, and the measure was taken at the time of mating. The results presented in these studies indicate that inbreeding in spatially-structured populations was increased, with the implication that this reduced the number of heterozygotes in the population.

The previous studies of inbreeding did not directly model diploid individuals, so the results from these investigations can only be considered an approximation of heterozygote counts within the population. Figure 3.3 shows the effect of deme size on the number of heterozygotes within a population of diploid individuals in a ring-structured population. Each experiment is compared with the expected number of heterozygotes for an equivalent-sized panmictic population at time $t$, given by the equation:

$$H_t = H_0 \left(1 - \frac{1}{2N}\right)$$

where $H_0$ is the number of heterozygotes present in the population in the initial generation (Hartl and Clark, 2006). At small deme sizes, the number of heterozygotes within the
Figure 3.3: Comparison of decay of heterozygosity in a ring structure and a panmictic population. Small values of $d$ result in a rapid elimination of heterozygotes from the ring.

population drops rapidly, although more heterozygotes are preserved in later stages of the run than for a panmictic population.

The results shown here indicate that the high levels of inbreeding quickly eradicate most of the heterozygotes from within the population. However, a small number of heterozygotes persist for a long time. Where exactly do these individuals reside? Figure 3.4 provides some insights into this. In this figure, the evolution over time via genetic drift for two populations is shown. In Figure 3.4(a) the population members are haploids, while Figure 3.4(b) represents a population of diploid individuals. As can be seen, heterozygotes within the diploid population solely exist at the boundaries between homozygous subpopulations. Any subpopulations that attempt to break off into independent groups are quickly engulfed by the other genotypes. This raises an interesting point; although heterozygotes are eradicated quickly, both homozygous genotypes remain in the population in reasonable numbers. Since
low numbers of heterozygotes survive into later generations, heterozygote advantage would probably not increase the overall diversity in the population. In practical terms, this may have a negative impact on the use of heterozygote advantage as a diversity maintaining mechanism within spatially-structured EAs. However, the use of such a diversity maintenance method may not be necessary in a ring-structure; the high levels of inbreeding in a ring seem to maintain reasonable proportions of both alleles in the system without the addition of heterozygote advantage.

Figure 3.4 raises another point of interest; aside from the small number of heterozygotes present in the diploid population, the overall dynamics of the diploid and haploid populations are very similar. This would imply that the dynamics of genetic drift in a spatially-structured population may operate independently of genetic representation of an individual. To confirm this, the mean fixation times for identically structured populations of haploids, diploids and tetraploids (4 alleles per locus) were found. These values are shown in Figure 3.5(a). The
curves for each of the three genotypes follow very similar trends, in fact if the panmictic fixation time for each genotype is removed from the curves we can see that they coincide, as shown in Figure 3.5(b).

3.2 A Model for Genetic Drift in Ring Topologies

The empirical investigation into genetic drift on a ring has provided many insights into the behaviour of random mating on a ring. What is needed now is a model that encompasses as much of the observed behaviour as possible in a simple and compact manner. Our case study for this section is a ring-structured population of size \( N = 4 \) with \( d = 1 \). As before, a two-allele, single locus model of individuals will apply. This value of \( N \) gives us a population with 16 possible configurations. In addition, the value of \( d = 1 \) provides eight possible deme structures: 000, 001, 010, 011, 100, 101, 110 and 111.

A suitable starting point in our quest for a model describing drift in a ring might be to take the Wright-Fisher model (Appendix B) and alter the transition matrix to take into account mating on a ring. Such an attempt would quickly meet with complications however, as the Wright-Fisher model assumes that all transient steps are reachable from each other via a single step. In the case of a ring, this assumption no longer applies. For example, say we have a current population of 1101 and a target population 0101. Under the Wright-Fisher model (Equation (B.3)), a panmictic population will move from the current population to
the target with a probability of 0.421875. However, for the ring population, the left-most location is fixed for the given time step, as the deme centred around that location is 111. Therefore the transition from the current population to the target is not possible. Therefore, the Markov chain must take all unique populations into consideration.

For the example ring of size $N = 4$, there are $2^N$ possible populations. The transition probability from one population $X$ to another population $Y$ is given by:

$$\prod_{i=1}^{N} \Pr\{\text{Next individual at location } i = Y_i | \text{Deme centred around } X_i\}$$  

(3.1)

The transition matrix for our example ring is then:

$$P = \frac{1}{81}$$

\[
\begin{bmatrix}
12 & 12 & 6 & 0 & 0 & 0 & 12 & 6 & 6 & 3 & 0 & 0 & 0 & 24 \\
12 & 12 & 6 & 12 & 6 & 6 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 24 \\
8 & 8 & 16 & 2 & 4 & 4 & 8 & 2 & 4 & 4 & 8 & 1 & 2 & 2 & 4 & 4 \\
0 & 12 & 0 & 12 & 0 & 6 & 0 & 12 & 0 & 6 & 0 & 6 & 0 & 3 & 0 & 24 \\
2 & 8 & 4 & 2 & 1 & 4 & 2 & 8 & 4 & 16 & 8 & 4 & 2 & 8 & 4 & 4 \\
2 & 8 & 4 & 8 & 4 & 16 & 8 & 2 & 1 & 4 & 2 & 4 & 2 & 8 & 4 & 4 \\
0 & 3 & 6 & 0 & 0 & 6 & 12 & 0 & 0 & 6 & 12 & 0 & 0 & 12 & 24 & 0 \\
12 & 0 & 0 & 12 & 6 & 0 & 0 & 12 & 6 & 0 & 0 & 6 & 3 & 0 & 0 & 24 \\
8 & 2 & 4 & 2 & 4 & 1 & 2 & 8 & 16 & 4 & 8 & 4 & 8 & 2 & 4 & 4 \\
8 & 2 & 4 & 8 & 16 & 4 & 8 & 2 & 4 & 1 & 2 & 4 & 8 & 2 & 4 & 4 \\
3 & 0 & 6 & 0 & 6 & 0 & 12 & 0 & 6 & 0 & 12 & 0 & 12 & 0 & 24 & 0 \\
2 & 2 & 1 & 8 & 4 & 4 & 2 & 8 & 4 & 4 & 2 & 16 & 8 & 8 & 4 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 6 & 6 & 12 & 6 & 12 & 12 & 24 & 0 \\
0 & 0 & 0 & 3 & 6 & 6 & 12 & 0 & 0 & 0 & 0 & 6 & 12 & 12 & 24 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 81 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 81 \\
\end{bmatrix}
\]

where the index into the matrix for each population is given by a simple binary-to-integer mapping (0000 at the left and top, 1111 at the bottom and right).

As with the standard Wright-Fisher model, we can compute the fundamental matrix $W$ of $P$ (Equation (B.4)), and hence the mean time to fixation. Whereas each row in the matrix for the Wright-Fisher model covers multiple population instances, each row in $W$ for a ring corresponds to a unique population. Therefore, following Equation (B.6), the mean time to fixation for a ring-structured population becomes:

$$\tau = \sum_{i=1}^{2^N-2} p_{n_i} (1 - p)^{N-n_i} E_i$$  

(3.2)
Figure 3.6: Comparison between simulation and extended Wright-Fisher model for genetic drift on a ring.

where \( p \) is the initial frequency of 1 alleles in the population, \( n_i \) is the number 1 alleles in the state represented by \( i \) and \( E_i \) is the absorption time (Equation (B.5)). Figure 3.6 shows the accuracy of this extended Wright-Fisher model for various combinations of population size, deme size and initial allele frequency. As can be seen, the agreement between model and empirical data is very good.

3.2.1 Random walks

The Markov model gives an exact value for the expected number of generations until fixation in a ring-structured population. Unfortunately, the model’s usefulness is hindered by the large transition matrices it requires. This limitation is commonly encountered with Markov chain models of evolutionary algorithms (Nix and Vose, 1992). A simpler, more compact
approach is needed in order to be able to compute any useful predictions on time to fixation for populations of reasonable size.

Random walks are a commonly encountered model of phenomena within the realms of physics and biology (Berg, 1983). The theory behind random walks has been used previously to describe the behaviour of certain aspects of evolutionary algorithms (Harik, Cantú-Paz, Goldberg and Miller, 1999). Indeed, the fixation of an allele in a panmictic population via genetic drift can be viewed as a random walk (Crow and Kimura, 1970).

Four rules are present in the simple model of the random walk (Berg, 1983). First, it is assumed that a particle has constant velocity, that is, it moves a constant amount at each time step. Second, a particle is never stationary; it must move at each time step. Third, the probabilities of a particle moving either left or right are constant; at each iteration the particle will move left with a probability $p$ and to the right with a probability of $q = 1 - p$. Finally, the movement of a particle in space is independent of any other particles present in the system. When the above assumptions hold, information about the position of a particle after a certain number of steps can be easily established. Another important aspect that can be measured is the time it takes on average for a particle to move from its starting position to an absorbing boundary.

**Walks with absorbing boundaries**

Imagine a particle in one-dimensional space as shown in Figure 3.7. The space is of length $b$. The particle performs a random walk through the space, from an initial point $x$. At each time step the particle moves a distance $\delta$. The particle has an equal chance of moving left or right ($p = q = 0.5$), and the probability of not moving at any time step is zero.

At each end of space there is a boundary. If the particle reaches a boundary, it is considered captured and will no longer move. A property of such a random walk is that the particle is guaranteed to eventually reach one of these boundaries. Given this, the expected time to
capture, \( W \), is established via the differential equation (Berg, 1983):

\[
\frac{d^2W}{dx^2} - \frac{1}{D} = 0.
\]

(3.3)

In this equation, \( D \) is a quantity known as the diffusion coefficient. \( D \) is a value that describes the rate of movement of a particle in space and is defined by the following equation (Berg, 1983)\(^1\)

\[
D = \frac{\delta^2}{2}.
\]

(3.4)

In order to determine \( W \), some boundary conditions are required. The mean time to capture at a boundary is \( W(0) = W(b) = 0 \), and therefore the equation to model \( W \), derived from (3.3) gives:

\[
W(x) = \frac{1}{2D}(bx - x^2)
\]

(3.5)

where \( x \) is the initial point in space. If the particle starts at the furthest point from the boundaries, namely \( \frac{b}{2} \), the equation reduces to

\[
W\left(\frac{b}{2}\right) = \frac{b^2}{8D}
\]

(3.6)

Substituting the equation for \( D \) into (3.6) yields the following function for mean time to capture:

\[
W\left(\frac{b}{2}\right) = \left(\frac{b}{2\delta}\right)^2
\]

(3.7)

### 3.2.2 Genetic drift in structured populations as a random walk

Recall from Figure 3.5(b) that the fixation time required by a ring-structured population is approximately equal to the panmictic fixation time for a given genotype plus a component that appears to be independent of genotype. This would imply that an equation to model the time to fixation in a ring should take the form of:

\[
\tau_{\text{ring}} = \tau_{\text{panmixia}} + X
\]

(3.8)

where \( X \) is a function characterising the ring component of genetic drift. An insight into a possible solution for \( X \) is given by Figure 3.4; in each case the overall characteristic of drift is similar, with large blocks of homogeneity forming. Each of these subpopulations then slowly expands or contracts over time until the subpopulation becomes extinct or takes over the whole ring. In essence, the boundaries of each subpopulation are performing a random walk

---

\(^1\)The equation for \( D \) also has a parameter \( \tau \) which is the interval over which distance samples are measured. As used here, \( \tau = 1 \) and hence can be removed from the equation.
to an absorbing boundary. This idea is emphasised when we consider the actual dynamics of mating within a ring. An individual in a ring-structured population is only allowed to mate with individuals from within its deme. In order for the genes of two disparate individuals to interact, they must travel through space via the overlapping demes. A ring is simply a line in which the ends meet with the distance between neighbouring individuals defined as 1. The maximum distance between two individuals in a ring of size \( N \) is therefore \( \frac{N}{2} \). When an individual mates with its neighbours, parts of its genotype are placed in the neighbouring locations. The genetic information of that individual is now available to new, previously unreachable, demes. The probability of this mating occurring to the left or the right of the individual is equal. At each generation, the flow of genes may reach out to new demes, or may move back towards the deme from which they originated.

The description above for the behaviour of genes moving back and forth between demes meets the criteria for a random walk as outlined in §3.2.1. It is therefore possible to use the mean time to capture described previously as a basis for modelling fixation in a ring structure. Taking Equation (3.7) as a starting point, two things are needed: the distance between absorbing boundaries \( b \), and a mean displacement at each time step, \( \delta \). A value for \( b \) is fairly straightforward, we set it to the furthest distance between any two points on a ring, which is \( \frac{N}{2} \). The value for \( \delta \) is not as simple to define; at any time step a gene in a deme of length \( d \) may move \([0 : d]\) steps from its starting point in the ring. The random walk model assumes a uniform, non-zero displacement of a particle at each time step. One way to approximate this requirement would be to take the mean displacement of a gene for a single time step. Given a deme length of \( d \), \( \delta \) becomes:

\[
\delta(d) = \frac{d^2 + d}{2d + 1}
\]

and the final equation for the random walk model for a ring of size \( N \) is then:

\[
\tau_{\text{ring}} = \tau_{\text{panmictic}} + \left( \frac{N}{4\delta(d)} \right)^2.
\]

Figure 3.8 shows a comparison between this model and data acquired through simulation.

### 3.2.3 Genetic drift in spatially-structured populations with multi-locus individuals

So far, genetic drift on a ring has been confined to single locus models. A single locus system is useful to describe the overall dynamics of a ring topology subject to genetic drift, but it does not encompass the behaviour of systems using more realistic representations. An
Figure 3.8: Comparison between simulation of genetic drift and the proposed random walk model.
extension to the proposed model is therefore required in order to understand the behaviour of genotypes with multiple loci within a ring structure.

One method to model arbitrary length genotypes is to take the view that each locus is a separate particle in space undergoing a random walk. Each random walk is running in parallel independently from the other loci in the system. As is shown in Figure 3.9, the observed time for all $n$ particles to be absorbed is $\log_e n$ times the expected time for a single particle. A similar logarithmic trend for panmictic populations was established by Asoh and Mühlenbein (1994). The equation to model fixation in a ring topology with $n$ loci is therefore extended to become:

$$
\tau_r(N, n, d) = \tau_p(N, n) + (\log_e n + 1.0) \left( \frac{N}{4\delta(d)} \right)^2.
$$

A comparison between this model and empirical data for various combinations on population size and number of loci is shown in Figure 3.10. $R^2$ correlation values for these comparisons are presented in Table 3.1. As can be seen, the agreement between empirical data and the proposed model is very high.

Figure 3.9: The change in fixation time due to increased number of loci in the genotype. The increase in time is approximately linear with respect to the logarithm of the number of loci.
Figure 3.10: Comparison between simulation and model for mean generations to fixation via genetic drift on a ring with multiple loci.
Table 3.1: $R^2$ correlation values for the random walk model of genetic drift on a ring of size $N$ and $d = 1$.

<table>
<thead>
<tr>
<th>N</th>
<th>1-loci</th>
<th>10-loci</th>
<th>50-loci</th>
<th>100-loci</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.997</td>
<td>0.979</td>
<td>0.985</td>
<td>0.995</td>
</tr>
<tr>
<td>100</td>
<td>0.998</td>
<td>0.991</td>
<td>0.997</td>
<td>1.000</td>
</tr>
<tr>
<td>500</td>
<td>0.996</td>
<td>0.996</td>
<td>0.999</td>
<td>0.999</td>
</tr>
<tr>
<td>1000</td>
<td>0.997</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2.4 Limitations of proposed model

The proposed model (Equation (3.11)) matches the given simulations to a high degree. It does however have some limitations, one of which is that it is restricted to two-allele systems with equal initial frequencies. However, this scenario is frequently encountered in genetic algorithms using bitstring representations, and therefore the model is still of considerable use.

The proposed model for drift on a ring measures the mean time to fixation for a ring-structured population. It does not differentiate between cases where a given allele is lost from the population (extinction) and when the allele takes over the entire population (becomes established). The Wright-Fisher model, and many of its derivatives also make this assumption (Watterson, 1962; Ewens, 1963). Kimura and Ohta (1969) demonstrated that these two events may be modelled separately through the use of diffusion equations. While such a distinction is not important for modelling bitstring GAs, it may have uses for dealing with representations with more than two alleles, as will be shown in §3.3.2.

3.3 Other Properties of Spatially-Structured Genetic Drift

The chapter so far has developed a model of genetic drift suitable for use in bitstring GAs with ring-structured populations. There are other representations and population structures in common use within the field of evolutionary computation. Although a full analysis of all these combinations is outside the scope of the thesis, this section analyses the behaviour of some of these systems and suggests directions to extend the proposed models for these situations.
3.3.1 Genetic drift in two-dimensional structures

Another spatial structure frequently encountered is that of the two-dimensional torus. The torus introduces some additional complexities over the ring; demes in a ring have only one dimension and hence there is only one way to construct them. The additional dimension of the torus allows for more complex deme structures to be developed, such as the linear, compact and diamond demes (Sarma, 1998). In addition, the torus has shorter distances between population members; in a ring of size \( N \), the greatest distance between any two population elements is proportional to \( N \), in a torus the distance is proportional to \( \sqrt{N} \). The torus therefore has a tendency to show faster takeover times (Giacobini et al., 2003, 2004).

Even though there are significant differences between the ring and torus structures, the overall behaviour is similar. Figure 3.11 shows the time to fixation via genetic drift for a torus with various ploidy levels. As with the ring, fixation times rapidly decrease as the deme size increases. Similarly, the additional time required for the torus to converge (over an equivalent size panmictic population) is independent of the particular genotype of individuals. This is supported by the visualisations of two randomly chosen simulations of haploids and diploids shown in Figure 3.12. Again, like the ring, there is a rapid formation of homogenous subpopulations, and in the case of diploids, there is a corresponding rapid elimination of heterozygotes.

