Compositional Evolution:

Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis.

A Dissertation Presented to

The Faculty of the Graduate School of Arts and Sciences

Brandeis University

Department of Computer Science

Jordan B. Pollack, Advisor

In Partial Fulfilment of the Requirements for the Degree

Doctor of Philosophy

by

Richard A. Watson

May, 2002

This dissertation, directed and approved by Richard A. Watson's Committee, has been accepted and approved by the Graduate Faculty of Brandeis University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Dean of Arts and Sciences

Dissertation Committee

Jordan B. Pollack, Brandeis University.

David L. Waltz, Brandeis University.

Kenneth A. De Jong, George Mason University.

Günter P. Wagner, Yale University.

Acknowledgements

The ideas developed in this dissertation have benefited from the invaluable input of many people. Even before arriving at Brandeis, I was fortunate to have enjoyed the benefits of a vigorous academic community. My MSc advisor Inman Harvey, and members of the community in COGS at Sussex, provided a great environment for the initial ideas to germinate. On commencing at Brandeis, I was welcomed by Maja Mataric but on her departure shortly afterwards, I settled into Jordan Pollack's group known as DEMO - the Dynamical and Evolutionary Machine Organisation. From the outset, Jordan has provided a breadth of background knowledge, an ability to draw connections with existing work, and an incisive ability to see through smoke screens that has been invaluable in the PhD process. His willingness to allow our work to develop in whatever direction it flows naturally allows us to participate in his own enthusiasm for new things and new knowledge.

The members of DEMO, past and present, have provided that essential source of detailed extended conversations where learning and insights happen. Through the years, essential pieces of the puzzle I have been working on have been provided by Alan Blair, Paul Darwen, Hugues Juille, Pablo Funes, Ofer Melnik, Greg Hornby, Elizabeth Sklar, Sevan Ficici, Simon Levy, Anthony Bucci, Hod Lipson, Edwin De Jong, and Shivakumar Viswanathan. Most notably, Sevan – several of the most critical insights in the work would not have become clear without his relentless and rigorous encouragement to get to the heart of the matter.

The Santa Fe Institute Summer School in Complex Systems, 1999, provided introductions to a number of people that have provided a network of invaluable collaborations including Martin Oates, Torsten Reil, Josh Knowles, Rob Shipman, Mark Ebner, and Hugo Touchette. Many members of the evolutionary computation community at large have taken the trouble to share insights and provide perspective: both through anonymous reviews of papers, and in person - David Goldberg, Vladimir Kvasnicka, Bart Naudts, Darrel Whitley; and in particular, Dave Cliff, Jason Noble, Peter Todd, Una May O’Reilly, and Alden Wright.

I have also been fortunate enough to have benefited from valuable feedback from biologists including Mark Ridley, Joel Peck, Dick Lewontin, and John Maynard Smith. But in the last year or so, it has been the many hours of patient discussion with John Wakeley that has done most to clear-up the details.

Finally, the insightful direction of my thesis committee typifies that which can only be provided by experience and expertise in their fields: Gunter Wagner, Dave Waltz, and Ken De Jong.

Some number of these people have been fruitful colleagues and collaborators, and many of them have become my friends – but the love that supports me in these endeavours, and in all things, comes from Laura.

My indebted thanks to all.

ABSTRACT

Compositional Evolution: Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis.

A Dissertation Presented to the Faculty of the Graduate School of Arts
and Sciences of Brandeis University, Waltham, Massachusetts

by Richard A. Watson

Conventionally, evolution by natural selection is almost inseparable from the notion of accumulating successive slight variations. Although it has been suggested that symbiotic mechanisms that combine together existing entities provide an alternative to gradual, or 'accretive', evolutionary change, there has been disagreement about what impact these mechanisms have on our understanding of evolutionary processes. Meanwhile, in artificial evolution methods used in computer science, it has been suggested that the composition of genetic material under sexual recombination may provide adaptation that is not available under mutational variation, but there has been considerable difficulty in demonstrating this formally. Thus far, it has been unclear what types of systems, if any, can be evolved by such 'compositional' mechanisms that cannot be evolved by accretive mechanisms.

This dissertation takes an interdisciplinary approach to this question by building abstract computational simulations of accretive and compositional mechanisms. We identify a class of complex systems possessing 'modular interdependency', incorporating highly epistatic but modular substructure. This class typifies characteristics that are pathological for accretive evolution - the corresponding fitness landscape is highly rugged, has many local optima creating broad fitness saddles, and includes 'irreducibly complex' adaptations that cannot be reached by a succession of gradually changing proto-systems. Nonetheless, we provide simulations to show that this class of system is easily evolvable under sexual recombination or a mechanism of ‘symbiotic encapsulation’. Our simulations and analytic results help us to understand the fundamental differences in the adaptive capacities of these mechanisms, and the conditions under which they provide an adaptive advantage. These models exemplify how certain kinds of complex systems, considered unevolvable under normal accretive change, are, in principle, easily evolvable under compositional evolution.