The model for genetic drift proposed in this chapter is based on one-dimensional random walks, and therefore is not applicable to the two-dimensional torus. However, two-
Figure 3.12: Evolution of a torus-structured population over time for haploids and diploids. Homozygotes are presented in blue or yellow, with heterozygotes highlighted in green. As with the ring structure, the majority of heterozygotes in diploid population are quickly eliminated and the remnants exist largely between the boundaries of homozygotes. Large homogeneous subpopulations also quickly appear within the population.
dimensional random walks can be modelled via the equation (Berg, 1983):

\[ \nabla^2 W + \frac{1}{D} = 0. \]  

(3.12)

Solutions to this equation would require a value for mean displacement that is dependent on deme-structure and are beyond the scope of this thesis.

### 3.3.2 Genetic drift with more than two alleles

The discussion of genetic drift thus far has been limited to populations with two initial allele states. This provides insights into how spatially-structured populations will behave in the context of bitstring individuals. However, often an evolving population will contain many unique alleles. The behaviour of genetic drift in a spatially-structured population when multiple allele values are present is then of interest.

In the case of genetic drift, all alleles are equally fit. If a population contains an even distribution of \( n \) alleles, the probability of converging to a given allele is \( \frac{1}{n} \). Asoh and Mühlenbein (1994) suggest that, for the purposes of modelling fixation, an \( n \)-allele system may be represented as a two allele system where one of the alleles, labelled \( A \), has an initial frequency of \( \frac{1}{n} \). The mean time to fixation for the \( n \)-allele system is then the same as the mean time for the equivalent two-allele system to converge to ‘all \( A \)’ (i.e. ignoring cases where \( A \) becomes extinct).

Figure 3.13 shows a comparison between the time to fixation for a ring population with multiple alleles versus an equivalently initialised two-allele system. In each case, the two-allele system was initialised with \( p = \frac{1}{n} \), where \( n \) is the number of alleles in the multi-allele system. Additionally, the times for the two-allele system ignore runs where the chosen allele became extinct. As can be seen, the two different systems behave almost identically. This concurs with the findings of Asoh and Mühlenbein (1994) and would imply that future work in modelling genetic drift in spatially-structured populations should focus on separating extinction and fixation and should also incorporate varying initial allele frequencies in a two-allele system.

### 3.4 Implications of SSEAs as Multimodal Optimisers

Recall that previous researchers have had mixed results in multimodal optimisation with SSEAs. However, the results of this chapter indicate that the genetic drift properties of spatially-structured populations should allow for the maintenance of diversity for sufficient
Figure 3.13: Comparison of genetic drift on a ring with multiple alleles and with two alleles for a given initial allele frequency.
lengths of time. This would suggest that the behaviour of selection in SSEAs, not genetic drift, is the probable cause of failure in previous work.

The impact that selection has on the ability for an SSEA to maintain multiple optima is easily tested. Consider the two problems \textit{M4} and \textit{MMDP} (see Appendix E); both problems use a bitstring genotype of 30 bits in length. In the absence of selection, the effects of drift on both problems will be identical. However, the two problems have entirely different phenotypes, and hence the behaviour of selection on both problems will be markedly different. Additionally, the optima in \textit{M4} are of unequal value. Therefore, if there is any observable difference in the performance of an SSEA on these two problems, it must be due to the nature of selection in an SSEA.

A toroidal SSEA (with selection) with population size 1024 was run using the two problems \textit{M4} and \textit{MMDP}. The basic parameter settings followed those described in Appendix C, and the D5 deme construction method was used (Sarma, 1998). The number of optima present in the population was measured at each generation. Rather than run the SSEA for a prescribed number of generations, a halting window technique (Beasley et al., 1993) was used to terminate the EA once the average fitness of the population had stopped improving. For each problem, the SSEA was compared with a panmictic EA. The results averaged over 100 runs are shown in Figure 3.14. The SSEA halted execution on average after 62 generations (with a standard deviation of 3) and at this time maintained around 20 of the 32 optima. However, on \textit{M4} the selection properties of the SSEA quickly overwhelm its genetic drift properties; the SSEA halts after only 41 generations and has maintained only the fittest optima.
optimum. In both cases, the panmictic EA did not halt until a single optimum was left in the population.

Visualisation of a typical run of an SSEA on these problems helps to explain this behaviour, as seen in Figure 3.15. In each population the presence of a species (as determined by the method outlined in Appendix D) is represented by a coloured cell, while a white cell indicates the absence of an optimal species. On M4, small clusters of optima are immediately present in the population. However, selection does not allow these optima to form any significant clusters in the population, and by generation 14, only two optima remain in the population to any significant degree. Finally, at generation 42, all but the fittest optima have disappeared from the population. The behaviour of the SSEA on MMDP is dramatically different and is much closer to the patterns shown in Figure 3.12. Like M4, small clusters of optima quickly form in the population. However, because each optima has the same fitness, they are able to stabilise in the population without being overwhelmed by selection. When the SSEA finally halts (on generation 60), reasonable numbers of optima are still present in the population. These results confirm the original hypothesis stated earlier; it is the selection properties of SSEAs that limit their ability to maintain multiple solutions over fitness landscapes that present optima of varying fitness.

The conclusion drawn from this chapter is that behaviour of genetic drift within spatially-structured populations make them ideal candidates as the basis for EAs for multimodal op-
timisation. A simple EA, as typically implemented in previous work, is capable of maintaining multiple optima in a fitness landscape so long as the desired optima are of equal fitness. However, this makes the suitability of an SSEA entirely dependent on the nature of the fitness landscape; if the fitness of peaks varies even slightly, selection will bias the fittest optimum, which will then propagate throughout the entire population. Therefore, the behaviour of selection in SSEAs must be appropriately managed in order to make a useful multimodal optimiser based upon spatially-structured populations. In the following chapters, we will investigate possible methods for managing selection within SSEAs so that they are more reliable multimodal optimisers.
Chapter 4

Environmental Gradients, SSEAs and Parapatric Speciation

The previous chapter demonstrated the limited niching abilities of spatially-structured evolutionary algorithms. In particular, it was noted that in order for SSEAs to become reliable niching EAs, they must incorporate mechanisms that control selection and prevent takeover of the population by the single fittest optimum.

Spatially-structured EAs are considered to be an implementation of the parapatric speciation concept (Mayr, 1970). Indeed, localised mating and imposed reproductive isolation via vast distances are properties that both parapatric speciation and SSEAs employ. However, SSEAs are lacking a vital component of the parapatric speciation concept; in a traditional SEA, the fitness of an individual is not tied to its location in space. The same fitness function is used throughout space; consequently, if a phenotype exists that is globally optimal, it will naturally propagate to all locations in space, leaving no areas for less-fit optima. However, in the parapatric speciation model, an individual’s fitness is heavily dependent on the environment in which it resides; if the conditions of the environment vary with space, then it is expected that the individual’s fitness would change were it to move to a new location. If this property could be implemented within SSEAs, it would likely increase the ability of an SSEA to converge upon several optima of a fitness landscape using a single run.

The purpose of this chapter is to analyse the behaviour of SSEAs that incorporate environmental gradients. The empirical and theoretical properties of the gradient-based SEA (GBSSEA) approach are investigated through two simple one-dimensional problems. In particular, the conditions under which a GBSSEA is expected to maintain all optima, and likewise the conditions for when it is expected to fail, are identified. Once the elementary properties of the GBSSEA have been determined, an investigation into its behaviour when
applied to harder problems is investigated, with particular attention to its behaviour on problems in which deception is present.

Most evolutionary algorithms that are designed to identify multiple solutions are expected to produce niche-proportionate populations, that is, a population in which the individuals are allocated to optima in the fitness landscape in proportion to their relative fitnesses. This requires global knowledge of the fitness landscape. This chapter will demonstrate the niche-allocation properties of the GBSSEA. In particular, it will demonstrate that the allocation of individuals to optima in a GBSSEA does not follow the ‘expected’ model; instead, the GBSSEA allocates individuals to optima through a process that uses the relative fitnesses of neighbouring optima, rather than global information of the fitness landscape.

The final part of this chapter investigates the use of randomly-generated rugged environments. The empirical evidence presented suggests that using rugged environments may help increase the number of inter-species recombinations events, which may be useful in problems in which crossover between existing optima produces new optima with high probability. Additionally, it will be shown that the rugged approach to gradient generation may be useful when the phenotype of a given problem is difficult to define, for example in the case of many bitstring-encoded problems.

### 4.1 A Localised Environment for SSEAs

Before describing the implementation of the gradient-based SSEA approach, it is necessary to define the concept of local fitness. An individual $i$ resides at an arbitrary location $l$. The local fitness of $i$ is therefore its objective fitness (its fitness with respect to the actual problem being searched) multiplied by its response to the environment presented at $l$:

$$\text{localFitness}(i, l) = f_i \cdot \text{response}(i, l)$$

where $f_i$ is the fitness of the individual with respect to the fitness landscape of the problem. For most SSEA implementations, the response function is simply:

$$\text{response}_{\text{SSEA}}(i, l) = 1$$

in other words, local fitness of an individual is determined purely through the problem’s fitness landscape.

Unlike the environment presented by a simple SSEA, real-world environments vary with location. For example, the temperature at the North and South Poles is significantly different to that found at the Equator. Species that thrive at the Equator would not be expected to do
so well at either Pole (and *vice versa*). Essentially, the temperature gradient encourages local adaptation at the expense of general fitness. Figure 4.1 shows a possible implementation of this environmental gradient concept suitable for use in an SSEA. In this case, there are two phenotypic traits, A and B. In the case of trait A, the ideal phenotype varies with each column in the grid. Likewise, the ideal phenotype for trait B varies with each row in the grid. When combined, they create an environment that produces all combinations of ideal phenotypic traits. Note also that, as the basic population structure is a torus, the sides of the grid wrap to meet with similar ideal phenotypes.

The purpose of the environment is to create a fitness function that introduces a location-varying element while still permitting identification of the optima of the problem’s fitness landscape. Implementing an environment is relatively straightforward; we start by introducing the concept of an *ideal phenotype*, which is the set of phenotypic traits representing the characteristics individuals that are best suited to a given location. Each location will have an ideal phenotype and it is expected that the characteristics for the ideal phenotype will vary throughout space. Once the ideal phenotype is established, we need a way of measuring an individual’s response to the presented environment. This can be done through a gaussian function similar to that proposed by Doebeli and Dieckmann (2003). For an individual $i$ residing at location $l$, the response becomes:

$$
\text{response}(i, l) = e^{-\frac{1}{2} \left( \frac{\Delta(p_l, p_i)}{\sigma} \right)^2}.
$$

(4.2)

Here $p_i$ and $p_l$ represent the phenotypes of the individual and location (respectively). The operator $\Delta(p_l, p_i)$ measures the dissimilarity between the two phenotypes; for real-coded phenotypes this is simply a Euclidean distance measure. Finally, $\sigma$ controls the rate of decay.
for the environmental response; high values of $\sigma$ de-emphasise the environmental response, while smaller values place a high emphasis on optimising toward the gradient over the problem space. Figure 4.2 shows the effect of various values for $\sigma$ on the response function.

The impact of the response function on the resultant fitness landscape is clear in Figure 4.3, which shows the local fitness landscape for the $M4$ problem in the presence and absence of an environmental response. Without the use of a gradient, four out of the five optima in the fitness landscape are dominated by the fittest peak and are therefore not expected to survive. However, with the introduction of the gradient, individuals that exploit each of the five optima can find locations in the gradient in which they are locally the fittest phenotype. Therefore, the chances of maintaining all optima are greatly increased.

The algorithm for the GBSSEA method is shown in Algorithm 4.1. The overall approach, besides the calculation of fitness, is virtually identical to that of a simple SSEA (Algorithm 2.5)
input: A given problem
output: A spatially-structured population of evolved candidate solutions to the problem.

1 population ← {};
2 foreach location in space do
3     population[location] ← newRandomIndividual();
end

while not done do
1 generation ← {};
2 foreach location in space do
1     c ← Individual currently at location;
2     deme ← constructDeme(location);
3     parents ← select(deme);
4     o ← breedOffspring(parents);
5         $f_o$ ← evaluate(o) $\times$ response(o, location);
6         $f_c$ ← evaluate(c) $\times$ response(c, location);
7         if $f_o > f_c$ then
8             generation[location] ← o;
9         else
10             generation[location] ← c;
11         end
12 end
13 population ← generation;
end

return population;

Algorithm 4.1: The general sequence for a spatially-structured evolutionary algorithm that incorporates an environmental gradient.
Figure 4.3: The effect that response to a local environment has on the overall fitness landscape using the M4 problem. Note that, when a local environment response is introduced, the fitness landscape is modified so that each optima maintains a region in the environment in which it is locally the fittest solution.

4.2 Elementary Behaviour

The two problems, $M1$ and $M4$ (see Appendix E), were selected to analyse the basic behaviour of the GBSSEA approach. While these functions are very simple, they present problems for a simple SSEA. Additionally, their simple construction means that the underlying properties of the GBSSEA can be identified easily.

In the initial experiments, the GBSSEA uses a population size of 400 within a $20 \times 20$ torus. This population size is typically larger than that used in previous experiments for these two problems. However, this population size was specifically chosen as it emphasises the behavioural aspects of the GBSSEA.

Both $M1$ and $M4$ are one-dimensional problems. Therefore, the environmental gradient has the same number of phenotypic traits. The following method was used to construct the environmental gradient for the GBSSEA: first, each column in an $N \times N$ grid was labelled from 0 to $(N-1)$; then, given a location with row $r$ and column $c$, the gradient was computed by:

$$ p_t(r, c) = |1 - x_c| $$

where $x_c$ is defined as:

$$ x_c = \begin{cases} \frac{2c+1}{N+(N \text{ mod } 2)}, & \text{if } c < \frac{N}{2} \\ \frac{2c+1+(N \text{ mod } 2)}{N+(N \text{ mod } 2)}, & \text{otherwise.} \end{cases} $$

An example gradient for a $20 \times 20$ torus is shown in Figure 4.4.
<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20</td>
</tr>
</tbody>
</table>

**Figure 4.4:** A simple definition of a gradient for M1 and M4 problems using a $20 \times 20$ torus.
The GBSSEA was run with the base configuration outlined in Appendix C. Additionally, demes were constructed using the D5 approach (Sarma, 1998), that is the individual at the current location and its north, south, east and west neighbours\(^1\). Selection for parents within demes was performed using local fitness (Equation (4.1)). A value of \(\sigma = 0.25\) was chosen for the response function as it seemed to be a suitable intermediate value for this parameter; an analysis of the effects of this parameter on overall algorithm behaviour is presented in \(\S\)4.3.2.

The maximum peak ratio performance (see Appendix D) of the GBSSEA approach, compared to a simple SSEA, is shown in Figure 4.5. As can be seen, for the \(M1\) problem, the SSEA slowly loses peaks from the population; given enough time, one would expect the population to converge to a single optimum. Conversely, the GBSSEA has no problems maintaining all five optima. A similar trend appears with respect to the \(M4\) problem. In this case, the SSEA quickly converges upon the single fittest solution. As with \(M1\), the GBSSEA approach fares much better, with a stable maximum peak ratio of 0.934. However, unlike the \(M1\) problem, the GBSSEA fails to maintain all five optima.

The chi-square-like performance (see Appendix D) of the SSEA and GBSSEA is shown in Figure 4.6. In a similar trend to the maximum peak ratio performance, the distribution of individuals among optima produced by the SSEA slowly deviates from the expected distribution. This deviation is considerably quicker on the \(M4\) problem. The chi-square-like performance of the GBSSEA is much closer to that of the expected distribution; in the case of \(M1\), the distribution produced by the GBSSEA approach quickly reaches perfection, indi-

\(^1\)In cellular automata literature, this would be referred to as a Von-Neumann neighbourhood.
cating that the ideal number of individuals per optimum has been reached. The performance of the GBSSEA on \( M4 \) is again much improved with respect to the SSEA. However, it is still some way off from the distribution expected to be produced by a niching evolutionary algorithm.

A possible explanation for the performance of the two SSEA approaches is shown in Figures 4.7-4.10. Here, a typical run for each algorithm on both problems is displayed, with a snapshot of the population shown for generations 0, 10, 20, 50, 100 and 200. In each population the presence of a species (as determined by the method outlined in Appendix D) is represented by a coloured cell, while a white cell indicates the absence of an optimal species. In the case of \( M1 \), a simple SSEA is able to quickly form homogeneous subpopulations of optima within the population (Figure 4.7). However, by generation 20, it is already starting to lose optima from the population, and by generation 200 only individuals from two optima remain. The behaviour of the SSEA on the \( M4 \) problem (Figure 4.8) shows a similar trend, however because the desired optima are of varying value, the rate that optima are removed from the population is increased.

The behaviour of the GBSSEA is markedly different to that of the standard SSEA. Although starting from a similar initial population, the GBSSEA quickly forms ordered bands of locally-optimal individuals within the \( M1 \) problem. By generation 50, the population has stabilised to a point where all 5 optima from the problem are represented (Figure 4.9). In the case of \( M4 \), the GBSSEA cannot maintain the lowest-valued optima, but otherwise the behaviour it presents is similar to that shown when applied to the \( M1 \) problem (Figure 4.10). An explanation for why the GBSSEA could not support lowest-value optimum in \( M4 \), and a prediction of the conditions needed to ensure its survival are presented in §4.3.1.
4.3 Analytical Properties

The results obtained by running the GBSSEA on \( M1 \) and \( M4 \) produced some interesting observations. The behaviour of the GBSSEA on \( M1 \) perfectly matched that expected of a niching EA; all optima in the fitness landscape were discovered and maintained and the expected distribution of individuals to these optima exactly matched the expected values. However, the behaviour of the GBSSEA on \( M4 \) displayed qualities that need further investigation if the GBSSEA can be used as a general purpose multimodal search method.

The GBSSEA managed to maintain four out of the five optima in the \( M4 \) fitness landscape on every run. However, although the fifth optimum was regularly discovered, the GBSSEA was unable to maintain it within the population for a sustained length of time. The reason for this behaviour is straightforward to determine. The ideal phenotypes presented by the gradient are 0.05, 0.15, 0.25, \ldots, 0.95. The local fitness of each optimum in the \( M4 \) problem at locations with these ideal phenotypes is shown in Table 4.1. Upon inspection of these local fitnesses, it is clear that there is no ideal phenotype within the gradient that promotes the fifth optimum as the locally fittest solution. Hence, the fifth optimum is outcompeted at all locations in space, giving it no chance of being maintained within the population.

**Table 4.1:** The local fitness values of the optima in the \( M1 \) and \( M4 \) problems against each possible ideal phenotype when using a \( 20 \times 20 \) torus. The fittest phenotype for each location has been highlighted. Note that there are no positions in the gradient that favour the fifth optimum of the \( M4 \) problem.

<table>
<thead>
<tr>
<th>Optima ((M1)) Phenotype ((p))</th>
<th>Fitness</th>
<th>Ideal Phenotype at Gradient Location ((l))</th>
<th>localFitness ((p, l))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100</td>
<td>1.000</td>
<td>0.980 0.980 0.835 0.607 0.375 0.198 0.089 0.034 0.011 0.003</td>
<td></td>
</tr>
<tr>
<td>0.300</td>
<td>1.000</td>
<td>0.607 0.835 0.980 0.980 0.835 0.607 0.375 0.198 0.089 0.034</td>
<td></td>
</tr>
<tr>
<td>0.500</td>
<td>1.000</td>
<td>0.198 0.375 0.607 0.835 0.980 0.980 0.835 0.607 0.375 0.198</td>
<td></td>
</tr>
<tr>
<td>0.700</td>
<td>1.000</td>
<td>0.034 0.089 0.198 0.375 0.607 0.835 0.980 0.980 0.835 0.607</td>
<td></td>
</tr>
<tr>
<td>0.900</td>
<td>1.000</td>
<td>0.003 0.011 0.034 0.089 0.198 0.375 0.607 0.835 0.980 0.980</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optima ((M4)) Phenotype ((p))</th>
<th>Fitness</th>
<th>Ideal Phenotype at Gradient Location ((l))</th>
<th>localFitness ((p, l))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.080</td>
<td>1.000</td>
<td>0.993 0.961 0.793 0.557 0.334 0.170 0.074 0.027 0.009 0.002</td>
<td></td>
</tr>
<tr>
<td>0.247</td>
<td>0.949</td>
<td>0.696 0.880 0.948 0.871 0.681 0.454 0.258 0.125 0.052 0.018</td>
<td></td>
</tr>
<tr>
<td>0.451</td>
<td>0.770</td>
<td>0.213 0.374 0.558 0.710 0.770 0.712 0.560 0.376 0.215 0.105</td>
<td></td>
</tr>
<tr>
<td>0.681</td>
<td>0.503</td>
<td>0.021 0.053 0.113 0.209 0.328 0.438 0.499 0.484 0.404 0.282</td>
<td></td>
</tr>
<tr>
<td>0.934</td>
<td>0.250</td>
<td>0.000 0.002 0.006 0.016 0.038 0.077 0.131 0.191 0.236 0.250</td>
<td></td>
</tr>
</tbody>
</table>
Figure 4.7: Emergence of species over time for the M1 problem using a simple SSEA.

Figure 4.8: Emergence of species over time for the M4 problem using a simple SSEA.
Figure 4.9: Emergence of species over time for the M1 problem using a GBSSEA.

Figure 4.10: Emergence of species over time for the M4 problem using a GBSSEA.
There are two possible methods that would allow the GBSSEA to maintain all five optima of the $M4$ problem. The first solution is to ensure that sufficient granularity of ideal phenotypes is produced within the gradient, which is investigated in the following section. An alternative solution could be a shift towards more emphasis on local adaptation via a lower value of $\sigma$ for the response function. Stronger local adaptation would help to promote survival of the fifth optimum. The method by how this should be done, and the implications of doing so, are discussed in §4.3.2.

4.3.1 Equivalence points

The previous section identified some interesting properties with regards to the behaviour of the GBSSEA method. In particular, the snapshots of the population over time revealed that the niching properties of the GBSSEA produced distinct bands of species within the population. The boundaries between two niches are of particular interest; this suggests that a single fittest peak cannot take over the entire population as there exists at least one location in space where it no longer represents the fittest solution. This raises the question - is it possible to determine a location in space in which two competing optima possess equivalent local fitness?

We can use Equation (4.1) to help answer this question. Ultimately, we are trying to find a location $l$ that is between two optima $x$ and $y$ such that the relationship:

$$local\text{Fitness}(x, l) = local\text{Fitness}(y, l)$$
holds. In the context of the GBSSEA, this will be true when:

\[
f_x e^{-\frac{1}{2} \left( \frac{\Delta(p_x,p_x)}{\sigma} \right)^2} = f_y e^{-\frac{1}{2} \left( \frac{\Delta(p_y,p_y)}{\sigma} \right)^2}.
\]

Rearranging this equation to solve for \( p_l \) gives:

\[
p_l = \frac{2\sigma^2 \log_e \left( \frac{f_x}{f_y} \right) + p_y^2 - p_x^2}{2(p_y - p_x)}.
\] (4.5)

The equivalence points for each of the five optima of the \( M1 \) and \( M4 \) problems is shown in Figure 4.11. Here, each colour represents the ‘territory’ of the phenotype space that each optima occupies. The boundary between colour changes identifies the equivalence point for neighbouring optima. In the case of \( M1 \) the equivalence points line up perfectly with the boundaries of each optimum, which explains the excellent chi-square-like performance of the GBSSEA on this problem. However, the equivalence points for optima in \( M4 \) do not line up well with the optima boundaries of the fitness landscape; from the fittest optima down, the territory occupied by each optimum increasingly encroaches on the phenotype space of the next optimum. In the most extreme case, the final optimum has a very small region in which it is the locally-fittest optimum.

**Table 4.2: The equivalence points between peaks for the \( M4 \) function.**

<table>
<thead>
<tr>
<th>Optimum (x)</th>
<th>0.080</th>
<th>0.247</th>
<th>0.451</th>
<th>0.681</th>
<th>0.934</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.080</td>
<td>-</td>
<td>0.183</td>
<td>0.309</td>
<td>0.452</td>
<td>0.608</td>
</tr>
<tr>
<td>0.247</td>
<td>0.183</td>
<td>-</td>
<td>0.412</td>
<td>0.555</td>
<td>0.712</td>
</tr>
<tr>
<td>0.451</td>
<td>0.309</td>
<td>0.412</td>
<td>-</td>
<td>0.682</td>
<td>0.838</td>
</tr>
<tr>
<td>0.681</td>
<td>0.452</td>
<td>0.555</td>
<td>0.682</td>
<td>-</td>
<td>0.981</td>
</tr>
<tr>
<td>0.934</td>
<td>0.608</td>
<td>0.712</td>
<td>0.838</td>
<td>0.981</td>
<td>-</td>
</tr>
</tbody>
</table>

Equation (4.5) can be used to formulate a table of locations where any two peaks in the fitness landscape become equivalent, which is presented in Table 4.2. From this table, we can see that the fitness of peaks four and five are equivalent at locations where the ideal phenotype = 0.981. This explains why the GBSSEA was unable to maintain all five optima within the population. The method used in this thesis to generate the gradient is unable to produce a location in space with this value. In fact, in order to provide enough granularity in the gradient so that a location with this ideal phenotype is created, a minimum population size of 2704 individuals (a \( 52 \times 52 \) torus) is required. To test this, the runs of the GBSSEA on the \( M4 \) problem were repeated, this time using the larger population size. The maximum peak ratio and chi-square-like performance are shown in Figure 4.12, while visualisations of
Figure 4.12: The maximum peak ratio and chi-square-like performance of the gradient-based SSEA approach using a 52 × 52 torus.

Figure 4.13: Emergence of species over time for the M4 problem using a Gradient-Based SSEA and a 52 × 52 torus. The gradient includes a single ideal phenotype that promotes the fifth optimum as the locally fittest solution.

the population over time are shown in Figure 4.13. As expected, the GBSSEA is now able to support all five optima, with the fifth optimum occupying a single column of cells on each side of the population.
Figure 4.14: A ‘piecewise’ tunable three-peak function. The parameters $\alpha_1$, $\alpha_2$ and $\alpha_3$ control the relative heights of the peaks.

**Expected peak survival**

Equation (4.5) can be used to determine the equivalence point of any two optima in the fitness landscape. In the $M4$ problem, the fifth optimum was effectively being crowded out of the population; as it lies close to the edge of possible phenotypes there were no locations presented by the gradient where it could effectively reside as the fittest representative. The fitness of the fifth optimum in $M4$ is one quarter that of the global optimum. In a real-world problem, it is unlikely that we would want to maintain such an optimum, however it is important to note this behaviour as a potential limitation of the GBSSEA approach.

A related problem to that presented by the GBSSEA is the situation in which a lesser-valued optimum lies between two optima of greater value. For example, consider the simple three-peak function that is shown in Figure 4.14 and defined as:

$$PW(x, \alpha_1, \alpha_2, \alpha_3) = \alpha_i [1 - |6x - 2i + 1|]$$

where $0 \leq x \leq 1$ and $i = \lceil 3x \rceil$. This function has three optima and the height of each peak is tunable through a parameter $\alpha_i$. In Figure 4.14, the values of $\alpha_1$ and $\alpha_3$ are set to 1.00 and 0.75 respectively. Using these values, it is important to know for which values of $\alpha_2$ we can expect the GBSSEA to maintain all three optima within a single population.
Figure 4.15: Emergence of niches over time with the three-peak problem ($\alpha_2 = 0.300$).

Figure 4.16: Emergence of niches over time with the three-peak problem ($\alpha_2 = 0.400$).
As configured in Figure 4.14, the equivalence point for peaks one and three is found at a location with an ideal phenotype of 0.527 and the local fitness of each peak at this location is 0.354. Therefore, in order to ensure the survival of the second peak, the local fitness of the middle peak at this location must be greater than 0.354. Given that the fitness of the second peak is equal to $\alpha_2$, a suitable value for this parameter is when the condition

$$\alpha_2 > \frac{0.354}{e^{-\frac{1}{2}(\frac{\Delta(0.527, 0.5)}{\sigma})^2}}$$

holds. This means, to ensure survival of the second optimum, the value of $\alpha_2$ must be at least 0.357 or greater. To test this, the GBSSEA was run on two configurations of the three-peak function; once with a value of $\alpha_2 = 0.300$, and once with $\alpha_2 = 0.400$. The population plots over time for randomly selected runs are shown in Figures 4.15 and 4.16. As can be seen, the GBSSEA is unable to maintain the second peak when $\alpha_2$ is set to 0.300. However, it is able to maintain a reasonable number of individuals on the second peak when $\alpha_2 = 0.400$.

### 4.3.2 Parameter sensitivity

The GBSSEA approach introduces two new parameters to the basic SSEA method. The first is the actual representation used to specify the ideal phenotype at location in the gradient; in many cases (for example, in real-parameter optimisation) this is given to us by the nature of the problem, and so is relatively easy to define. The second parameter, $\sigma$, defines the strength of the environmental response function. The role of this parameter is to encourage a suitable balance between optimisation of individuals to the actual problem being solved, and optimisation to the local environment to improve overall population diversity. Determining suitable values for this parameter is not straightforward, as one must play the balance between problem and environment optimisation carefully.

The effect of the value of $\sigma$ on the territories dominated by each species in the $M1$ and $M4$ problems is shown in Figure 4.17. In the case of $M1$, it can be seen that the territories that each species dominates does not vary with respect to the value of $\sigma$. An examination of Equation (4.5) explains this behaviour; if two optima have the same fitness, the term $2\sigma^2\log_e \left( \frac{f_x}{f_y} \right)$ in the equivalence point equation will always equal zero. Therefore, in the case of equal optima, Equation (4.5) simplifies to become:

$$p_t = \frac{p_x + p_y}{2}$$

In constrast to the $M1$ problem, Figure 4.17 shows that the value of $\sigma$ plays an important role in the territories dominated by each species of $M4$. As the value of $\sigma$ increases, the dominant
Figure 4.17: The effect of $\sigma$ on the equivalence points between species in the M1 and M4 fitness landscapes. The equivalence points in M1 are particularly robust with regards to different values of $\sigma$. However, the equivalence points in M4 are heavily dependent on the value of $\sigma$. 
range of the fittest species is expanded. This increasingly leaves no room for the less fit species and, as such, moderate to high values of \( \sigma \) are unable to support all the species in the fitness landscape.

Based upon these findings, the initial assumption would be that a general purpose value for \( \sigma \) would be small to ensure that all species are given sufficient territories to permit their maintenance. However, this ignores one important factor; the introduction of the gradient turns the fitness landscape into a multiobjective problem. In essence, there is a trade-off between the system optimising to the fitness landscape of the actual problem, while at the same time evolving towards the values presented by the gradient. The role of \( \sigma \) in this context is to control the emphasis placed on these two factors; high values of \( \sigma \) place a greater emphasis on the ‘real’ problem at the expense of overall diversity, while smaller values of \( \sigma \) encourage strong selection towards the gradient (at the expense of the actual problem being searched). This is confirmed in Figure 4.18, where the gradient-based SSEA fails to find all optima in both \( M1 \) and \( M4 \) with a value of 0.01 for \( \sigma \). In the case of \( M1 \) selection pressure toward the gradient was so strong that no optima were maintained. In the case of \( M4 \), several optima were maintained, however this is because three ideal phenotypes in the gradient happened to coincide with three optima in the problem’s fitness landscape. If another population size was used, even if only slightly different, the resultant gradient would shift the ideal phenotypes away from these peaks and the number of species discovered in \( M4 \) would have been even lower.

The evidence provided here suggests that a moderate value of \( \sigma \), in the region of 0.1–0.3, is suitable if the goal in mind is to search for all species irrespective of relative fitnesses. If, on the other hand, the goal is to find only the globally optimal species, then a large value for \( \sigma \) is appropriate.

### 4.4 Extended Behaviour

So far, the GBSSEA has been tested on simple, one-dimensional problems. These problems are useful, however they do not test the GBSSEA’s ability to scale to harder problems in terms of the number of optima to be discovered and the difficulty of a fitness landscape that includes deceptive optima. The GBSSEA was therefore tested against more problems, with the intent of gaining further insights into its behaviour as a general purpose multimodal search algorithm.

The first problem, Foxholes (Appendix E), adds another dimension to the problem, and increases the number of optima to find to 25. Being a real-parameter problem, the gradient
Figure 4.18: The effect of $\sigma$ on the maximum peak ratio and chi-square-like performance of the gradient-based SSEA approach.
Figure 4.19: The maximum peak ratio and chi-square-like performance of the gradient-based SSEA approach on the Shekel’s Foxholes problem.

was again relatively simple to implement; each location had an ideal phenotype with two traits, determined by:

\[ g_{r,c} = \{|1 - x_r|, |1 - x_c|\} \]  

(4.6)

using \(x_i\) as defined by Equation (4.4). The resultant gradient has a similar appearance to that presented earlier in Figure 4.1.

The GBSSEA was run on the Foxholes problem using a 289 population size (a 17 × 17 torus). This population size was similar to the recommended population size for this problem from previous work (Mahfoud, 1995c). Other parameters were kept the same as described in §4.2. As can be seen in Figure 4.19, the GBSSEA has little difficulty finding all the desirable solutions to the Foxholes problem; the maximum peak ratio quickly reaches one, and the chi-square-like performance quickly stabilises. In Figure 4.20, the visualisation of the population over time shows a similar trend as before; the environment encourages local adaptation that results in the formation of homogeneous subpopulations.

The GBSSEA easily located the 25 optima present in the fitness landscape of the Foxholes problem. In this case, the gradient was well suited to the problem; the two phenotypic traits of the problem mapped efficiently to the two dimensional population structure. However, the question of scalability to higher dimension problems remains. For example, given the method for gradient generation used so far in this chapter, it is not possible to create a three-parameter gradient in which all combinations of phenotypic traits are produced. In order for all possible phenotypes to be produced, the dimensionality of the population structure must be at least as great as the number of dimensions in the problem space. This too will encounter problems, as the size of the population will need to increase to ensure sufficient
granularity of the gradient. For example, if a three-parameter problem requires a granularity of 0.1 per step in each trait, a population size of $20^3$ is needed. Clearly, this will lead to an exponential increase in population size as the number of dimensions in the fitness landscape increases.

To avoid the rapid increase in population size, one could try to ‘stack’ additional traits onto the two dimensional population structure. For example, given a three-trait phenotype, the gradient generation function could be:

$$g_{r,c} = \{|1 - x_r|, |1 - x_c|, |1 - x_c|\}.$$ 

While such a gradient generation method might not produce all possible combinations of ideal phenotype, it would at least produce all the possible combinations of two phenotypic traits. To test this method, we extended the $M1$ problem to multiple dimensions:

$$M1^n (x) = M1 (x_1) \cdot M1 (x_2) \cdots M1 (x_n).$$

The GBSSEA was then run with a population size of 1024 (a $32 \times 32$ torus) on the problems $M1^1$, $M1^2$ and $M1^3$. The maximum peak ratio performance for the GBSSEA is shown in

---

**Figure 4.20:** Emergence of species over time for the Shekel’s Foxholes problem using a Gradient-Based SSEA.
Figure 4.21: The maximum peak ratio performance of the GBSSEA on the M1 problem with increasing dimensionality of the problem space.
Figure 4.21. As can be seen, the GBSSEA has little problem finding all optimal species for one and two dimensions, however it quickly fails when the third dimension is added.

The \textit{M1}, \textit{M4} and \textit{Foxholes} problems are all real-parameter problems and, as such, have a true phenotype with which to construct the gradient. However, many problems have no natural phenotype; they operate directly on the genotype of an individual to determine its fitness. In order for the GBSSEA to work on these types of problems, a suitable phenotype is needed. One such problem that has no immediately identifiable phenotype is the massively multimodal deceptive problem (\textit{MMDP}), described in Appendix E. Therefore, it is an ideal candidate to test the behaviour of the GBSSEA on artificial phenotypes.