Contents

ABSTRACT vi

List of Figures xiii

List of Equations xvi

List of Tables xvi

Chapter 1 - Accretive and compositional change in natural and artificial evolution 1

1.1 Exchange between evolutionary biology and evolutionary computation 2

1.1.1 Gradualism and the probability of large adaptive changes 2

1.1.2 Evolutionary adaptation and artificial optimisation methods 3

1.2 Accretive and compositional mechanisms 4

1.3 Complex systems with modular interdependency 7

1.3.1 Evolvability and algorithmic paradigms 7

1.3.2 Modular interdependency 10

1.4 The (un)evolvability of systems with modular interdependecy 14

1.5 Conditions for the adaptive advantage of composition 17

1.5.1 Composition based on sexual recombination 18

1.5.2 Composition based on symbiotic encapsulation 20

1.6 Impact for evolutionary biology and evolutionary computation 22

1.7 Related issues 26

1.7.1 Related issues in Evolutionary Biology 26

1.7.2 Related issues in Evolutionary Computation 28

1.8 Summary 32

1.8.1 Motives, research questions, and claims 32

1.8.2 Approach and outline of models 33

1.8.3 Contributions 35

1.8.4 Scope 36

1.8.5 Dissertation structure 37

Chapter 2 - Accretion and Composition in Evolutionary Biology 40

2.1 The accretive model 40

2.1.1 Successive slight modifications/ gradualism 40

2.1.2 Evolutionary difficulty under accretion 42

2.1.3 Consequences of the accretive model 47

2.2 Compositional mechanisms 48

2.2.1 A spectrum of compositional mechanisms 48

2.2.2 Sexual recombination 49

2.2.3 Interspecific genetic integration 52
2.2.4 Compositional evolution 56
2.2.5 Is gradualism necessary and/or sufficient? 57

2.3 Other related mechanisms 59
2.3.1 Large phenotypic changes from small genotypic changes: ontogenesis 60
2.3.2 Mechanisms for large genetic changes within a lineage: Gene duplication 61
2.3.3 Large changes within a lineage and self-similarity 61
2.3.4 Within a lineage vs between parallel lineages 62

2.4 Evolutionary difficulty under composition 63
2.4.1 Transferring concepts of accretive difficulty (where possible) 63

2.5 Discussion 68
2.5.1 Single inheritance and multiple-inheritance 68
2.5.2 The rest of the genotype as a ‘backdrop’ vs important epistasis 71
2.5.3 Selection on interaction systems: Shifting Balance Theory 72
2.5.4 Other species as a ‘backdrop’ vs important interspecific relationships 74
2.5.5 Intragenomic dependencies and intergenomic dependencies 75

2.6 Summary 75

Chapter 3 - Accretion and Composition in Evolutionary Computation 76

3.1 Evolutionary Algorithms 76
3.1.1 Evolutionary algorithm basics 76
3.1.2 The Simple Genetic Algorithm 78
3.1.3 EAs as Evolutionary Models 80

3.2 Evolutionary difficulty for the Simple EA 82

3.3 Compositional mechanisms in EAs 83
3.3.1 The operation of crossover 84
3.3.2 The Building block Hypothesis, BBH 86
3.3.3 Algorithmic Principles - Divide-and-Conquer Problem Decomposition 89
3.3.4 Mechanisms of encapsulation in EAs 93
3.3.5 Other related divide and conquer methods 94
3.3.6 Explicit symbiosis/symbiogenesis models 96
3.3.7 Summary of existing methods 100

3.4 Evolutionary difficulty under compositional mechanisms 101
3.4.1 Difficulties of building block processing under recombination 101
3.4.2 Schema disruption: genetic linkage, partial specification, and competing conventions. 101
3.4.3 Evaluation of parts 105
3.4.4 Selection on parts: The credit assignment problem 106
3.4.5 Coevolution of parts: premature convergence/competitive exclusion and fitness sharing. 107
3.4.6 The multi-dimensional interpretation of fitness and ‘Pareto coevolution’ 108

3.5 Test problems 110
3.5.1 Building block problems 111
3.5.2 Interdependency 112
3.5.3 Other test problems for EAs 113
3.5.4 Modularity and interdependency: non-separable modules 114

3.6 Summary **115**

3.7 Accretion and composition in evolutionary biology and evolutionary computation **116**

Chapter 4 - Defining Modular Interdependency **119**

4.1 Philosophical provisos: The 'problem' for an evolving entity **120**

4.2 Pairwise interactions **120**
4.2.1 Pairwise interactions and fitness saddles 120
4.2.2 Defining 'interdependency' 123

4.3 Large-Scale interdependency structure **124**
4.3.1 Separability and decomposability to *modular interdependency* 126
4.3.2 An example non-separable but decomposable system of four variables 130

4.4 Generalised models **134**
4.4.1 Aggregate effects of modules: dimensional reduction 134
4.4.2 Hierarchy and scaling-up 137
4.4.3 General recursive form 139
4.4.4 Hierarchical consistency 140
4.4.5 Alternate base functions 141

4.5 Hierarchical-Equality and Hierarchical-if-and-only-if (HIFF) **143**

4.6 Discussion **146**
4.6.1 Symmetries and variations 146
4.6.2 The HIFF landscape and natural hierarchy 146
4.6.3 Other hierarchical modular interdependency functions 147
4.6.4 Comparison with Royal Roads, concatenated trap functions, and NKC landscapes 147
4.6.5 Modular interdependency as a hierarchical cooperation problem 149
4.6.6 How changes in Module A affect the fitness landscape for Module B 150
4.6.7 Relation to philosophical provisos 151

4.7 Summary **151**

Chapter 5 - Mutation on Modular Interdependency **153**

5.1 Cross-sections through the fitness landscape **153**

5.2 Difficulty of modular interdependency for accretive mechanisms **157**
5.2.1 Ruggedness 157
5.2.2 Width of fitness saddles 158
5.2.3 Irreducibility - HIFF *appears* to be irreducibly complex (but is not) 159
5.2.4 HIFF *appears* to have low epistasis (but does not) 161

5.3 Expected time to solution for accretive mechanisms **162**
5.3.1 Preliminary Comparison with compositional mechanisms 163

5.4 Simulation results for mutation 164

5.5 Summary 166

Chapter 6 - Sexual Recombination on Modular Interdependency 168

6.1 Overview of models 170

6.2 Basic Model: Single panmictic population—Simple GA 170
6.2.1 Simulation results 171
6.2.2 Discussion 173

6.3 Diversity: Subdivided/niched population—GA with crowding 174
6.3.1 Mutation is not essential 180
6.3.2 Gradualism is neither sufficient nor required 181

6.4 Genetic linkage 181
6.4.1 No genetic linkage: ‘Free recombination’ or ‘uniform crossover’ 182
6.4.2 Re-ordered linkage: Shuffled gene positions 186
6.4.3 Convergence controlled variation 190
6.4.4 Linkage learning 192

6.5 Summary of simulation results 193

6.6 Analysis of sexual recombination on HIFF 197
6.6.1 Analysis on separable problems 199
6.6.2 Analyses on HIFF 203
6.6.3 Analyses for a recombinative hill-climber 205
6.6.4 From a recombinative hill-climber to a recombinative population 211

6.7 Empirical illustration of analytic results 213
6.7.1 Simulation for Theorem 4: Recombinative Hill-climber with one-point crossover on H-IFF. 213
6.7.2 Simulation for Theorem 5: GA with two-point crossover applied to H-IFF2 214
6.7.3 Summary of analysis section 217

6.8 Summary of sexual recombination on hierarchically modular problems 217

Chapter 7 - Symbiotic Encapsulation on Modular Interdependency 219

7.1 Overview of symbiotic encapsulation model 219

7.2 Integration model: Entities and their encapsulation 221
7.2.1 Mutually exclusive characters and the ‘overlap’ of feature sets 221
7.2.2 A model of entities and their encapsulation 225
7.2.3 Consequences for adaptation 227

7.3 Evaluation and selection model 228
7.3.1 Evaluating an entity in different contexts 229
7.3.2 Pareto dominance 230
7.3.3 Building environmental contexts: Group evaluation 232

7.4 The Symbiogenic Evolutionary Adaptation Model (SEAM) 233

7.5 Simulation results 234
7.5.1 Control experiments 235

7.5.2 Simulation results for SEAM and control experiments 236
7.5.3 Summary of main simulation experiments using SEAM and GA. 239

7.6 Discussion 241
7.6.1 Coevolution and SEAM 241
7.6.2 Canalisation of successful groups 242
7.6.3 Scale-invariant evolutionary processes 243

7.7 Analysis of SEAM on Shuffled HIFF 245
7.7.1 Idealised-SEAM 245
7.7.2 From idealised-SEAM to SEAM 250
7.7.3 Summary of analytic results 255

7.8 Summary 256

Chapter 8 - Implications for Evolutionary Biology 258

8.1 Impact for evolutionary biology 258
8.1.1 Evolutionary difficulty and gradualism 258
8.1.2 Symbiosis as source of evolutionary innovation 259
8.1.3 Multiple-inheritance allows large non-random genetic changes 260
8.1.4 Assumptions concerning genetic systems 261

8.2 Related results 261
8.2.1 Hierarchical encapsulation, the Baldwin effect, and symbiotic scaffolding 261

8.3 Future research 263
8.3.1 The relationship of accretive and compositional mechanisms 263
8.3.2 The open-endedness of evolutionary processes 266
8.3.3 The inherent tension of innovation and reproductive fidelity (change and non-change) 267
8.3.4 Selfish genes and the evolution of cooperation 268
8.3.5 Modular interdependency in natural systems 269
8.3.6 The ubiquity of compositional mechanisms 271

Chapter 9 - Implications for Evolutionary Computation 272

9.1 Impact for Evolutionary Computation 272
9.1.1 Building Block Hypothesis 272
9.1.2 Building blocks and modules - genetic linkage vs. interdependency 272
9.1.3 GA-easiness 273
9.1.4 Interdependency and NK landscapes 274
9.1.5 Module acquisition and linkage learning 275
9.1.6 Crossover is not just macro-mutation 275
9.1.7 Competing conventions problem 275
9.1.8 Credit assignment 275
9.1.9 Separation of the Building Block Hypothesis from the Schema Theorem 276
9.1.10 Evolution and coevolution in the GA with crossover 276
9.1.11 Optimisation at several scales vs. optimisation at any one scale 277

9.2 Understanding modularity 280
9.2.1 Modular interdependency, Ising models, and renormalisation groups 280
9.2.2 Simon and nearly decomposable systems 282
9.2.3 Deception 284

9.2.4 Relationship of HIFF to some related concepts of problem difficulty 286

9.3 Understanding composition 287

9.3.1 Recombining versus combining 288

9.3.2 Problems with partial evaluation, bloat, and diversity 288

9.3.3 Template differencing 290

9.3.4 The pressure to fill-in missing features 291

9.3.5 The Baldwin effect and 'stochastic lookahead' 292

9.3.6 Stochastic lookahead avoids need for 'backtracking' in greedy optimisation 293

9.3.7 Pareto Coevolution 295

9.3.8 Relationship of SEAM to some other algorithmic methods 297

9.4 Future research 298

9.4.1 Theoretical 298

9.4.2 Possible applications of SEAM 299

9.4.3 Extensions of SEAM 300

Chapter 10 - Summary and Conclusions 303

Subject Index 308

References 311

List of Figures

Figure 1-1: Mechanisms of accretive change. 5

Figure 1-2: Mechanisms of compositional change. 5

Figure 1-3: Subdivision and integration 6

Figure 1-4: A fitness saddle created by interdependency between two variables. 11