The manufactured phenotype developed for the \textit{MMDP} problem is deliberately simple; it is simply the count of the number of activated bits in the bitstring. Keeping the genotype-phenotype mapping simple allows several key aspects to be analysed:

- \textit{Dimensionality Compression}: Essentially, the \textit{MMDP} problem is a concatenation of five smaller units and can be considered a five-dimension problem. The genotype-phenotype mapping used here converts the five-dimension genotype into a single real-valued value. If the GBSSEA is able to perform well on this problem, then it may suggest a possible solution to the problem of constructing gradients for high-dimension phenotypes.

- \textit{Redundant Encoding}: The naïve implementation of the genotype-phenotype mapping results in several genotypes producing the same phenotype. This means, from the point of view of the problem, that several different solutions will be competing with each other for the same locations in space.

- \textit{Gradient Robustness}: The chosen mapping function can be considered a baseline performance measure. The GBSSEA is being tested under less than optimal conditions and, therefore, we can expect any observed behaviour to be a good measure of the GBSSEA’s performance in real-world conditions. A more intelligent mapping function would only be expected to improve upon this baseline performance.

As the constructed phenotype contains only one trait, we will use the same gradient generation methods as used on \textit{M1} and \textit{M4}.

### 4.4.1 Manufactured phenotypes, deception and dead zones

The GBSSEA was run on the \textit{MMDP} problem using a $26 \times 26$ torus; the population size was sourced from work investigating other niching methods on this problem (Mahfoud, 1995a;
Pétrowski, 1996). The maximum peak ratio and chi-square-like performance for these runs is shown in Figure 4.22. The results show that, on average, the introduction of a gradient actually hinders the performance of an SSEA. Interestingly, although the GBSSEA manages to find fewer optima on average, it appears to be more precise in the allocation of individuals to the optima that it finds.

An interesting observation can also be made about the visualisation of the population over time, as shown in Figure 4.23; while the other problems investigated so far have all produced populations that possess homogeneous bands of species, the phenotype used with the MMDP problem has promoted a large amount of interspecies competition for places in the population. For example, the strings \{111111 000000 000000 000000 000000\} and \{000000 000000 000000 000000 111111\} both have the same number of activated bits, and will therefore express the same phenotype. As they possess the same phenotype, each individual will be at their fittest within the same locations in the population. At first glance, this might seem like a desirable trait, as crossover interactions between optima can sometimes produce a new optimum. However, in this case crossover will almost certainly produce offspring with either a higher or lower number of activated bits. Therefore, the offspring are likely to have a phenotype that is not well suited to the location of their parents, and as such they will not survive in the environment. In effect, any benefit that we get from crossover between optima will be lost.

One final observation can be made about the behaviour of the GBSSEA on the MMDP problem. In Figure 4.23 two bands of sub-optimal individuals exist within the population. The ideal phenotype at these locations equals 15. Obviously, this environment will be best
suited to individuals that possess 15 activated bits in their genotype. However, the nearest optima in the fitness landscape have a phenotype of either 12 or 18. The local fitness of either of these optima at these locations is 4.616, which is in fact lower than that of the deceptive optima with a phenotype of 15. Therefore, the deceptive optima are outcompeting the desirable optima at certain locations in space. This is expected to occur wherever an ideal phenotype that matches a deceptive optima is produced. In other words, this will occur whenever an ideal phenotype of 3, 9, 15, 21 or 27 is encountered. In the population used, only one of these ideal phenotypes was present. However, both smaller and larger population sizes exhibit this phenomenon, as shown in Figure 4.24. This problem can be overcome by using a larger value for $\sigma$, but the motivation for doing so requires *a priori* knowledge of the fitness landscape and is therefore not easily justified as a solution for general purpose use.

### 4.4.2 Alternative gradient approaches

The gradient generation method used in in this chapter attempts to make a smooth gradient that varies consistently throughout space from the minimum to the maximum of the possible range of phenotypic traits. This approach has been ideal to analyse and discover the
Figure 4.24: Emergence of regions in the population that are occupied by deceptive solutions in the MMDP problem. In the most extreme case, every cell in the $10 \times 10$ torus is occupied by a deceptive solution.
behavioural properties of the GBSSEA. However, it possesses several traits that limit its general purpose usefulness, including:

1. **Rigid Granularity**: The gradient generation uses fixed step sizes for determining ideal phenotypes. As seen on the M4 problem, this can cause problems when the occupiable territory for a given optima is close to the boundary of acceptable phenotype values. The only way to resolve this problem is to increase the population size.

2. **Limited Applicability to High-Dimension Problems**: Ideally, the gradient generation method should produce all possible combinations of ideal phenotype traits. To do this, the current gradient generation method requires at least as many dimensions for the population structure as are present in the problem space. This in turn requires population sizes that quickly grow to be unmanageable.

3. **Reduction in Crossover Effectiveness**: One of the greatest benefits of using crossover is that it can potentially use existing optima as stepping stones to discovering new optima. As suggested in §4.4.1, this benefit may be diminished through the use of the current gradient.

One possible way to at least partially overcome all of these problems would be to replace the current gradient generation method with one that randomly generates the ideal phenotype at each location. An example of how this might be applied to a two-dimensional phenotype is shown in Figure 4.25. The resultant rugged gradient would increase the granularity of the gradient; the possible values for ideal phenotype is now limited only by the cycle space of the random number generator. Additionally, it is more likely to produce an even spread
of ideal phenotypes over the entire phenotype space, which should increase its applicability to problems with higher dimensionality. Finally, the ruggedness of the gradient is likely to produce conditions where crossover between optima is promoted, while at the same time producing locations in which the offspring of these crossover interactions will be viable.

The GBSSEA was reapplied to 
\textit{Foxholes} and 
\textit{MMDP}, this time using the new rugged gradient approach. Additionally, the \( M1^n \) problems, as defined in §4.4, were used to measure the ability of the rugged gradient approach to maintain optima in problems in which the number of dimensions in the phenotype exceeded the population dimensions. In all cases, the parameters other than the gradient generator were kept the same.

The performance of the rugged GBSSEA on the 
\textit{Foxholes} and 
\textit{MMDP} problems is shown in Figure 4.26. In both cases, the maximum peak ratio performance exceeded that of the smooth gradient approach. A similar improvement in chi-square-like performance was os-

\hspace{1cm}

\begin{figure}[h]
\centering
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{max_peak_ratio_foxholes}
\caption{Maximum peak ratio (\textit{Foxholes})}
\end{subfigure}
\hfill
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{chi_square_performance_foxholes}
\caption{Chi-square-like performance (\textit{Foxholes})}
\end{subfigure}
\hfill
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{max_peak_ratio_mmdp}
\caption{Maximum peak ratio (\textit{MMDP})}
\end{subfigure}
\hfill
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{chi_square_performance_mmdp}
\caption{Chi-square-like performance (\textit{MMDP})}
\end{subfigure}
\caption{The effect of gradient implementation on maximum peak ratio and chi-square-like performance of the gradient-based SSEA approach.}
\end{figure}
observed, suggesting that the rugged gradient approach improved the ability of the GBSSEA in terms of allocation of individuals to optima. Finally, Figure 4.27 shows the final population configurations of sample runs using a GBSSEA with a rugged gradient. Interestingly, the population for the MMDP appears to have a greater number of suboptimal individuals present in the population than that observed when using the smooth gradient. In both problems, the homogeneous blocks that appear by using a smooth gradient are not present to the same degree. This may, in fact, be beneficial, as it is likely that crossover between the boundaries of these blocks was more successful at creating viable offspring through combining existing optima.

The maximum peak ratio performance of the GBSSEA on the $M1^n$ problem is shown in Figure 4.28. Note that in the $M1^n$ and $M1^6$ problems, there are respectively 3125 and 15625 desired optima in the fitness landscape. In these cases, the number of optima exceeds that of the population size used (1024), so it would be impossible for the GBSSEA to maintain every optimum in the fitness landscape. The y-axis in Figures 4.28(c) and 4.28(d) have been altered to reflect the upper limit on the attainable maximum peak ratio. Although not able to support the maximum number of optima in the population, the rugged gradient approach still manages to substantially outperform both the simple gradient approach and an SSEA with no gradient.

A genotypic gradient

The phenotype for the MMDP problem is deliberately naïve and was intended to demonstrate the behaviour of the GBSSEA under the worst conditions possible. Perhaps its biggest shortcoming was the fact that multiple genotypes mapped to the same phenotype, and this caused contention between species for the same locations. As an alternative implementation,
Figure 4.28: The effect of gradient implementation on maximum peak ratio performance of the GBSSEA on the M1^n. At higher problem dimensions, the rugged gradient implementation is able to maintain more optima within the population than both the simple gradient and SSEA (no gradient) approaches.
one could extend the rugged gradient approach to have an *ideal genotype*; at each location, a randomly generated bitstring could represent the ideal individual. The distance measure would then change to be the Hamming distance between the individual and the location’s bitstring.

This simple modification of the rugged gradient approach was tested against the *MMDP* problem, and the results are shown in Figure 4.29. By using this rugged genotypic gradient, both the maximum peak ratio and chi-square-like performance are improved.

### 4.5 Summary

In this chapter, we have shown that the introduction of an environmental gradient within spatially-structured evolutionary algorithms can promote parapatric speciation-like behaviour suitable for use in multimodal problems. We have seen that, in the case of equal-valued optima and a one-to-one mapping between phenotype and gradient, the GBSSEA exhibits near-optimal performance with regards to niche discovery and allocation of individuals. When the difference in fitness between optima varies greatly, the GBSSEA will favour locally-optimal solutions at the expense of less-fit solutions, ensuring that the solutions returned are of high quality. Additionally, we showed that the allocation of individuals to niches in the GBSSEA does not follow the traditional model of a niching EA. Instead of using global knowledge of the fitness landscape to determine the distribution of individuals, the GBSSEA uses local interactions between optima in the fitness landscape; it identifies equivalence points – the points in a gradient where two optima are considered equal – and uses these points to de-
termine the territory that each niche dominates. In the case of unequal-valued optima in the search space, this can result in markedly different niche allocation behaviour.

The choice of gradient is particularly important in determining the success of the GB-SSEA. Ideally, the implementation of the gradient will: allow for a high granularity in the range of ideal phenotypes; produce combinations of ideal phenotype that thoroughly sample the problem space; cope with high dimensionality of the problem space; limit the competition between niches for similar locations; and encourage local adaptation without unnecessarily impeding the ability to thoroughly search the fitness landscape of the problem. Based on the findings of this chapter, it appears that a rugged gradient implementation, with a randomly generated ideal phenotype for each location, may provide the key for providing an environment that fulfils all these requirements.
Chapter 5

Local Niching I: Local Sharing

The previous chapter presented a gradient-based solution to managing convergence of a population via selection. The proposed solution was able to maintain multiple optima in a fitness landscape. However, it is not the only possible solution to allow SSEAs to maintain diverse populations. The goal of this chapter is to explore the behaviour of local niching methods – SSEAs that have been hybridised with niching methods that are typically applied globally to populations. In particular, the chapter will focus on the behaviour of one particular method that hybridises fitness sharing with SSEAs, dubbed local sharing. The behaviour of this method will be tested on problems that have previously been shown to be difficult to search by using either an SSEA or fitness sharing in isolation. It will be shown that, when appropriately configured, local sharing produces search capabilities that are beyond that of either fitness sharing or SSEAs when they are used separately.

5.1 Local Sharing

The general process for an SSEA, as shown in Algorithm 2.5, constructs a deme at each location in space from which parents are selected. Once this deme is constructed, the remaining steps are largely the same as that for an unstructured EA; parents are selected, then offspring are produced before a set of rules are applied to determine the individual that survives through to the next generation. It seems reasonable then, that any operators that unstructured EAs incorporate prior to selection and replacement could be integrated into SSEAs with the expectation that they would provide similar benefits. Doing so would allow the various niching methods that were discussed in Chapter 2 to be included in the demes of an SSEA, resulting in a family of algorithms that can be best described as local niching methods.
The goal of this chapter is to examine a specific instance of local niching that integrates fitness sharing into the demes of an SSEA. The process for the local sharing method is defined in Algorithms 5.1 and 5.2. As can be seen in Algorithm 5.1, the actual process of sharing within local sharing is almost exactly the same as that for global sharing, with the only difference being that a separate instance of sharing is applied to each deme in the population. This means that the shared fitness of an individual only depends on the interaction with its fellow deme members, not the entire population.

| input | A spatially-structured population. |
| output | The same population with correctly calculated shared fitness of individuals. |

```plaintext
foreach location in space do
  current ← population[location];
  deme ← constructDeme(location);
  m ← 0;
  foreach member ∈ deme do
    d ← dist(current, member);
    m ← m + sh(d);
  end
  current.fitness ← evaluate(current) / m;
end
return population;
```

**Algorithm 5.1: The local sharing method.**

The process of local sharing deviates from that of a typical SSEA once we move to the creation of a new generation, as shown in Algorithm 5.2. Before individuals are selected for mating, the local sharing method calculates the shared fitness of each individual in the population using Algorithm 5.1. After this, selection takes place as for a typical SSEA, only this time the shared fitness of individuals is used. The offspring that are produced are then placed in the next generation as candidates for survival. Once all offspring are created, a second round of Algorithm 5.1 is applied, this time to the candidate generation. Finally, the shared fitness of each offspring is compared to that of the current occupant at the offspring’s location; if the shared fitness of the offspring is greater, then it survives into the next generation. Otherwise, the current individual replaces the offspring and is carried through into the next time step.
input : A given problem

output: A spatially-structured population of evolved candidate solutions to the problem.

1 population ← \{\};
2 foreach location in space do
3    population[location] ← newRandomIndividual();
4 end
5 while not done do
6    localSharing(population, \sigma_{sh});
7    generation ← \{\};
8    foreach location in space do
9        deme ← constructDeme(location);
10       parents ← select(deme);
11       generation[location] ← reproduce(parents);
12    end
13    localSharing(generation, \sigma_{sh});
14    foreach location in space do
15        if population[location].fitness ≥ generation[location].fitness then
16            generation[location] ← population[location];
17        end
18    end
19    population ← generation;
20 end
21 return population;

Algorithm 5.2: The general sequence for a spatially-structured evolutionary algorithm with local sharing.
The implementation of the local sharing method presented here offers two potential benefits over the existing fitness sharing and SSEA methods:

1. **Controlled Elitism**: In a typical SSEA, there is a tendency for the fittest solution to quickly propagate throughout the entire population. However, in the local sharing approach, if the fittest solution starts to spread throughout the population, it will quickly start to decrease its shared fitness due to its increased presence. As elitism in local sharing takes into consideration the shared fitness of an individual, this should give less fit (but equally desired) solutions a chance to survive.

2. **Increased Selection Pressure**: It is well-known that fitness sharing significantly reduces the selection pressure on a population (Mahfoud, 1995b). However, the shared fitness elitist replacement present in the local sharing method should increase the selection pressure placed upon individuals and create a suitably managed selection toward fitter solutions in the fitness landscape.

There is also one additional benefit to the local sharing method that is not behavioural, but rather operational, in nature. For a population size of \(N\), the traditional sharing operator must perform at least \(\binom{N}{2}\) comparisons.\(^1\) In contrast to global sharing, the local sharing method replaces the single global sharing instance with a smaller instance for each deme in the population. As only the individual at the centre of the deme is compared with the other deme members, the number of comparisons performed is linear with respect to the deme size. This means that, for a deme size \(d\), local sharing performs only \(Nd\) comparisons per generation. Given that \(d \ll N\) typically holds for most deme sizes, this represents a considerably lower time complexity for the local sharing approach.

### 5.1.1 Basic local sharing behaviour

The behaviour of the local sharing method was compared to traditional implementations of fitness sharing and an SSEA. Two test functions, from Appendix E, were used to analyse the properties of local sharing: *Foxholes* and *MMDP*. Since SSEAs have difficulty maintaining multiple solutions to *Foxholes*, and fitness sharing fails similarly on *MMDP*, these two problems represent ideal cases with which to test the local sharing approach. The parameters that were used in the previous chapter, and described in Appendix C, were again used here

\(^1\)Goldberg and Richardson (1987) suggest a method for sampling the population to reduce the complexity of the sharing method. However, most implementations of sharing do not take this approach and instead focus on performing all the required distance measures.
Figure 5.1: The performance of the local sharing method on the Foxholes problem. The performance of a standard SSEA and a fitness sharing EA are included as a point of reference.

(where applicable); in the case of Foxholes, a population size of 289 was used, while the MMDP runs used a population size of 676. For the sharing and local sharing methods, the recommended values for $\sigma_{ah}$ were used, namely 8 for Foxholes and 6 for MMDP (Mahfoud, 1995a).

The performance of local sharing on the Foxholes problem is shown in Figure 5.1. The results indicate that the local sharing approach was significantly more capable of finding multiple optima than a simple SSEA on this problem; local sharing identified on average approximately half of the optima in the fitness landscape. The chi-square-like performance was also much better for local sharing. However, the performance of local sharing on Foxholes was significantly worse than that of standard fitness sharing.

The results for the MMDP runs are shown in Figure 5.2. Here, the local sharing approach was clearly superior to both a simple SSEA and fitness sharing alone. Local sharing managed to find nearly all optima in the MMDP fitness landscape. The chi-square-like performance is also much better for local sharing than either of the other two algorithms.

5.1.2 Elitism in local sharing

The performance of the local sharing approach on the MMDP problem was very good, however its performance on Foxholes trailed significantly behind that of standard sharing. The number of optima present in the fitness landscape of the two problems is not significantly different, so the number of optima to maintain should not be a factor. Nor should the difficulty of fitness landscape, as the fitness landscape in MMDP presents deceptive optima, and is therefore harder than Foxholes. However, unlike MMDP, the optima in Foxholes vary
in fitness. The reason for the difference in performance is possibly the elitism strategy in local sharing; while the use of shared fitness during elitism certainly helped local sharing to maintain more solutions than by an SSEA alone, it was not enough to sufficiently control the selection pressure. Therefore, several of the less-fit solutions were lost from the population as they were out-competed by the fitter optima.