Figure 1-5: A system of variables exhibiting modular interdependency 12

Figure 1-6: Alternate dependency structures. 13

Figure 1-7: A particular cross-section through a fitness landscape derived from a system with modular interdependency 17

Figure 1-8: Alternate genetic linkage possibilities. 19

Figure 2-1: A fitness landscape. 43

Figure 2-2: Population mean climbing a local fitness peak. 44

Figure 2-3: A fitness saddle. 45

Figure 2-4: A hypothetical irreducibly complex system........46
Figure 2-5: Mechanisms of genetic composition........48
Figure 2-6: Crossover........50
Figure 2-7: Endosymbiosis.........54
Figure 2-8: Hypercycles with multi-level structure.........67
Figure 3-1: A procedural outline for a generational ‘Simple GA’, SGA.........79
Figure 3-2: Point mutations on a bit-string individual.........79
Figure 3-3: smoothness and local optima........83
Figure 3-4: Crossover operations on bit-string individuals. 84
Figure 3-5: Schemata of various orders and defining lengths 87
Figure 3-6: Outline of algorithmic advantage from problem decomposition 91
Figure 3-7: Building block combination and disruption........102
Figure 3-8: Representation of genomes/schemata for GA, Messy GA, and ‘partial specification’.........104
Figure 3-9: ‘Splice’ recombination in the Messy GA.........104
Figure 3-10: An alternate representation of the splice operation in Figure 3-9.........104
Figure 4-1: Different kinds of fitness interactions.........121
Figure 4-2: A fitness saddle created by interdependency between two variables.........122
Figure 4-3: A dependency graph illustrating the dependencies of the four variables in Equation 3. 132
Figure 4-4: A particular cross section through the fitness landscape of Equation 3.........133
Figure 4-5: Abstracting the interaction between modules.........135
Figure 4-6: The fitness landscape for A, for different configurations of B. 150
Figure 5-1: ‘Ridge mutation walker’ 154
Figure 5-2: A particular cross-section through the HIFF fitness landscape........155
Figure 5-3: A section through the hierarchical modular interdependency system built on f2.........156
Figure 5-4: Base fitness functions 157
Figure 5-5: Results of RMHC on a 128-variable decomposable system.........165
Figure 5-6: Results of RMHC on a 128-variable decomposable system.........166
Figure 6-1: A procedural outline for a simple ‘panmictic’ GA. 171
Figure 6-2: Results of simple panmictic GA with one-point crossover on a 128-variable HIFF. 172
Figure 6-3: Results of simple GA with one-point crossover on a 128-variable HIFF.........173
Figure 6-4: A procedural outline for a simple steady-state GA with Deterministic Crowding. 175
Figure 6-5: Results of deterministic crowding GA with one-point crossover on 128-variable HIFF........178
Figure 6-6: Results of deterministic crowding GA with one-point crossover on 128-variable HIFF........179
Figure 6-7: Variations on crowding algorithm 180
Figure 6-8: Results of DC GA with uniform crossover on a 128-variable HIFF. 183
Figure 6-9: Results of DC GA with uniform crossover on a 128-variable HIFF 184
Figure 6-10: Crossover operations on bit-string individuals. 185

Figure 6-11: Crossover operations with random genetic linkage - i.e. random gene ordering. 186
Figure 6-12: Alternate genetic linkage possibilities. 188
Figure 6-13: Results of DC GA with one-point crossover on a 128-variable Shuffled HIFF. 189
Figure 6-14: Results of DC GA with one-point crossover on a 128-variable Shuffled HIFF. 190
Figure 6-15: Summary of simulation experiments 1a - RMHC and GA methods on HIFF. 194
Figure 6-16: Summary of simulation experiments 1a - size of largest module discovered. 195
Figure 6-17: Summary of simulation experiments 1b - RMHC and GA methods on Shuffled HIFF. 196
Figure 6-18: Summary of simulation experiments 1b - size of largest module discovered. 197
Figure 6-19: A simple form of the Gene Invariant GA. 199
Figure 6-20: A modified Gene Invariant GA. 201
Figure 6-21: Recombinative hill-climber based on GIGA. 205
Figure 6-22: Illustrating that Condition 1 holds. 208
Figure 6-23: Performance of recombinative hill-climber on H-IFF. 214
Figure 6-24: Performance of GA on H-IFF2, various problem sizes & population sizes...........................215
Figure 6-25: Performance of GA on H-IFF2 using smallest reliable population. 216
Figure 7-1: A caricature of processes in SEAM. 221
Figure 7-2: Mechanisms of integration from two donor individuals, P1 and P2. 224
Figure 7-3: Symbiotic encapsulation 226
Figure 7-4: A partially specified entity must be assessed in a context. 229
Figure 7-5: Building a context from other entities. 233
Figure 7-6: Pseudocode for a simple implementation of SEAM. 234
Figure 7-7: Simulation results for SEAM and control experiments. 237
Figure 7-8: Simulation results for SEAM and control experiments: Size of modules discovered. 239
Figure 7-9: Summary of simulation experiments 2 - SEAM and GA. 240
Figure 7-10: Summary of simulation experiments 2: SEAM and GA - size of largest correct module. 241
Figure 7-11: Size of modules in individuals and in groups in SEAM. (log scale) 243
Figure 7-12: Pseudocode for ‘idealised-SEAM’ i.e. SEAM using ‘ideal contexts’. 247
Figure 8-1: Possible interdependency structures 269
Figure 9-1: Relationship of HIFF to other test problems...287
Figure 9-2: Relationship of SEAM to other EA methods...298

List of Equations

Equation 1: Sum of pairwise interactions 125

Equation 2: Set of maximal configurations 128

Equation 3: The weighted sum of pairwise interactions over N=4 variables 130

Equation 4: Re-writing Equation 3 using aggregate variables. 136

Equation 5: The symmetric weighted sum of pairwise interactions over N variables 137

Equation 6: 5 rewritten 138

Equation 7: Recursive form of pairwise modular interdependency 139

Equation 8: Recursive form of 'k-wise' modular interdependency 140

Equation 9: General Fitness function for Hierarchically Consistent Modular Interdependency 140

Equation 10: Hierarchical-Equality no-scaling 144

Equation 11: Hierarchical-if-and-only-if (HIFF) and Hierarchical-Equality (H-Equal) 145

Equation 12: Expected time to solution in terms of steps on path to optimum 204

Equation 13: 12 more accurately 209

Equation 14: 13 simplified 209

Equation 15: HIFF2 = HIFF with arbitrary global optima. 212

Equation 16: Composition of two entities 227

Equation 17: Weighted sum of context-sensitive-fitnesses 229

Equation 18: Stability of a+b. 231

Equation 19: Building a context 233

Equation 20: Expected time to solution for SEAM 249

List of Tables

Table 1-1: Algorithmic classes and their evolutionary analogues 9

Table 3-1: Components of an evolutionary algorithm 77

Table 3-2: Summary of features in existing methods 100

Table 3-3: Analogies between accretion and composition in EB terms and EC terms. 118

Table 4-1: Truth table for if-and-only-if, IFF 124

Table 4-2: A weight matrix defining (with Equation 3) a system of two interdependent modules. 130

Table 4-3: Configurations for a 4-variable system with their fitnesses. 131

Chapter 1 - Accretive and compositional change in natural and artificial evolution

Systems of many complex interdependent parts can appear unevolvable - especially when any small change in the system causes the system to cease functioning. In general, when faced with the existence of a particular complex system in nature, the normal approach to explain how it may have evolved is to show that there exists a succession of proto-systems that are gradually increasing in function where each one is a small modification of the last. Accordingly, a straightforward approach to explain the plausible evolution of a system where it is proposed that any small change causes it to cease functioning is to show that this is in fact not the case - that there does after all exist a succession of gradually changing proto-systems approaching the system in question. If such a succession can be found, or reasonable hypotheses about the plausible existence of such a path can be upheld, then all well and good. But is this strictly necessary?

That is, is it *essential* to show that there exists a succession of gradually improving proto-systems arriving at a given complex system in order to support the possibility that the system was evolved? And likewise, is it the case that it is *only* systems that permit such a gradual succession that can be evolved?

In this dissertation we consider the impact that a certain class of evolutionary mechanisms has on these questions and the necessity of gradualism in evolution. Specifically, we consider the effect of mechanisms such as sexual recombination and symbiotic encapsulation which are instances of a class of mechanisms that can enable what we will term *compositional evolution*. We arrive at the following thesis:

> Certain kinds of complex systems, considered unevolvable under normal accretive change, are, in principle and under certain circumstances, easily evolvable under compositional change.

In the course of this dissertation we will describe the distinction between compositional and accretive mechanisms, the class of complex system that is at issue, the properties of such systems that make them difficult for accretive evolution, and the circumstances under which they are easily evolvable via compositional mechanisms. In the remainder of this introduction we will briefly outline the main ideas and findings of the dissertation in these respects. Throughout this work, our approach to addressing these issues involves the use of abstract computational models. In the following section, we outline the motivations for this interdisciplinary approach.

1.1 Exchange between evolutionary biology and evolutionary computation

This dissertation examines a number of conceptual links between evolutionary biology and evolutionary computation. The work develops conceptual understanding of natural genetic and evolutionary processes by utilizing concepts and algorithmic approaches from computer science. It also develops understanding and new approaches to artificial evolution and optimisation techniques in computer science by utilising concepts and inspiration from evolutionary biology. Interdisciplinary exchange provides an opportunity to explore biological issues where algorithmic principles are integral to our understanding of the biological processes. In particular, questions about evolvability, the likelihood of complex adaptations, and the class of landscapes in which continued adaptation is possible provide a fertile area of investigation. By examining different assumptions and mechanisms in natural systems in light of their relationship with different algorithmic classes and computational mechanisms in computer science we gain leverage in our understanding of each discipline by borrowing from the conceptual frameworks of the other.

1.1.1 Gradualism and the probability of large adaptive changes

Consider further the questions concerning gradualism in evolution raised above. It will be clear to some readers that it is *not* essential to show that there exists a succession of gradually changing proto-systems arriving at a given complex system in order to support the possibility that the system could be evolved. Natural selection does not in itself exclude the possibility of large changes in function (so long as they are not significantly deleterious in fitness) and there are natural mechanisms available that can provide large changes in function. For example, an accumulation of neutral genetic changes can subsequently arrive at a configuration that is non-neutral and genetically distant from the ancestor type. Alternatively, sophisticated ontogenetic mechanisms can provide large changes in phenotype from small changes in genotype.