If the problem encountered by local sharing on Foxholes was due to uncontrolled elitism, then removal of elitism from the algorithm should significantly increase the number of optima found and maintained. As shown in Figure 5.3, the performance of local sharing on Foxholes was significantly improved by the removal of elitism. However, when elitism was removed, the performance of local sharing on MMDP suffered significantly (as shown in Figure 5.4). Clearly, elitist replacement is a vital component in determining the success of the local sharing method.

Based upon the performance observed here, it is clear that elitism cannot be removed from local sharing without unduly restricting its search capabilities. Therefore, the next logical step would be to test the performance of the local sharing approach when the elitism strategy is replaced with a probabilistic tournament to determine offspring survival. In this approach, offspring enter a tournament with the current occupant to determine who is passed into the next generation. The probability of an offspring \( o \) winning the tournament over the current occupant \( i \) is:

\[
p(o) = \frac{f_{sh}(o)}{f_{sh}(o) + f_{sh}(i)}
\]

where \( f_{sh} \) is the shared fitness of an individual. The results of using this approach on Foxholes are presented in Figure 5.5. The performance of local sharing improved significantly
Figure 5.3: The performance of the local sharing method on the Foxholes problem. Elitist replacement was removed from both spatially-structured EAs.

Figure 5.4: The performance of the local sharing method on the MMDP problem. Elitist replacement was removed from both spatially-structured EAs.
by the introduction of the probabilistic elitism strategy, with almost all optima being maintained in the population. However, as was the case with no elitism, the ability of local sharing to discover multiple optima within the MMDP fitness landscape is significantly hindered by the introduction of probabilistic elitism (as shown in Figure 5.6).

5.1.3 Deme size in local sharing

As predicted earlier, elitism clearly plays a vital role in discovery of optima for the local sharing method. Likewise, elitism in the local sharing method is guided by the feedback it receives from the sharing process that is applied to the demes. However, elitism in local
sharing using the D5 deme does not appear to be fully controlled; if it were, then all the optima in Foxholes would be maintained. This suggests that the small, five-member deme size cannot provide a large enough sample for the sharing operator to adequately ascertain the prevalence of certain optima within the population. Hence, the shared fitnesses of some optima are not being reduced to the level that they should have been, and are therefore dominating the locations in the population.

If the deme size were made larger, then the feedback from sharing might be more accurate and elitism would be more successfully controlled. To test this hypothesis, the experiments using Foxholes and MMDP were repeated using a larger D25 deme size (Sarma, 1998). An example of this configuration of deme was presented earlier in Figure 2.4. The performance of the local sharing method using the larger deme size is shown in Figure 5.7 for the Foxholes problem, while the results for the MMDP problem are presented in Figure 5.8. With the larger deme size, local sharing now has no difficulty maintaining all 25 optima of the Foxholes problem within the population. The increased level of feedback, provided by the larger sample used for sharing, makes it easier for local sharing to perform more accurate and controlled elitism.

The improved performance on Foxholes when using a larger deme size with local sharing does not carry over to the MMDP problem. Here, the performance drops considerably when the larger D25 deme is used. There are two possible explanations for this. The first explanation is that the larger deme size reduces the spatial effect of the population (as demonstrated in Chapter 3), which in turn may have a negative effect on search. As the SSEA showed a
similar drop in performance when using the D25 deme, this is not entirely unlikely. However, a second, more likely scenario may also be true; as the deme size is increased, the emphasis of the algorithm is shifted away from the SSEA component and shifted toward its underlying sharing element. As sharing is very inefficient at searching the \textit{MMDP} problem space, we would expect local sharing to demonstrate progressively poorer performance on this problem as the deme size increases. This is confirmed in Figures 5.9 and 5.10, which show the performance of local sharing and an SSEA on \textit{Foxholes} and \textit{MMDP} when using increasingly larger deme sizes. In these figures, the z-axis measures the maximum peak ratio observed over time (in one axis) against a given deme size (in the other axis). As the deme size in an SSEA increases, the maximum peak ratio performance tapers off slightly but is not greatly affected. This is true for both problems. The \textit{Foxholes} problem poses a difficult challenge for the local sharing approach; with the smallest deme size, the sharing effect within local sharing is minimal and therefore the search behaviour is governed primarily through the population topology. However, as the deme size increases, the performance of local sharing on \textit{Foxholes} improves substantially. As fitness sharing performs well on this type of problem, one can surmise that the effect of increasing the deme size is to place more emphasis on the sharing characteristics of the local sharing method.

The performance of local sharing on the \textit{MMDP} problem portrays a markedly different behaviour to that encountered on the \textit{Foxholes} problem. Here, the performance of local sharing when using a small deme is excellent. However, as the deme size increases, the maximum peak ratio performance of local sharing drops off rapidly. This drop off in perfor-
performance is much greater than that exhibited by a simple SSEA. We know that fitness sharing performs poorly on this type of problem, so the logical conclusion to draw from this is the same as that drawn for the behaviour of local sharing on the Foxholes problem; as the deme size increases, the behaviour of local sharing moves further away from that of an SSEA and instead becomes more closely aligned with a traditional fitness sharing approach.

5.2 A Weighted Local Sharing Approach

The components of local sharing appear somewhat at odds with each other; the SSEA component ideally needs a smaller deme size to maximise the spatial effect promoted by the population topology. Conversely, the sharing component ideally needs a moderate-sized deme in order to accurately manage and prevent takeover within the population. It would appear then, that the two operators – sharing and selection – should operate at two different levels within the local sharing population. This could easily be implemented by having two deme sizes; one used when sharing is to take place, and a second smaller deme size for selection of mates.

While the introduction of a second deme size into local sharing might be a suitable solution, another potential solution would take into account the topological distances between individuals within the population structure. It seems reasonable that the contribution of an individual to another population member’s shared fitness could be weighted according to the level of separation between them in the population. For example, given a distance $d_{ij}$ between two locations in space, the function:

$$W_{sh}(i, j) = \left( \frac{1}{1 + d_{ij}} \right)^2$$

would place a greater emphasis on pairs of individuals that are ‘close’ within the population, while still allowing individuals that are slightly further apart to contribute to the niche count, albeit at a reduced level. A plot of this function is shown in Figure 5.11; as can be seen, the weighting is reasonably strong between close locations and becomes negligible after approximately five steps.

Another set of experiments was performed using the local sharing method, this time using the weighted sharing approach. A D5 deme was used for parent selection, thereby maximising the spatial effect promoted by the population structure. The results of these experiments are shown in Figure 5.12 for the Foxholes problem and Figure 5.13 for the MMDP problem. The performance of local sharing on both problems is significantly improved; in both cases every optima in the fitness landscape is discovered and maintained. Additionally, the
Figure 5.9: The effect of increasing deme size on maximum peak ratio performance of the Foxholes problem.
Figure 5.10: The effect of increasing deme size on maximum peak ratio performance of the MMDP problem.
Figure 5.11: A potential function for determining the weights applied to the sharing function in the Local Sharing SSEA.

Figure 5.12: The performance of the weighted local sharing method on the Foxholes problem.

chi-square-like performance improved, suggesting that the weighted local sharing method is performing well with respect to allocation of individuals to optima.
Figure 5.13: The performance of the weighted local sharing method on the MMDP problem.

5.3 Summary

This chapter has introduced the concept of local niching methods, which are spatially-structured evolutionary algorithms that incorporate traditional niching methods into parent selection and offspring replacement. The concept and properties of local niching were demonstrated through the use of local sharing, which is one implementation of local niching based around the concept of fitness sharing.

Local sharing demonstrates a number of interesting properties that make it a viable candidate for use in real world problems; unlike standard SSEAs, it is able to maintain multiple optima from a fitness landscape in which the desiredoptima vary in fitness. This was demonstrated through the use of the Foxhole problem. Additionally, it is able to overcome the limitations of sharing to solve the MMDP problem. When extended through the use of a weighting system, the local sharing method proved to be a fast and stable multimodal search method.

The weighting model allows for sharing and selection to occur at different scales within the population. Additionally, the weighting approach implicitly creates large deme sizes, as each population member is now compared with every other individual, while still managing to bias the sharing process toward topologically close individuals. Therefore, the tight, small range spatial effects required to emphasise the SSEA component are maintained, while at the same time the sharing component is given a larger sample with which to analyse the current state of the population. The end result is an algorithm that appears to work well on both of the test functions from this chapter, which is something that both fitness sharing and the SSEA cannot claim to achieve when used in isolation.
There is one aspect of local niching that this chapter has not explored; traditional niching methods typically require accurate calibration of one or more parameters used to identify species within the population. As traditional niching operators are typically applied only once per generation, incorrect values for these parameters will make differentiation of close optima difficult. However, the local niching concept does not rely on a single round of niching per generation; rather, it performs numerous rounds of niching using a much smaller subset of the population each time. Therefore, each individual instance of the niching method need not be as accurate, as the different locations in space are likely to concentrate on different optima. This should allow the local niching method to demonstrate a level of parameter robustness not possible when using global niching methods. As the sharing method is already relatively stable to parameter settings (Deb and Goldberg, 1989; Horn, 1997; Watson, 1999), it would be difficult to detect this phenomenon using local sharing. However, in the next chapter, we will introduce another form of local niching – local clearing – based around the clearing niching method, which is known to be particularly sensitive to parameter settings.
Chapter 6

Local Niching II: Local Clearing and Parameter Robustness

The previous chapter introduced the concept of local niching and demonstrated some of the benefits of hybridising niching methods with spatially-structured evolutionary algorithms through the local sharing method. One of the potential benefits of local niching is that it may help reduce the need for precise parameterisation of the underlying niching method in order to operate effectively. However, the fitness sharing method that local sharing is based upon is fairly robust with respect to parameter settings. Clearly, the local sharing method will not help us in testing this theory.

An alternative niching method to fitness sharing is clearing (Pétrowski, 1996). The clearing method is a powerful multimodal search technique, however it has been shown to be particularly sensitive with respect to parameterisation (Li et al., 2002; Parrott and Li, 2006). Like fitness sharing, clearing is able to be hybridised into a local niching method. The goal of this chapter is to test this method, dubbed local clearing, and demonstrate the properties that the local niching approach uses to instill an increased robustness toward parameterisation in the clearing method.

If the traditional clearing approach is incorrectly configured then it will fail to correctly identify all optima within the population. This chapter will demonstrate that, at the level of a deme, local clearing will also fail to identify all species when incorrectly parameterised. However, each location in space will in fact maintain diverse sets of species through stochastic processes and sampling of different regions of the fitness landscape. Therefore, from a global point of view, the likelihood of identifying and preserving all species within a population increases when local clearing is used. Additionally, it will be shown that this global species information can be used to automatically adapt the parameters of the local clearing method to near optimal values during the course of a single run.
6.1 The Local Clearing Method

The process for local clearing is defined in Algorithm 6.1. The clearing invocation on line 11 uses exactly the same implementation as that of Algorithm 2.3; the only difference is that the current deme is passed as a parameter rather than the entire population. Aside from the use of clearing, there is only one point of significant difference between local clearing and a typical SSEA; in the local clearing method, elitism takes place before any offspring are produced. By doing this, the local clearing method reduces the number of evaluations required. Also, by using the information gained from clearing the deme, local clearing can easily decide if this individual should survive or be replaced.

The clearing method is an efficient operation that exhibits low time complexity; if there are \( q \) optima in the fitness landscape, then the complexity of the algorithm is \( O(qN) \) for a given population size \( N \) (Pétrowski, 1996). In the case of local clearing, each instance of clearing is applied to a deme of size \( d \). Therefore, the time complexity for each instance of local clearing is \( O(qd) \). However, there will be \( N \) instances of local clearing performed in each generation, so the overall complexity of local clearing is \( O(qdN) \). Given that, like any SSEA, \( d \ll N \) is typically true, this represents a small increase in time complexity for the local clearing method.

6.1.1 Base performance

The intent of this chapter is not to establish a performance comparison (in terms of function evaluation requirements) between the local clearing method and global clearing. Rather, its purpose is to demonstrate the parameter robustness that is brought about through the use of niching methods within a spatially-structured population. However, the robustness of parameterisation must not come at the expense of excessive function evaluations or a reduction in the ability to search a fitness landscape. Therefore, the local clearing method was tested against four benchmark problems from work investigating the clearing method or its derivatives. Two of these four problems, Foxholes and MMDP, have been encountered previously in this thesis. The remaining problems, Shubert and MINHD, have been used previously to demonstrate the qualities of the clearing approach (Li et al., 2002; Pétrowski, 1997b). All four problems are described in detail in Appendix E. The experiments were run using the following population sizes that were sourced from previous work (Mahfoud, 1995a,c; Pétrowski, 1997b; Li et al., 2002): Foxholes, 289; Shubert, 676; MMDP, 676; and MINHD, 1024. All other parameters that were not specific to clearing were the same as those used in Chapter 4 and are described in Appendix C. Additionally, the following values for
input: A given problem
output: A spatially-structured population of evolved candidate solutions to the problem.

\[
\text{population} \leftarrow \{\};
\]

\begin{algorithm}
\begin{algorithmic}
\State \For {location in space}
\State \hspace{1em} \text{population}[location] \leftarrow \text{newRandomIndividual()};
\EndFor
\While {not done}
\State \hspace{1em} \text{generation} \leftarrow \{\};
\For {location in space}
\State \hspace{2em} \text{current} \leftarrow \text{population}[location];
\State \hspace{2em} \text{deme} \leftarrow \text{constructDeme(location)};
\State \hspace{2em} \text{survivors} \leftarrow \text{clearing(deme, } \sigma_{cl} \text{)};
\State \hspace{2em} \text{if current} \in \text{survivors and } \text{fitness(current)} \geq \text{meanFitness(deme)}
\State \hspace{3em} \text{then}
\State \hspace{4em} \text{generation}[location] \leftarrow \text{current};
\State \hspace{2em} \text{else}
\State \hspace{3em} \text{parents} \leftarrow \text{select(deme)};
\State \hspace{3em} \text{generation}[location] \leftarrow \text{reproduce(parents)};
\State \hspace{3em} \text{evaluate(generation}[location]);
\State \hspace{2em} \EndIf
\EndFor
\State \text{population} \leftarrow \text{generation};
\EndWhile
\State \Return \text{population};
\end{algorithmic}
\end{algorithm}

\textbf{Algorithm 6.1:} The general sequence for a spatially-structured evolutionary algorithm with local clearing.
\( \sigma_{cl} \) were used: Foxholes, 8; Shubert, 0.44; MMDP, 6; and MINHD, 3. All values for \( \sigma_{cl} \) were sourced from previous work and allow for perfect discrimination between niches (Mahfoud, 1995a; Li et al., 2002).

As with the local sharing approach from the previous chapter, the deme size within local clearing plays an important role in controlling the balance between emphasising the SSEA or clearing aspects of the algorithm. In the next section, we will investigate the role of deme size within the local clearing method, however for now, we will settle upon the same D25 deme size used in the previous chapter.

The maximum peak ratio performance of the local clearing method and global clearing on the four test problems is presented in Figure 6.1. Both approaches are able to identify all optima in each problem with little difficulty. The local clearing method appears faster at finding optima in the MMDP problem than global clearing, however there is virtually no difference in performance between the two algorithms on the remaining problems.

### 6.2 Parameter Robustness

The clearing method was specifically chosen for its sensitivity toward the value of \( \sigma_{cl} \). So far, we have claimed that the local clearing method will be more robust with regards to imprecise values of \( \sigma_{cl} \) than standard clearing. The argument is that local clearing performs numerous small instances of clearing per generation, and that each of these runs will tend to focus on different parts of the fitness landscape. This claim will be tested by two methods. First, we will run both standard and local clearing on the four test problems, however this time each problem will be tested against a range of values for \( \sigma_{cl} \). After this, we will examine the behaviour of local clearing on the MMDP problem, and in particular look at the number and range of species maintained at random locations within the population at an arbitrary point in time. These two pieces of information should help either confirm or reject the claim that the inherently parallel and autonomous nature of clearing in local clearing leads to the maintenance of more optima than by using clearing alone.

The four problems were run on the standard and local clearing methods using the values for \( \sigma_{cl} \) detailed in Table 6.1. The results of these runs were then averaged and used to create the plots shown in Figures 6.2 through 6.5. In these plots, the z-axis represents the maximum peak ratio achieved by the given algorithm over time (in one axis) for the specified range of values for \( \sigma_{cl} \) (in the final axis). In each of the four problems, the performance of clearing immediately degrades once the critical value, the point where \( \sigma_{cl} \) fails to identify all optima, is surpassed. The local clearing method, however, consistently maintained more optima than
Figure 6.1: The maximum peak ratio performance of the local clearing approach compared to an EA using simple clearing.
the standard clearing approach. Moreover, on at least two of the four problems, it was able
to maintain nearly all optima within the population even when the value of $\sigma_{cl}$ was larger
than the critical value. Therefore, the benefit of using local niching argued at the beginning
of this chapter, that local niching methods can be more robust toward parameterisation, has
been confirmed.

**Table 6.1:** The range of values used for $\sigma_{cl}$ for each of the four test problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Start</th>
<th>End</th>
<th>Increment</th>
<th>Critical Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foxholes</td>
<td>3.0</td>
<td>30.0</td>
<td>3.0</td>
<td>16.00</td>
</tr>
<tr>
<td>Shubert</td>
<td>0.4</td>
<td>8.0</td>
<td>0.4</td>
<td>0.88</td>
</tr>
<tr>
<td>MMDP</td>
<td>1.0</td>
<td>15.0</td>
<td>1.0</td>
<td>6.00</td>
</tr>
<tr>
<td>MINHD</td>
<td>1.0</td>
<td>12.0</td>
<td>1.0</td>
<td>3.00</td>
</tr>
</tbody>
</table>

The experiments demonstrating the robustness of the local clearing method all used a D25
deme. Like the local sharing method, the deme size plays an important role in controlling the
balance between the SSEA and clearing characteristics; if the deme size is too small, then
the sample size provided to the clearing operator will not be sufficient for it to perform its job
properly. Likewise, if the deme size is too large, the clearing operator will be overemphasised
and parameter sensitivity will once again be a concern. Figure 6.6 shows the relationship
between the maximum peak ratio performance of the local clearing method over time for a
range of deme sizes. In this example, $\sigma_{cl}$ was set to 1.0, which is insufficiently small and
cannot discriminate between all optima in the Shubert problem’s fitness landscape. When
the deme size is kept small, we can see that the incorrect value for $\sigma_{cl}$ is overcome and local
clearing manages to find all optima in the problem with high probability. However, as the
deme size is increased, the errors that result from clearing’s inability to distinguish between
optima become an important factor, and the performance of local clearing begins to drop off.
The choice to limit local clearing to a D25 deme seems sound, and therefore this deme size
will be used for the remainder of the chapter.

With the superior parameter robustness of local clearing confirmed, we must now turn to
finding out why this behaviour occurs. The argument that this chapter has put forward is that
rather than trying to maintain all optima within a population using a single run of clearing,
the local clearing approach relies on numerous instances of clearing in a single generation,
with each instance concentrating on only a few optima at a time. The local clearing method
then relies on the fact that each instance will focus on a slightly different set of optima that,
when pooled together, will contain representatives from all solutions in the fitness landscape.
To test if this is true, the local clearing method was run for 200 generations on the MMDP
Figure 6.2: The effect of $\sigma_{cl}$ of the maximum peak ration performance for standard and local clearing on the Foxholes problem.
Figure 6.3: The effect of $\sigma_{cl}$ of the maximum peak ratio performance for standard and local clearing on the Shubert problem.
Figure 6.4: The effect of $\sigma_{cl}$ of the maximum peak ration performance for standard and local clearing on the MMDP problem.
Figure 6.5: The effect of $\sigma_{cl}$ of the maximum peak ration performance for standard and local clearing on the MINHD problem.
Figure 6.6: The effect of deme size on the ability of the local clearing method to maintain solutions in the presence of an incorrect value for \( \sigma_{cl} \). The Shubert problem was being searched with a value for \( \sigma_{cl} \) of 1.00.

problem. At the end of 200 generations, 20 locations were drawn at random and clearing was applied to the deme centred around each location. Two configurations of the tests were run, one with a value for \( \sigma_{cl} \) of 6, and another when the value was 15.

Table 6.2 shows the individuals that survived the clearing process at each location when using \( \sigma_{cl} = 6 \). Each location maintained around 18 individuals after clearing. More importantly, we can see that no single deme contains all 32 optima. However, when we examine all the rows together, we can see that the local clearing method has managed to maintain all 32 optima within the population.

The previous experiment was performed under ideal configurations; the value for \( \sigma_{cl} \) was six, which enabled perfect discrimination of the optima in the fitness landscape. Table 6.3 shows a similar experiment involving 20 randomly selected locations at the end of 200 generations. This time, however, the value for \( \sigma_{cl} \) was 15, which is an extreme value for this parameter that does not allow for much discrimination between optima to take place. As can be seen, the average number of individuals that survived clearing at each location was much lower than when using the ideal value for \( \sigma_{cl} \) and so, like global clearing, the accuracy of clearing within demes was very poor. However, when we consider all the sampled locations, we once again see that all 32 optima survived the clearing process and were maintained into
Table 6.2: A sample of the species maintained at various locations in the population for the MMDP problem using a value for $\sigma_{cl}$ of 6. Note that although no single deme maintains all 32 optima, collectively every optima in the fitness landscape is being maintained.

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>Count</th>
<th>Species Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>17</td>
<td>5,8,9,10,12,13,14,16,17,18,19,21,23,25,26,27,31</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>18</td>
<td>1,2,4,5,7,8,10,11,14,16,17,19,20,21,23,25,26,28</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>17</td>
<td>2,4,5,6,9,10,14,15,17,18,20,21,22,23,25,26,28</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>18</td>
<td>1,4,5,7,8,10,11,12,16,17,18,19,21,23,25,26,27,31</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>17</td>
<td>2,4,5,7,8,10,12,14,16,18,19,21,22,23,25,27,31</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>20</td>
<td>2,5,6,8,9,10,12,14,15,19,21,22,23,25,26,27,29,30,31,32</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>21</td>
<td>1,2,3,4,5,6,7,8,9,10,11,12,14,15,17,19,22,25,30,31,32</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>16</td>
<td>1,2,3,5,7,11,12,14,15,16,19,20,26,28,29,32</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>17</td>
<td>2,3,5,9,10,12,13,15,16,18,22,23,26,27,29,30,31</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>16</td>
<td>1,2,3,4,5,7,11,13,14,15,18,19,20,21,24,28</td>
</tr>
<tr>
<td>10</td>
<td>17</td>
<td>17</td>
<td>2,3,5,7,9,10,12,13,15,16,18,19,21,24,26,29,31</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>20</td>
<td>2,3,5,7,9,10,12,13,14,16,17,18,19,21,22,26,28,29,30,31</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>19</td>
<td>1,5,6,8,9,13,14,15,17,18,19,20,21,22,26,28,29,30,32</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>18</td>
<td>1,2,3,4,5,6,7,8,11,13,14,16,17,18,21,22,26,28,29</td>
</tr>
<tr>
<td>19</td>
<td>5</td>
<td>18</td>
<td>2,3,4,6,8,10,13,14,15,18,22,23,24,25,27,29,31,32</td>
</tr>
<tr>
<td>22</td>
<td>25</td>
<td>19</td>
<td>1,2,3,5,9,10,11,15,17,19,20,21,22,23,25,26,28,29,30</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>20</td>
<td>2,5,6,7,8,10,11,12,13,16,17,19,21,22,23,25,27,29,30,31</td>
</tr>
<tr>
<td>23</td>
<td>5</td>
<td>16</td>
<td>2,4,5,6,7,8,10,11,14,23,24,28,29,30,31,32</td>
</tr>
<tr>
<td>24</td>
<td>7</td>
<td>18</td>
<td>2,4,5,6,7,8,10,11,14,18,20,21,22,23,24,25,27,30</td>
</tr>
<tr>
<td>24</td>
<td>16</td>
<td>16</td>
<td>3,5,8,13,15,16,17,20,22,23,24,25,26,27,29,30</td>
</tr>
</tbody>
</table>

the next generation. Therefore, the hypothesis that local clearing is fairly robust to parameterisation because each of the small instances of clearing focuses on different sets of optima appears to be valid.

### 6.3 Automatic Identification of $\sigma_{cl}$

An interesting observation can be made pertaining to the behaviour of local clearing under incorrect values for the parameter $\sigma_{cl}$; although the local clearing method sometimes fails to maintain all the optima within a single population, it seems to have little problem locating the optima within the fitness landscape. This is probably best explained by the performance of local clearing on the Foxholes and Shubert problems (as shown in Figures 6.2 and 6.3). In each case, the local clearing method locates most of the optima within the fitness landscape in the early generations, only to slowly let them die out due to the errors made by the clearing operator using the poor choice of $\sigma_{cl}$. Ideally, we should be able to use the evolved information about the location of various identified optima to adapt the value of $\sigma_{cl}$ over time. Doing so may help to prevent the optima that have been identified by the local clearing method from being eliminated from the population.
Table 6.3: A sample of the species maintained at various locations in the population for the MMDP problem using a value for $\sigma_{cl}$ of 15. Note that although no single deme maintains all 32 optima, collectively every optima in the fitness landscape is being maintained.

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>Count</th>
<th>Species Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>3</td>
<td>11,20,29</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>3</td>
<td>7,22,28</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>3</td>
<td>3,13,28</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>3</td>
<td>6,20,25</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>3</td>
<td>1,14,27</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>3</td>
<td>2,11,23</td>
</tr>
<tr>
<td>8</td>
<td>24</td>
<td>2</td>
<td>13,20</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>4</td>
<td>3,12,31,32</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>3</td>
<td>9,18,20</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td>11</td>
<td>25</td>
<td>3</td>
<td>9,20,21</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>3</td>
<td>4,15,30</td>
</tr>
<tr>
<td>13</td>
<td>22</td>
<td>2</td>
<td>3,14</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>3</td>
<td>4,14,24</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>4</td>
<td>5,16,20,25</td>
</tr>
<tr>
<td>19</td>
<td>11</td>
<td>3</td>
<td>10,15,19</td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>4</td>
<td>4,17,22,27</td>
</tr>
<tr>
<td>20</td>
<td>23</td>
<td>3</td>
<td>8,21,28</td>
</tr>
<tr>
<td>22</td>
<td>19</td>
<td>2</td>
<td>19,22</td>
</tr>
<tr>
<td>24</td>
<td>17</td>
<td>2</td>
<td>19,26</td>
</tr>
</tbody>
</table>
Tables 6.2 and 6.3 provide some hints into a possible path that an adaptive local clearing method might take to adapt the value $\sigma_{cl}$ during the course of evolution. We could start by pooling together the individuals that were preserved through elitism and survived into the next generation. We can use these individuals because we know that they passed through the clearing process. Then we can find the smallest non-zero distance between any two of these individuals. This minimum distance forms the basis for the new value of $\sigma_{cl}$; if Euclidean distance measures are taken, then $\sigma_{cl}$ equals half of the minimum distance, otherwise the actual minimum distance is used. This approach is more thoroughly documented in Algorithm 6.2.

The described method for adaptive discovery of $\sigma_{cl}$ should work well for discrete problems such as the MMDP and MINHD problems. However, a slight complication arises with real-parameter problems, such as Foxholes and Shubert; often, solutions to these problems will not converge to the exact location of an optima and instead the individuals will form clusters of good solutions around the true optima in the fitness landscape. The minimum distance calculation for $\sigma_{cl}$ should not consider the distances between individuals that reside on the same optima, as the distances between these individuals is actually much smaller than the actual value that $\sigma_{cl}$ should settle upon. A number of solutions are possible to overcome this problem: we could include an option in the algorithm to ignore any pairs of individuals that are within some distance $\delta$ from each other; alternatively, each individual in the pool of elite survivors could perform a local search to find the true optimum; finally, we could use the hill-valley concept (Jelasity, 1998; Ursem, 1999) to determine if any two individuals reside on the same peak in the fitness landscape. Then we could force the calculation for the new value of $\sigma_{cl}$ to overlook the distances between individuals from the same optima. It is this last option that is used in this chapter.

The proposed adaptive local clearing method was tested on the four benchmark problems. Algorithm 6.2 can also be used with standard clearing, so an equivalent set of runs were performed using an adaptive standard clearing method as a point of comparison. The mean distance between the randomly generated individuals from the initial population served as the initial value for $\sigma_{cl}$, subsequent values for this parameter were established at the end of each generation in accordance with Algorithm 6.2. The maximum peak ratio performance of these runs is shown in Figure 6.7; while the adaptive local clearing approach does not work as well as an adequately parameterised instance of local clearing, it appears to work remarkably well considering the lack of information that it was given about the shape of the fitness landscape of each problem. This is particularly relevant when one examines the performance of the adaptive standard clearing, which performed poorly on all the tested problems.
**Algorithm 6.2:** The general sequence for adapting $\sigma_{cl}$ for clearing using information garnered from the current population.
Figure 6.7: The maximum peak ratio performance of the local clearing approach compared to an EA using simple clearing.
In addition to the maximum peak ratio performance, the adapted value of $\sigma_{cl}$ was recorded and the average value observed at each generation is presented in Figure 6.8. The adaptive local clearing method was able to evolve either exact or near-exact values for the ideal parameter setting for $\sigma_{cl}$ for each of the four problems. This is in stark contrast to the performance of adaptive standard clearing, which exhibited continually worsening performance with regard to identifying the correct value for $\sigma_{cl}$ on MMDP and MINHD, while the adaptation of $\sigma_{cl}$ for the Foxholes and Shubert problems improved but was still considerably far from the ideal value.

Figure 6.8: The evolution of $\sigma_{cl}$ over time for the four test problems.
6.4 Summary

This chapter introduced a second variant of the local niching concept – local clearing. The goal of this chapter was to investigate the ability of local niching methods to overcome poor parameterisation and maintain a reasonable ability to preserve many optima within a single population. We identified the reason for local niching’s robustness; each location in local niching operates largely in isolation, producing a search with high levels of redundancy. Collectively, each location will attempt to manage a subset of the optima in the fitness landscape, rather than attempt to manage every optima within a single instance. It was shown that the set of optima managed by each location varies slightly. Overall, the population is able to preserve a greater number of optima than when using a single instance of niching that considers the entire population.

Despite demonstrating a high level of robustness, it was acknowledged that local niching will still perform best when configured correctly. To this end, the chapter introduced a simple mechanism whereby the ideal parameter for the local clearing method was determined during the course of evolution. On four test problems, the adaptive local clearing method achieved high levels of optima discovery and maintenance, despite being given no \textit{a priori} information pertaining to the nature of the fitness landscapes being explored. This level of performance was unachievable by using the same operators within a standard clearing method. While the performance of adaptive local clearing did not reach that of ideally-parameterised static local clearing, it was certainly at a level that would be more than acceptable for real-world use.
Chapter 7

Conclusion

Searching for diverse sets of solutions via evolutionary methods is a difficult task. Often researchers have used analogies to speciation in natural systems to develop methods for maintaining multiple solutions to a given problem. One such avenue that has been explored is the use of spatially-structured populations. However, the multimodal search capabilities of these methods have previously been poorly understood. The goal of this thesis was to explore the behaviour of spatially-structured evolutionary algorithms as multimodal search methods and develop simple extensions to SSEAs that would allow for robust and reliable discovery of multiple solutions in a multimodal fitness landscape.

Before any study on the nature of multimodal search via SSEAs could take place, it was necessary to review the existing literature pertaining to the SSEAs and niching EAs in general. To this end, Chapter 2 serves as a comprehensive document describing the successes and failures of previous work into multimodal search.

Before this study, the local divergence behaviour of SSEAs was poorly understood. In Chapter 3, the behaviour of genetic drift in SSEAs was investigated in isolation of selection and other genetic operators. Through extensive simulations on one and two-dimensional SSEAs, it was shown that the local divergence properties of a spatially-structured population differed significantly from that of a panmictic population. Perhaps the most interesting discovery from this chapter was that genetic drift, in particular fixation time, in the regular topologies typically encountered in SSEAs was additive and independent of the ploidy of individuals. Following this, it was shown that genetic drift in SSEAs was a special case of the Wright-Fisher model of drift; each transient state could not be represented by an allele count, but must consider the actual arrangement of alleles in the population. An exact model for genetic drift was developed, followed by a useful approximation based upon the concept of random walks to absorbing boundaries. The conclusion drawn from this chapter was
that the nature of genetic drift in SSEAs made them ideal candidates for use in multimodal search. However, the behaviour of selection within SSEAs needed modification if they were to become truly effective multimodal search methods.

Based upon the findings in Chapter 3, an exploration into the use of environmental gradients was performed. This chapter extended previous work exploring multiobjective optimisation and theoretical work into parapatric speciation to develop a unified, gradient based spatially-structured evolutionary algorithm suitable for general purpose multimodal search. Testing the GBSSEA on several benchmark problems produced some interesting observations; unlike traditional niching methods, the GBSSEA uses local interactions between neighbouring optima in the fitness landscape to allocate individuals to peaks. This differed substantially from existing niching methods, which use global information about the fitness landscape for allocation of individuals to niches. Additionally, it was shown that there was an interesting trade-off between optimisation toward the problem at hand and maximisation of diversity through adaptation to the local environments produced in the gradient.

The thesis then examined existing niching methods as an alternative to gradients as a way to control the selection qualities of SSEAs. The concept of local niching – incorporating existing niching methods like fitness sharing into the deme selection process – was explored in Chapters 5 and 6. The local sharing method, introduced in Chapter 5, is one implementation of the local niching concept. Local sharing offers a number of benefits of the basic fitness sharing approach, but perhaps the most important property that it offers over basic fitness sharing is elitist replacement. Elitism in the local sharing method was performed by comparing the shared fitnesses of individuals; this promoted a strong selection pressure toward optima in a fitness landscape, while helping to ensure that no one solution was able to take over the entire population. Also of particular interest in Chapter 5 was the introduction of a weighted local sharing approach; this method allowed the sharing and selection/reproduction components of local sharing to operate at different levels, which in turn substantially improved the search capabilities of the local search method on both of the examined test problems.

Chapter 6 was a further investigation into the local niching concept, this time focusing of parameter robustness and the ability to search for multiple solutions in the absence of suitable parameters. A second local niching algorithm – local clearing – was implemented and tested against four well-known functions from niching EA literature. It was shown that the approach of local niching – to apply a smaller instance of the global niching operator to every deme in the population – creates a high level of redundancy in niche maintenance that is not present when niching methods are applied at a global level. When little or no information
about the fitness landscape is available, poor choices of niching parameters are likely to be made. The local niching method can call upon the redundancy it produces to minimise the errors that the niching method will make. The end result is that the local niching approach is more resilient to parameter choice than its underlying global niching method. Additionally, local niching can actually make use of this redundancy to adapt its parameters during the course of a run. This was put to good use on the four benchmark problems; in all cases, the ideal, or almost-ideal, parameter was identified during the course of evolution.

7.1 Resolved Issues in Niching Using Spatially-Structured Populations

Four issues were raised in Chapter 2 that identified a lack of knowledge in the field of SSEAs with respect to searching multimodal fitness landscapes. Addressing these items formed the backbone of this thesis.

Issue 1: Can genetic drift, as it manifests in SSEAs, be an effective tool in locating multiple optima in a multimodal landscape?

Yes. The nature of genetic drift within spatially-structured populations is an excellent promoter of local divergence of genotypes.

The results presented in Chapter 3 show that genetic drift within a spatially-structured population quickly forms large blocks of homogeneous genotypes within the population. These regions were shown empirically to remain within the population for sustained lengths of time. These empirical findings were then supported by models that showed that the mean time for loss of diversity within a spatially-structured population was significantly longer than the running times typically encountered by traditional evolutionary algorithms.

Issue 2: Is there a class of multimodal problem in which current SSEAs, without modification, can discover multiple optima within a single run?

Yes, so long as the fitness of all desired optima are equal. In the case of unequal fitness of peaks, the SSEA will converge upon a single peak, typically the global optimum.