Similarly, it is a well-known hypothesis that the encapsulation of symbionts into a new reproductive entity by, for example, endosymbiosis, as we shall discuss, may provide a means for large and relatively sudden increases in complexity.

The question then becomes not whether large changes in the function of an evolved system are possible in principle, but when, and for what class of domains, large changes provided by such mechanisms are *likely* to be adaptive. Moreover, in order for such mechanisms to warrant a significant change in our understanding of evolution, and in our understanding of what kind of system is evolvable, it is necessary to identify the circumstances, if any, under which the probability of evolving a complex system under such mechanisms is significantly greater than the probability of evolving such systems without such mechanisms. Given the absence of clear analyses showing that such mechanisms can provide adaptive change more easily than that which is available under gradual evolutionary change, it is not surprising that the assumption of gradualism remains ubiquitous in evolutionary thought.

However, we can make some progress in understanding the possibilities by defining cases where the differences in the relevant probabilities would be fundamental rather than marginal. Identifying fundamental differences in the adaptive capacities of such mechanisms assists us in understanding the impact that such mechanisms may have in natural evolution and it is in this respect that our understanding of different algorithmic frameworks and combinatorial analyses from computer science can provide a valuable contribution. In this dissertation we develop concrete abstract models of evolutionary processes under compositional mechanisms, and detail a sufficient set of conditions where the likelihood of evolving certain kinds of complex systems under compositional mechanisms is fundamentally different and significantly greater than the probability of evolving such systems without such mechanisms. The computational models that we develop provide a means to analyse the impact of such mechanisms in a formal manner and thereby complement existing intuitions about the biological possibilities.

1.1.2 Evolutionary adaptation and artificial optimisation methods

At the same time, there exists considerable interest within computer science in computational optimisation methods based loosely on Darwinian natural selection. Work in ‘evolutionary algorithms’, EAs, is a highly active and rapidly growing field. Moreover, there is significant controversy over the supposed utility of sexual crossover in such algorithms, and considerable interest in the possibility of methods that encapsulate primitive functional units into higher-order components that can be re-used in subsequent evolution. These issues have strong parallels with the biological mechanisms of composition we mentioned above. Accordingly, as we develop computational models to help us understand biological possibilities we also answer some important questions impacting evolutionary algorithms as optimisation techniques.

1.2 Accretive and compositional mechanisms

In this thesis we focus on the impact of a particular class of adaptive mechanisms on evolutionary change. We call these mechanisms *compositional* mechanisms in contrast to normal *accretive* mechanisms.

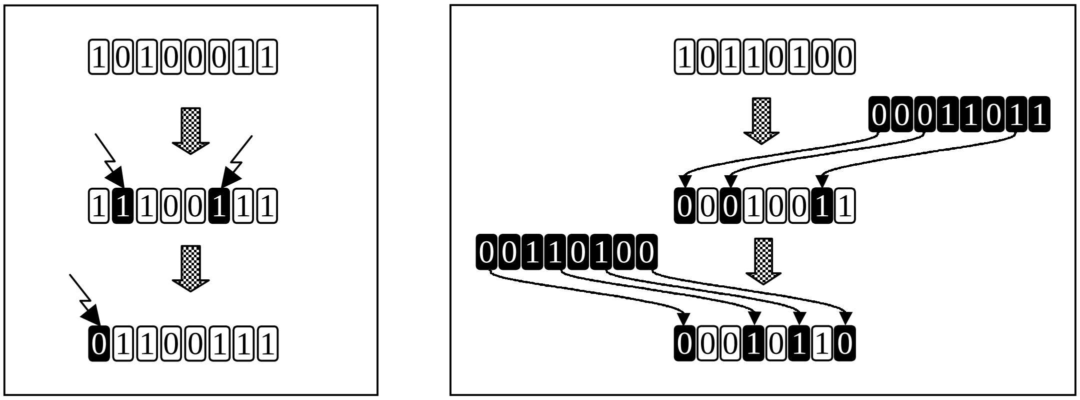
- We use the term ‘compositional’ to refer to variation mechanisms that combine together systems or subsystems of genetic material, or feature complexes, that have been semi-independently pre-adapted in parallel in different lineages.

  Examples in nature include: Normal mechanisms of sexual recombination (under particular conditions of population diversity and genetic linkage from the arrangement of genes on the chromosome), and mechanisms of interspecific combination such as horizontal gene transfer; also, mechanisms of ‘symbiotic encapsulation’ such as endosymbiosis or other mechanisms that encapsulate a group of simple entities into a complex entity at a higher level of organisation, as exhibited in several of the major transitions in evolution.

- In contrast, we use the term ‘accretive’ to refer to variation mechanisms that accumulate random variations in genetic material or features, i.e. the new genetic material or features that are introduced by such changes have not been pre-adapted elsewhere as a set. Thus accretive evolution is driven predominantly by small modifications, i.e. ‘successive slight modifications’ (Darwin 1859), which forms the basis of our common understanding of evolutionary change.

  Examples in nature include: Genetic mutation and sexual recombination in the absence of favourable genetic linkage.

Figure 1-1 & Figure 1-2 show simple representations of possible accretive and compositional mechanisms.