An experiment was conducted at the end of Chapter 3 involving the execution of an SSEA on two problems with identical genotype construction, but dramatically different phenotypes. Considering the problem with unequal optima, selection within the EA forced the population to converge onto the single fittest solution. A similar trend was observed in Chapters 4 and 5, where the SSEA consistently converged to the single fittest solution in the M4 and Foxholes
problems. When the SSEA was run against problems in which the desired optima were equal, it consistently produced populations containing multiple solutions. However, the difference in fitness between any two optima need only be minuscule before the SSEA starts to fail to maintain multiple optima.

**Issue 3: What are the implications of implementing true parapatric speciation within an SSEA?**

*Parapatric speciation is possible within an SSEA, although there are potential difficulties in producing an environmental gradient that will allow the SSEA to fully explore the fitness landscape.*

Traditional SSEAs only implement half of the parapatric speciation concept in that they restrict mating to topologically close individuals. However, the second facet of parapatric speciation – encouraging local adaptation through selection pressures that favour different phenotypes at varying locations in space – is typically omitted. The results from Chapter 4 show that the actual overheads involved in maintaining an environmental gradient are small; an additional function call per generation for each location in space. The inclusion of a gradient in the evolutionary process can sometimes increase the time required to find good solutions, although this is offset by an increase in the number of diverse solutions that gradient-based SSEAs can maintain within a single population.

Perhaps the greatest limitation of the GBSSEA approach is the difficulty it has in scaling to problems with high dimensionality of the problem space. Suggested paths for future research into parapatric speciation in SSEAs are suggested in §7.3 and these may go some way to improving the scalability of this model.

**Issue 4: Are SSEAs suited to hybridisation with existing niching methods?**

*Yes, the introduction of niching methods that are applied just prior to parent selection and offspring replacement in an SSEA is an excellent approach that increases the applicability of the two approaches to a larger domain of problems.*

Chapters 5 and 6 show that the local niching concept demonstrates a number of properties that make it desirable as a search method for many problems domains; problems that were difficult to solve by an SSEA alone are now solvable and *vice versa*. Additionally, problems that could be solved by an SSEA are more effectively searched when a niching method is incorporated into the intra-deme operations.

Perhaps the most useful property of the local niching concept is its demonstrated improvement in the robustness of parameter settings when compared to a traditional niching method. This has real value as a general purpose search method, as it means that we can
apply the method with a reasonable level of confidence that it will perform well even if we have little or no knowledge of the problem that is being explored.

### 7.2 Recommendations for Real-World Problem Solving

The algorithms developed in this thesis are largely ready for use in real world problem solving. This section serves as a list of recommendations of how each algorithm should be used.

First, it should be noted that the standard spatially-structured EA is very capable at searching for a single global optimum in a multimodal fitness landscape. If one is interested in obtaining only a single solution to a given problem, then they would do well to start by using a simple SSEA with a small deme size. Only when one is interested in obtaining multiple solutions should they look to the methods introduced in this thesis.

The most obvious parameter of interest in SSEAs is the population topology. A “one size fits all” population topology that outperforms all other topologies on every problem would be very difficult to define. Indeed, studies have suggested that ideal population topology is very problem dependent (Bryden et al., 2006). It should be noted however, that the effect of adopting different population topologies is to control the selection pressure within the system. Another way to achieve this is to adjust the ratio between deme size and population size; in other words to alter the deme size. It is this approach that was adopted for use in this thesis. Therefore, it seems a reasonable recommendation for real-world use.

The GBSSEA methods using the simple, smooth gradients are best suited to simple problems with no more than a couple of phenotypic traits. As implemented, smooth gradients are not suitable for high dimensionality problems as they simply cannot sample high-dimension phenotype spaces accurately. Therefore, important regions of the fitness landscape may be omitted from the gradient, and the search would stagnate. Recommendations on how this might be overcome are discussed in the Future Work section.

The real utility from the environmental gradient methods will most likely be found with the rugged gradient approach, as it has a tendency to promote useful crossover interactions and has fewer problems scaling to higher dimensionality problems. Evolution with the rugged gradient tends to occur at a much slower rate, so careful attention must be paid to the value of $\sigma$ for the response function. The following usage pattern may be of use:

1. Start with a large (e.g. 1) value for $\sigma$ and run the GBSSEA with a rugged gradient. The large value of $\sigma$ will speed up the rate of evolution, and the GBSSEA should still be able to detect the global optima in the fitness landscape.
2. Some local optima of reasonable fitness will also be returned with a large value of $\sigma$. If more optima are required, then a geometric sequence for $\sigma$ (i.e. $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$) could be used to avoid an excessive number of runs.

It is important that the utility of the genotypic-based rugged gradient is not overlooked; it should be excellent for use as a search method for global optima. Additionally, the overheads required to compute local fitness are minimal and would not impact on the computational time of any real-world use to a significant degree.

It was shown in Chapter 5 that local niching methods offer a level of parameter robustness that is not present in all panmictic niching methods. Therefore, for real-world use they should require less investigation into a suitable value for the niching radius. The niche estimation method proposed by Deb and Goldberg (1989), which has previously been criticised as being inadequate for panmictic niching, may provide a simple and reliable method for local niching parametrisation.

The weighted local sharing approach should be used over the basic local sharing method unless the distance measure between individuals is the dominating feature in terms of computation time. The distance weights can be cached, which greatly improves the runtime performance of the algorithm with only a small expense of some memory to store the lookup table.

Although the local clearing method improves the parameter robustness of the clearing method, it is not perfect and will still fail to maintain all optima in many fitness landscapes when $\sigma_{cl}$ is poorly calibrated. Therefore, the best recommendation that can be made is that either the weighted local sharing or adaptive local clearing methods are used over the basic local clearing method.

Finally, although only briefly examined in Chapter 3, the role of ploidy and heterozygote advantage within SSEAs should be mentioned. Heterozygote advantage could be a useful method for maintaining diversity within a panmictic population (Goldberg and Smith, 1987). However, the high levels of inbreeding in SSEAs tend to rapidly remove most intermediate genotypes from the population. Therefore, the usefulness of heterozygote advantage within spatially-structured populations must be questioned; until further research clarifies this matter, it is suggested that other diversity-maintaining mechanisms be explored for use in SSEAs.
7.2.1 General parameter settings

The spatially-structured EAs developed in this thesis share a number of parameters with that of simple EAs. It was not possible to examine the vast combinations of possible parameter settings within the confines of this thesis. There have been a number of studies into finding suitable values for these parameters, and it is still receives plenty of attention from evolutionary computation researchers (Lobo, Lima and Michalewicz, 2007). Most of this research is directly applicable to the methods developed in this thesis. Additionally, some of the more significant parameters for evolutionary algorithms in the context of SSEAs are worth commenting on:

- **Population size** - The correct population size to use is very problem dependent; several attempts at providing good rules for sizing panmictic populations have appeared in previous work. Given the level of redundant search in SSEAs (as demonstrated in Chapter 6), it is reasonable to assume that an SEA will need a larger population to solve a given problem than that needed by panmictic EAs. Although more research is needed to quantify this properly, a reasonable guide for sizing spatially-structured populations is to take the estimated panmictic population size and add 50%.

- **Selection** - In practice, the selection method used within demes to pick parents matters little to the overall performance of the SEA. This thesis used roulette wheel selection in all experiments to remain consistent with previous work. However, elitist replacement methods present a far more important selection pressure on the population than parent selection. Therefore, one suggestion is to forgo parent selection, pick parents randomly from within the deme, and rely on the selection pressure that emerges via elitist replacement to drive the population toward interesting regions of the fitness landscape.

- **Crossover** - The crossover rate in many EAs is set to less than one to ensure that some good individuals survive unmodified into the next generation. As the elitist strategies in SSEAs ensure the survival of good solutions, there is no reason to use anything other than 100% crossover rates.

- **Mutation** - Mutation is typically applied in crossover-oriented EAs with low probability; its primary role is to reintroduce alleles that have been lost from the population. Mutation rates that are too high can be disruptive to the search process. However, given the combination of elitist replacement and high levels of inbreeding in SSEAs, it is often useful to adopt slightly higher mutation rates to promote greater local diversity.
7.3 Future Work

There are a number of interesting research paths that could be taken to extend the ideas presented in this thesis. This section highlights a few of these possible research directions.

7.3.1 Population topologies

As mentioned in §7.2, the thesis only explored regular topologies for the population structure; in particular the majority of experiments were conducted using a toroidal population structure. Other researchers have observed interesting and useful behaviours on various problems when using alternative population structures (Bryden et al., 2006; Kirley and Stewart, 2007a,b). It would therefore be prudent to explore these topologies to see what, if any, benefit these population topologies present in the context of multimodal optimisation.

7.3.2 Gradient-based SSEAs

Perhaps the most fruitful paths in future research into the GBSSEA would be further investigation into gradient generation methods. Currently, the GBSSEA uses a very simple method of gradient generation that does not scale well to higher dimensions. One path might be to investigate the compressing of multiple dimensions in the parameter space into one dimension for the gradient. For example, two phenotypic traits $x$ and $y$ might each have the range $0, 1, 2, 3$. These two traits could be combined into one through the function $4y + x$. This would effectively reduce the number of phenotype traits, with respect to the environment, and make the search of higher dimensionality problems via a GBSSEA easier to implement.

Another possible path to investigate would be the generation of gradients for high dimension phenotypes using vector quantisation methods such as a self organising map (Kohonen, 1995).

Finally, there are several opportunities for the GBSSEA approach to feed back into the ecological and population genetics models from which it draws inspiration. For example, the GBSSEA could be used as a general framework for ecological models investigating speciation along environmental gradients; the only change needed would be to exchange the problems investigated in the evolutionary computation domain with problems specifically constructed to examine ecological dynamics.
7.3.3 Local niching

The local sharing methods tested in this thesis have only considered the canonical form of sharing as implemented by Goldberg and Richardson (1987). However, other forms of sharing, for example resource sharing, also exist. Logically, the next step would be to implement these other types of sharing into the local niching framework.

The local clearing approach demonstrated a particularly interesting and useful ability to automatically discover the ideal value for the niching parameter, $\sigma_{cl}$. However, the investigation into this problem would benefit from further research. For example, this thesis examined the use of the hill-valley method for determining if two individuals reside within the same niche; future work should explore the other methods, or indeed alternative methods, that might help to correctly identify the correct niching parameters during the evolutionary process.

7.3.4 General research

The test suite for this thesis was limited to six benchmark functions from the niching EA literature. These six were picked as they suitably highlighted the behavioural characteristics of the algorithms that we were exploring. Having performed the basic analysis of the algorithms created in this thesis, it would be prudent for future work to apply these new methods to large, real-world problems to help fully understand their nature as general purpose multimodal search operators. One possibility would be to continue the work of Perez et al. (2003) and investigate the applicability of the new methods on difficult job scheduling problems.
References


Appendix A

Basic Elements of Evolution

Evolution is the process by which organisms change over time, primarily in response to a given environment. Evolution works on populations of individuals (Futuyma, 1998). Each individual has a genome which contains all its genetic material. The genome of an individual is contained within chromosomes and is divided into smaller units called genes. Each gene has a locus (pl. loci), which is its specific position in the genome. A locus can potentially take on many different values; the set of unique values that appear at the locus are the alleles for that specific gene.

Sometimes individuals have only one set of chromosomes, in which case they are haploids. Often, individuals will have many sets of chromosomes; the number of chromosome sets in the genome determines the organism’s ploidy. In diploid organisms, the genome consists of two sets of chromosomes and each set contains half the genes for each loci in the genome. Other organisms, such as some plant hybrids, have multiple sets of chromosomes and are referred to as polyploids.

The combination of alleles present in an organism’s genome represents the genotype of the individual. The physical manifestation of the individual’s genes is its phenotype. The distinction between genotype and phenotype is important as in some cases different alleles may be functionally identical and hence many genotypes may produce the same phenotype.

A population exists within a given environment. This environment will present many challenges to the individuals of the population. The various phenotypes in the population present different “solutions” to these challenges. Some phenotypes will be better suited to the environment and individuals with these phenotypes are of greater fitness.
A.1 Mechanisms of Evolution

There are a number of ways in which the frequency of alleles within a population fluctuate over time, including:

- **Mutation** – Mutations are changes in DNA sequence of an organism that result in new genetic material.

- **Migration** – This is the introduction of genetic material into the population from an external source. Occasionally, the incoming material is not currently in the population, and so this has a similar effect to that of mutation.

- **Selection** – Increases the frequency of alleles in response to some stimulus from the environment. Selection favours the alleles that produce traits which helps an organism to survive in its environment.

- **Genetic Drift** – Alters the frequency of alleles in a population through stochastic events. Unlike selection, drift is direction-less, meaning that changes in allele frequencies are not down to any response to a given environment.

A.2 Speciation

A single term to describe exactly the concept of species is difficult to define. However, the processes in which species emerge, *speciation*, are somewhat easier to define (Mayr, 1970). There are three main forms of speciation (Futuyma, 1998):

- **Allopatric speciation** is the separation of a group of individuals from the larger population. The changes in allele frequencies in this group are now independent from the original population. Given enough time, deme and population may have diverged sufficiently that they become reproductively incompatible. Allopatric literally means “other place”, so this implies a geographic separation of demes.

- **Sympatric speciation** is the mode by which a population diverges into separate species in the absence of geographical separation. Instead, the separation may be seasonal (a part of the population chooses to mate at a different time of the year), or through specialisation to a particular resource (for example, the relationship between insects and plants). Another example is autopolyplody, where a plant may make a ‘mistake’ during the process of reproduction and double the normal number of chromosomes.
in its genome. Breeding between the normal genotype and the ‘accidental’ genotype creates sterile hybrids. Therefore, the ‘mistake’ plants are now reproductively isolated from the original population.

- A third model, *parapatric speciation* occurs when a population is spread out over a large area. Breeding is confined to local individuals, so there is not complete reproductive isolation. Each localised area will have different environmental conditions, so selection will act on different traits in different areas. Eventually, two distinct populations will have diverged sufficiently to the point where they are different species, and between them will be a hybrid zone.

**References**


Appendix B

Models of Selection and Drift

B.1 Selection

Selection has been modelled based on growth of particular allele frequencies. Selection does not operate directly on the genotypes of individuals; rather it works on the resultant phenotype. Usually, each phenotype in the population is given a fitness in direct comparison with the other phenotypes in the population (its viability). This viability, often denoted as \( w \), is the relative measure of the number of offspring an individual is expected to produce within the next generation. Given a population of single locus haploid individuals with two possible alleles of frequency \( p \) and \( q = (1 - p) \), the expected allele frequency after selection is:

\[
p' = \frac{pw_p}{pw_p + qw_q}.
\]  \hspace{1cm} (B.1)

For diploids, there are three possible genotypes, \( A_1A_1, A_1A_2 \) and \( A_2A_2 \). Given a frequency of allele \( A_1 \) of \( p \), the expected proportion of \( A_1 \) alleles after selection becomes:

\[
p' = \frac{p^2w_{11} + pqw_{12}}{p^2w_{11} + 2pqw_{12} + q^2w_{22}}.
\]  \hspace{1cm} (B.2)

B.2 Random Genetic Drift in Panmictic Populations

Natural selection works by increasing the frequency of advantageous alleles within a population. Sometimes, certain alleles produce traits in an individual which offer no selective advantage. In terms of natural selection, one would expect the frequency of such alleles to remain constant within a population. However, stochastic sampling of mates in a population introduces fluctuations in allele frequencies that, in time, can have a significant impact
Figure B.1: *Three examples of change in allele frequency via genetic drift. Unlike selection, random genetic drift evolves the population in no specific direction.*

Genetic drift does not drive the frequency of alleles in any particular direction. It is impossible to determine the outcome of successive generations of genetic drift within a population. Figure B.1 shows this graphically. In this figure, three independent simulations of genetic drift in a finite population are shown. Each run starts with two alleles in equal proportions. In each case, the outcome after 100 generations of random mating is different. In the first case, one allele has been completely removed from the population. Another run produces the exact opposite result; the allele that became extinct in the previous run has now come to occupy the entire population. Finally, a third run shows another possible scenario; the distribution of alleles is largely unchanged after 100 generations of random mating. Eventually, this third run would converge upon a single allele value, but the exact time and specific allele to which to will converge to is impossible to determine.
The Wright-Fisher Model of Genetic Drift

The effect of genetic drift of the eventual loss of an allele in a population. A model for genetic drift is therefore necessary in order to develop a more complete understanding of evolution as a whole.

A simple model for genetic drift was developed implicitly by Fisher (1930) and later explicitly by Wright (1931). This model made several small assumptions about random mating within a population:

1. The number of individuals (and hence alleles) in the population is finite and remains constant over time.
2. No new alleles are introduced into the population.
3. Breeding generations do not overlap.

The model for genetic drift is based around a population of size $N$ with $i$ copies of an allele of interest. In each generation, $N$ individuals are chosen (with replacement) from the population to breed. The probability that this new generation will have $j$ copies of the particular allele of interest follows a binomial distribution and is given by the equation:

$$p_{ij} = \binom{N}{j} \left( \frac{i}{N} \right)^j \left( \frac{N-i}{N} \right)^{N-j}.$$  \hspace{1cm} \text{(B.3)}

If, in the process of breeding, the new generation ends up with $j = 0$ or $j = N$ alleles, evolution via genetic drift stops. This is because all diversity within the population is removed and hence no fluctuation in gene frequencies can occur.