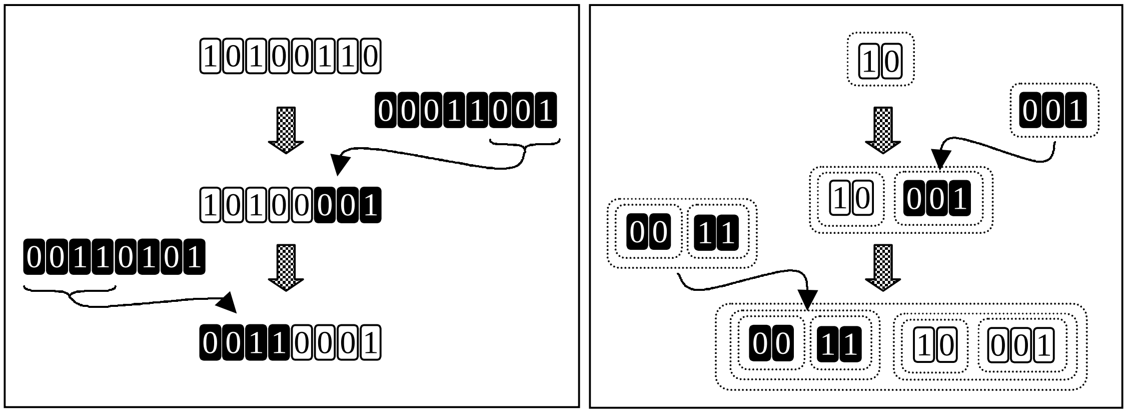


Mutation

Sexual recombination without genetic

Figure 1-1: Mechanisms of accretive change.

We use the term ‘accretive’ to refer to variation mechanisms that accumulate random variations in genetic material or features, i.e. the new genetic material or features that are introduced by such changes have not been pre-adapted elsewhere as a set. Left) Mutational variation: newly introduced genetic material (shown in grey) is entirely random. Right) Variation from sexual recombination without genetic linkage: i.e. the set of alleles that are newly introduced come from a subset of genes distributed randomly on the chromosome. Thus although the alleles may have been selected for in the donor individual, the particular subset of alleles introduced is not pre-adapted as a set (we will discuss this issue at length later). Since newly introduced genetic material is random in these mechanisms, large beneficial changes are likely to be rare and accretive evolution is driven predominantly by the accumulation of small modifications.



Sexual recombination with strong genetic linkage

Symbiotic encapsulation

Figure 1-2: Mechanisms of compositional change.

We use the term ‘compositional’ to refer to variation mechanisms that combine together systems or subsystems of genetic material, or feature complexes, that have been semi-independently pre-adapted in parallel in different lineages. Left) Sexual recombination with strong genetic linkage: When crossover points are few, subsets of adjacent genes may be incorporated as a set. Right) Hierarchical encapsulation of symbiotic groups: Pre-adapted sets of genetic material may be assembled together by endosymbiosis. The effect is much like that of sexual recombination with strong genetic linkage but the result is better described as a union of the donors' material rather than a 'half and half'. Importantly, since there is no alignment and exchange of corresponding sub-parts (as in sexual recombination), this operation is not necessarily sensitive to the position of genes on the chromosomes involved.

Compositional mechanisms, such as those depicted in the previous figure, form part of a general evolutionary process of subdivision and integration. Abstractly, a process of subdivision and integration exhibits the following components. Evolving entities of different types are evolved in parallel lineages - i.e. the population of evolving entities is *subdivided*. Subsequently, some mechanism of *integration* composes genetic material from different lineages together into a new entity. After integration, integrated genetic material reproduces together. Figure 1-3 shows a schematic of this process.

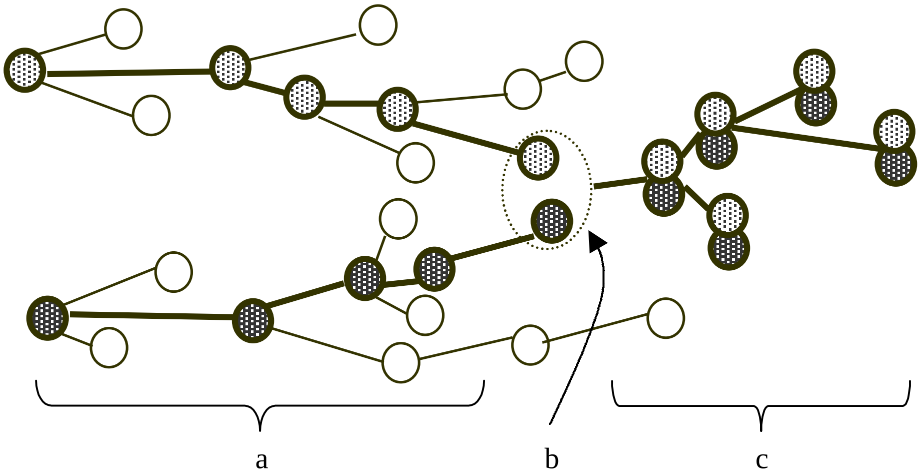


Figure 1-3: Subdivision and integration

A number of reproductive entities, shown by nodes, in different lineages. Lines in the graph represent ancestral relationships with evolutionary time progressing from left to right: a) two particular lineages (light shading and dark shading) progress independently in a subdivided population, b) some mechanism of integration, such as sexual recombination or symbiotic encapsulation, composes together pre-adapted subsets of genetic material from the previously separate lineages. c) the integrated genetic material is now replicated together. The integration of genetic material from subdivided populations may occur repeatedly in compositional evolution.