The Wright-Fisher model does not provide any insights into which of the two alleles in the system will survive and which will become extinct. It does, however, provide enough information with which to determine the number of generations that will occur before one of the alleles is lost. This is possible when the process of breeding subsequent generations is modelled as a Markov process (Taylor and Karlin, 1998). Each state in the Markov chain represents the number of copies of the chosen allele in the population, labelled $j = \{0..N\}$. The states $j = 0$ and $j = N$ are absorbing; once the population arrives at either of these states it is trapped and can no longer move to another state. All the possible population states are reachable from the remaining transient states. As an example, a transition matrix
for the Markov chain when \( N = 4 \) and initial allele frequencies are equal is:

\[
P = \begin{bmatrix}
1.0000000 & 0.0000000 & 0.0000000 & 0.0000000 & 0.0000000 \\
0.31640625 & 0.4218750 & 0.2109375 & 0.0468750 & 0.00390625 \\
0.0625000 & 0.2500000 & 0.3750000 & 0.2500000 & 0.0625000 \\
0.00390625 & 0.0468750 & 0.2109375 & 0.4218750 & 0.31640625 \\
0.0000000 & 0.0000000 & 0.0000000 & 0.0000000 & 1.0000000
\end{bmatrix}
\]

The transition matrix can be used to determine the number of generations to population fixation. First, it must be rearranged so that the absorbing states occupy the bottom right corner of the matrix. The transition matrix will then have the form:

\[
P = \begin{bmatrix}
Q & R \\
0 & I
\end{bmatrix}
\]

where \( 0 \) is a \( 2 \times (N - 1) \) matrix consisting entirely of zeros and \( I \) is a \( 2 \times 2 \) identity matrix. \( Q \) is an \( (N - 1) \times (N - 1) \) matrix consisting of the transition probabilities between transient states and \( R \) is an \( (N - 1) \times 2 \) matrix containing the transition probabilities from transient to absorbing states.

The matrix \( Q \) is of particular interest, as it can be used to determine the expected number of steps between transient states. The fundamental matrix of \( Q \) gives us the expected number of visits to a state \( j \) from a given start state of \( i \). The fundamental matrix, \( W \), is calculated via the operation

\[
W = (I - Q)^{-1}
\]

where \( I \) is an identity matrix of the same size as \( Q \). The mean time to absorption given a particular starting state \( i \) is found by calculating the sum of the \( i \)th row of \( W \):

\[
E_i = \sum_{j=1}^{N-1} W_{ij}.
\]

The rearranged transition matrix for the example population with \( N = 4 \) individuals becomes:

\[
P = \begin{bmatrix}
0.4218750 & 0.2109375 & 0.0468750 & 0.31640625 & 0.00390625 \\
0.2500000 & 0.3750000 & 0.2500000 & 0.0625000 & 0.0625000 \\
0.0468750 & 0.2109375 & 0.4218750 & 0.00390625 & 0.31640625 \\
0.0000000 & 0.0000000 & 0.0000000 & 1.0000000 & 0.0000000 \\
0.0000000 & 0.0000000 & 0.0000000 & 0.0000000 & 1.0000000
\end{bmatrix}
\]
and from here the fundamental matrix is computed as:

\[
W = \begin{bmatrix}
2.17931034 & 0.93103448 & 0.57931034 \\
1.10344828 & 2.34482759 & 1.10344828 \\
0.57931034 & 0.93103448 & 2.17931034
\end{bmatrix}.
\]

The expected time to fixation under the Wright-Fisher model of genetic drift is trivial to establish once the fundamental matrix is constructed. Each row \( i \) = \{1..(\( N - 1 \))\} in \( W \) represents a population with \( i \) copies of a given allele. Each row therefore represents \( \binom{N}{i} \) possible populations. The probability of encountering such a population given a starting allele frequency \( p \) is determined by the binomial distribution, so the mean time to fixation of the population, \( \tau \), is:

\[
\tau = \sum_{i=1}^{N-1} \binom{N}{i} p^i (1-p)^{N-i} E_i.
\]  
(B.6)

Figure B.2 shows the match between the Wright-Fisher model and data acquired through simulation of 10000 runs of genetic drift. As can be seen, the Wright-Fisher model can be used to compute an exact value for the expected number of generations until diversity loss occurs via genetic drift.

**Diffusion Approximations to the Wright-Fisher Model**

The Wright-Fisher model gives an exact measure of the mean time for a population to lose diversity. However, the transition matrix grows quickly and soon becomes difficult to manage. Kimura drew inspiration from previous researchers to produce a model for genetic drift based on diffusion equations (Kimura, 1964; Kimura and Ohta, 1969). Given a reasonable population size, the Wright-Fisher model above can be modelled via the equation:

\[
\tau \approx -2N(p \log_e(p) + (1-p) \log_e(1-p))
\]  
(B.7)

where \( N \) is the population size and \( p \) is the initial allele frequency.

Diffusion models also offer other advantages over directly solving the Markov process. One of the main limitations of solving the Markov chain is that it combines the two events of allele fixation and allele extinction. Ideally, a model would separate these two events and attempt to capture a more complete behaviour of genetic drift. Subsequent models proposed by Kimura and Ohta (1969) succeeded in developing a model that accurately computes the time to fixation of an allele and the time to extinction as two separate processes. For example, given a population of size \( N \), with the ploidy level of individuals \( P \) and an allele present
Figure B.2: The Wright-Fisher model of genetic drift as compared to experimental data.
in the population with frequency $p$, the expected time for this allele to disappear from the population is given by:

$$\tau_e = -\frac{2NPp \log_e p}{1-p}.$$  \hfill (B.8)

Likewise, the time required for the allele to take over the population is given by:

$$\tau_f = -\frac{2NP(1-p) \log_e (1-p)}{p}.$$  \hfill (B.9)

**References**


Appendix C

Experimental Framework

Any study into evolutionary algorithms will inevitably require some sort of empirical analysis or comparison. The models developed in this thesis were compared against several existing approaches. To assist in reproducing results, this appendix outlines the common methods and parameter settings used throughout all experiments used in the thesis.

Unless otherwise stated in the text, each experiment was repeated 100 times using a randomly generated initial state. Any reported values are the means calculated from these 100 samples, and are presented with 95% confidence intervals for the true mean. Error bars on all plots are also 95% confidence intervals of the true mean for the plotted variable.

C.1 Parameter settings

Population initialisation - The initial population in each experiment was created by sampling from a uniform random distribution.

Population sizes - Unless for the purposes of demonstrating specific behaviour, population sizes used for specific test problems were as near as possible to those used in previous studies. In some experiments, the population size differed slightly due to the restrictions placed on population sizing by adopting a toroidal population structure. In such cases, the population size used by all algorithms was rounded to the nearest perfect square.

Problem encoding - All individuals were processed as bitstrings. For real-parameter problems, a simple binary-to-decimal mapping was used; Gray coding (Caruana and Schaffer, 1989) was not used as it has been shown not to improve the search capabilities of niching EAs (Watson, 1999). The number of bits used by each problem to encode variables is spec-
ified in Appendix E; typically this was 30 bits per variable. Tight coding (where all the bits used to encode a variable are located contiguously in the genome) was used throughout the thesis, as opposed to a loose (i.e. random) approach (Goldberg, Deb and Korb, 1990).

**Recombination** - For problems where fitness is calculated directly via the bitstring, a simple one-point crossover operation was used. The crossover operation ensured that offspring received at least one bit from each parent by limiting crossover points to be internal to the bitstring (i.e. crossover points were selected from the range \([2 : L - 1]\) for bitstrings of length \(L\)). For real-parameter problems, two-point crossover was adopted, with the same criteria used for sampling crossover points as for one-point crossover. In all problems, crossover was applied with 100% probability. The use of one-point crossover for bitstring problems, and two-point for real-parameter problems, followed that of previous work (Mahfoud, 1995; Watson, 1999).

For experiments using panmictic populations, crossover produced two offspring per operation. However, the spatially-structured populations used in this thesis supported only one individual per location. In these cases, crossover produced a single offspring, and the order in which parents were selected determined the which parts of each parent were copied into the child.

**Mutation** - Mutation was applied to offspring during the crossover operation. In both bitstring and real-parameter problems, mutation was implemented as bit-flipping and applied with a probability of 0.002. A mutation clock (a method whereby the time for the next occurrence of mutation is drawn from an exponential distribution) was used to reduce the computational overhead of mutation (Goldberg, 1989).

**Selection** - All experiments used fitness-proportionate selection (see Appendix B). In keeping with previous studies, stochastic uniform selection (Baker, 1987) was used for panmictic populations, as it presents the most stable form of fitness-proportionate selection (Mahfoud, 1995; Pétrowski, 1996; Watson, 1999). In all spatially-structured populations, the faster and simpler to implement roulette-wheel method of selection was used. Selection within demes was done without replacement, ensuring that no individual could be selected as both parents for a given offspring.

**Random number generator** - The Mersenne twister pseudo-random number generator was used throughout this thesis (Matsumoto and Nishimura, 1998). This method has been shown to be faster than a number of other methods for generating random numbers, while at the same time possessing a high level of periodicity.
References


Appendix D

Performance Measures

The performance measures typically used for simple EAs (for example, average best individual fitness) do not provide the necessary feedback with which to measure the quality of a niching EA. Therefore, we need alternative measures that take into consideration the number, location and relative fitness of the optima present in the fitness landscape if we wish to make statements about the relative performance of niching EAs.

D.1 Species Assignment

In order to ascertain the performance of niching methods, we must first define a method by which individuals are assigned to optima (species) within the fitness landscape. This thesis adopts the method presented in previous studies of niching methods (Deb and Goldberg, 1989; Miller and Shaw, 1996). In particular, an individual is considered to be representative of a species if the following criteria are met:

1. For bitstring problems, an individual must match the optimal string exactly. In other words, the Hamming distance between the string representing the species and the current individual must be zero.

2. For real-parameter problems, the individual must reside within the basin of attraction of the given optimum.

3. Once inside the basin of attraction, an individual must be at least 80% as fit as the local optimum value.
D.2 Niche Discovery Performance – Maximum Peak Ratio

The maximum peak ratio was introduced by Miller and Shaw (1996) to measure not only the number of optima that have been located by the niching EA, but also the distance of individuals from the true optima in the fitness landscape. The maximum peak ratio measure starts by assigning individuals to the correct optimum; an individual is marked as belonging to a given optimum if it is inside the basin of attraction for the optimum and its fitness is at least 80% of the fitness for the optimum. Then, the fittest individual from each optimum is taken as the representative solution for that peak; if no such individual exists then the representative fitness is set to zero. Finally, the maximum peak ratio is defined as the sum of the representative fitnesses from each optimum divided by the sum of the fitness of all optima in the fitness landscape. Therefore, the maximum peak ratio statistic varies from zero to one, with zero indicating that no optima have been discovered while a maximum peak ratio of one suggests that the population has found the exact location of all optima in the fitness landscape.

D.3 Expected Niche Proportions – Chi-Square-Like Performance

Deb and Goldberg (1989) introduced the notion that a niching EA will ideally allocate individuals to the optima in a fitness landscape in proportion to the relative sizes of the peaks. The chi-square-like performance metric was introduced to measure the deviation of a given population from the ideal population; it is defined as the disparity between the observed allocation of individuals to the optima in the fitness landscape and the “perfect” allocation of individuals to peaks in proportion to their value. Under perfect conditions, we would expect a peak $i$ to be allocated $\mu_i$ individuals, which is calculated by:

$$\mu_i = N \frac{\text{fitness}(i)}{\sum_{j=1}^{q} \text{fitness}(j)} \quad (D.1)$$

where $N$ is the size of the population and $q$ is the number of desired optima in the fitness landscape. Similarly, we would expect the variance of the allocation of individuals for a given peak to be:

$$\sigma_i^2 = \mu_i \left(1 - \frac{\mu_i}{N}\right) \quad (D.2)$$

In addition to the estimates of the number of individuals for each optimum, we need an estimate of how many individuals we should expect to appear outside of any optimum.
Obviously, in a perfect population, every individual will be an optimum, so the expected proportion for the ‘non-peak’ individuals, $\mu_{np}$ equals 0. The variance on this statistic was determined by Deb and Goldberg (1989) to be:

$$\sigma_{np}^2 = \sum_{i=1}^{q} \sigma_i^2$$  \hspace{1cm} (D.3)

An example of the expected proportion and variance for the $M1$ and $M4$ problems (described in Appendix E) is presented in Tables D.1 and D.2

**Table D.1: The expected distribution of individuals over the fitness landscape of M1.**

<table>
<thead>
<tr>
<th>Peak</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$i$</td>
</tr>
<tr>
<td>1</td>
<td>0.100</td>
</tr>
<tr>
<td>2</td>
<td>0.100</td>
</tr>
<tr>
<td>3</td>
<td>0.100</td>
</tr>
<tr>
<td>4</td>
<td>0.100</td>
</tr>
<tr>
<td>5</td>
<td>0.100</td>
</tr>
<tr>
<td>Non-peak</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table D.2: The expected distribution of individuals over the fitness landscape of M4.**

<table>
<thead>
<tr>
<th>Peak</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$i$</td>
</tr>
<tr>
<td>1</td>
<td>0.080</td>
</tr>
<tr>
<td>2</td>
<td>0.247</td>
</tr>
<tr>
<td>3</td>
<td>0.451</td>
</tr>
<tr>
<td>4</td>
<td>0.681</td>
</tr>
<tr>
<td>5</td>
<td>0.934</td>
</tr>
<tr>
<td>Non-peak</td>
<td>-</td>
</tr>
</tbody>
</table>

After the determination of the expected number of individuals at each peak, the chi-square-like is defined as:

$$CSL = \sqrt{\left(\frac{X_{np}}{\sigma_{np}}\right)^2 + \sum_{i=1}^{q} \left(\frac{X_i - \mu_i}{\sigma_i}\right)^2}$$  \hspace{1cm} (D.4)

where $X_i$ is the actual number of individuals observed for a given optimum. In this study, we follow Miller and Shaw’s (1996) work and assume that an individual is a representative of an optimum if it lies within the basin of attraction and has a fitness of at least 80% of the optimum’s value.

165
Figure D.1: Possible scenarios for species discovery and maintenance.

D.4 Species Visualisation over Time

On several occasions, the thesis presents graphical representations of the distribution of species within a population over time. These measures provide a quick and powerful mechanism for describing the discovery of optima over time, and also the ability of a niching method to preserve optima once they have been found. In any given visualisation, each optimum in the fitness landscape is assigned a unique colour. Individuals that match the species-identification criteria, as defined in §D.1, are then assigned the appropriate colour. A white cell indicates an individual who is not in the basin of attraction for a given optimum, or has not attained a sufficiently high fitness to be considered as a representative of the species. Figure D.1 presents the typical scenarios encountered for population visualisation, from the complete absence of any species to near-complete occupation by one or more species.

References


Appendix E

Test Functions

The test functions used in this thesis are described below. For each function, we document the number of optima present in the fitness landscape, along with their locations and relative fitnesses.
E.1 Sine Functions: $M1$ and $M4$

The two sinusoidal problems, $M1$ and $M4$ each contain five optima (Beasley et al., 1993)

\[
M1(x) = \sin^6(5\pi x) \\
M4(x) = 2^{-2\left(\frac{x - 0.08}{\pi \times 0.08}\right)^2} \sin^6(5\pi \left[x^{0.75} - 0.05\right]) .
\]  

(E.1)

where $0 \leq x \leq 1$. The real-parameter $x$ was encoded into a 30-bit string for crossover and mutation. The two sine functions, as used in this thesis, are shown in Figure E.1.
E.2 Shekel’s Foxholes

The Shekel’s Foxholes problem contains 25 optima of varying height (Mahfoud, 1995),

$$\text{Foxholes}(x, y) = 500 - \frac{1}{0.002 + \sum_{i=0}^{24} \frac{1}{1+\left(\frac{x-a(i)}{b(i)}\right)^{1.1} + \left(\frac{y-b(i)}{b(i)}\right)^{1.1}}} \quad (E.2)$$

where both $x$ and $y$ are constrained to the range $[-65.536 : 65.535]$. Each parameter was encoded using 17 bits for each variable. The Foxholes problem, as used in this thesis, is displayed in Figure E.2.
E.3 The Shubert Function

The Shubert function

\[ Shubert(x_1, x_2) = \sum_{i=1}^{5} i \cos [(i + 1) x_1 + i] \cdot \sum_{i=1}^{5} i \cos [(i + 1) x_2 + i] \]  \hspace{1cm} (E.3)\]

has 18 global minima (Li et al., 2002). The values \( x_1 \) and \( x_2 \) were confined to the range \([-10 : 10]\). Additionally, the genotype for this problem was a 60-bit string, with 30 bits per variable. For this thesis, the Shubert function was transformed into a maximisation problem by multiplying by -1. The transformed Shubert function, as used in this thesis, is shown in Figure E.3.
E.4 The Massively Multimodal, Deceptive Problem (MMDP)

*MMDP* is a bitstring problem (of length 30) that has 32 desired optima and over 5 million deceptive optima (Goldberg et al., 1992). It is defined as a concatenation of 5 functions, each working on 6-bit substrings:

\[
MMDP(x) = u(x_{1..6}) + u(x_{7..12}) + u(x_{13..18}) + u(x_{19..24}) + u(x_{25..30})
\]  

(E.4)

where \( u(s) \) is defined as:

\[
u(s) = \begin{cases} 
1.000000, & \text{if } s \in \{0, 6\}; \\
0.000000, & \text{if } s \in \{1, 5\}; \\
0.360284, & \text{if } s \in \{2, 4\}; \\
0.640576, & \text{otherwise}.
\end{cases}
\]  

(E.5)

and \( s \) is the sum of the bits for the given substring. The *MMDP* problem, as used in this thesis, is shown in Figure E.4.
E.5 The Minimum Hamming Distance Problem (*MINHD*)

*MINHD* is a bitstring problem (of length 24) that has 27 desired optima and 2170 deceptive optima (Horn and Goldberg, 1995). It is defined as:

\[
MINHD(x) = f_{mdG}(x_{1..8}) + f_{mdG}(x_{9..16}) + f_{mdG}(x_{17..24})
\]  

(E.6)

where \( f_{mdG}(s) \) is defined as:

\[
 f_{mdG}(s) = \begin{cases} 
 10, & \text{if } s \in G; \\
 \min_{g \in G} H(s, g), & \text{otherwise.}
\end{cases}
\]  

(E.7)

and \( H(s, g) \) is the Hamming distance between any two bitstrings. \( G \) is the set of bitstrings \( \{00000000, 10001100, 01001010\} \). The *MINHD* problem, as used in this thesis, is shown in Figure E.5

**References**