Mechanisms of subdivision and integration are seen at different scales in natural systems. Specifically, at the single-species scale, the separate lineages involved may be sub-populations of a given species, semi-isolated perhaps by virtue of spatial location, and the mechanism of integration may be provided by migration and sexual reproduction. At the multi-species scale, the separate lineages may be separate reproductively isolated species, and integration may occur through horizontal gene-transfer or endosymbiosis. Several of the major transitions in evolution share the characteristic that "entities that were capable of independent replication before the transition can replicate only as part of a larger whole after it", (Maynard Smith and Szathmary 1995), and this notion is important in providing the idea that processes of composition may occur through several increases in scale.

In evolutionary biology, the single-species scale is relevant to models such as Shifting Balance Theory (Wright 1977), and the multi-species scale is relevant to models such as Serial Endosymbiosis Theory (Margulis 1993a). In evolutionary computation, the single-species scale is relevant to issues such as diversity maintenance methods, the utility of crossover, and the Building Block Hypothesis (Holland 1975, 2000, Goldberg 1989) - and the multi-species scale is relevant to models such as Cooperative Coevolution (Potter 1997) and the explicit encapsulation of primitive components into re-usable composites in models such as Automatic Module Acquisition (Angeline & Pollack 1993).

In the models that we detail in this dissertation we include models of each of the four mechanisms illustrated in Figure 1-1 & Figure 1-2 in different types of evolutionary algorithms incorporating, more or less explicitly, mechanisms of subdivision and integration. We contrast the impact that these mechanisms have on evolvability and the likelihood of evolving certain classes of complex systems. In the next subsection we describe a class of complex systems that we will use to exemplify their different adaptive capacities.

1.3 Complex systems with modular interdependency

1.3.1 Evolvability and algorithmic paradigms

We claim a fundamental distinction between accretive and compositional mechanisms based on the likelihood of evolving certain kinds of complex systems. The basic intuitions here are provided by analogy with well-known classes of optimisation algorithms in computer science. In optimisation, different kinds of optimisation algorithms are suitable in different kinds of problems according to the assumptions we have about the dependencies between the problem variables being optimised. A dependency between variables occurs when the optimal setting for one variable is dependent on the setting of other variables. Table 1-1 outlines three main classes of assumptions and the algorithm types that are appropriate for each of them. Starting with the leftmost column, when we have no knowledge of the nature or structure of interdependencies between the problem variables we might assume that the dependencies are arbitrarily difficult. Such dependencies create a random, and generally highly-rugged fitness surface, and the appropriate algorithmic paradigms for this class of problem are exhaustive search or random search. These methods have computational complexity that is exponential in the number of problem variables - that is, neither method uses any assumptions about the problem domain to guide or reduce the search necessary and therefore the expected time to find high-fitness configurations of variables is proportionate to the size of the entire search space.

At the other extreme, addressing the rightmost column, we may in other cases hold different assumptions about the interdependencies of variables - in this case, that interdependencies between variables are weak or few. This creates a relatively smooth, and generally mono-modal fitness landscape, and the appropriate algorithmic paradigms are ‘greedy’ algorithms making incremental improvements, and the simple class of ‘hill-climbing’ algorithms, climbing the local gradient in the fitness landscape. When the assumption of weak interdependencies between variables is correct, the expected time to find high-fitness configurations of variables is linear, that is, simply proportionate to the number of variables in the problem.

These two cases - on the one hand, arbitrary interdependencies and exponential time complexity, and on the other hand, weak interdependencies and linear time complexity - represent the naïve extremes of the possibilities that might occur in a problem domain. These extremes correspond closely to extreme positions prevalent in evolutionary thought: specifically, if one assumes that the interdependencies between components of a natural system are arbitrarily difficult and complex then it is inconceivable that that system may have been evolved, or at least, the size of complex systems of this type that can be achieved with ‘uninformed’ processes is limited; in contrast, if one assumes that the interdependencies between the components of a natural system are weak or few then such systems are easily explained by the normal accretive model of evolutionary change.

In between these two extremes there are many other possibilities for the number, structure, and nature of variable interdependencies, and in optimisation, there are important algorithmic paradigms that address this range. One possibility for the interdependencies of variables is that there may be strong and numerous interdependencies but that these dependencies may have a particular structure - for example, a modular structure. (It is not so obvious how to represent the high-dimensional landscape of a modular problem domain in the simple one-dimensional caricatures used here, but as we will see later, properties of symmetry and self-similarity may be representative.) The algorithmic paradigm that is appropriate for problems of this class broadly includes those known as ‘divide and conquer’ techniques. Divide and conquer, D&C, optimisation is a form of problem decomposition that utilises the modular structure of a problem by breaking it into smaller sub-problems that are easier to solve, and using solutions to these subproblems to compose a solution to the problem as a whole. For example, if a problem can be divided into two equal halves then each requires time exponential in N/2 (i.e. half the number of problem variables), at worst. Importantly, under most circumstances there is some extra work to be done (‘c’) in determining how to compose together subsolutions to form whole solutions.

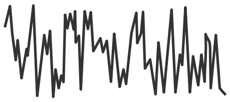
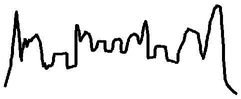
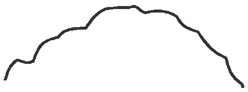
Dependency of variables	arbitrary interdependencies	modular interdependencies	few/weak interdependencies
Landscape			
Algorithmic paradigm	Exhaustive search, random search	Divide and Conquer problem decomposition	Hill-climbing – accumulation of small variations
Complexity	K^N	$2K^{N/2}$+c	KN
Evolutionary analogue	‘impossible’/ ‘Intelligent design’	Compositional evolution	Accretive evolution

Table 1-1: Algorithmic classes and their evolutionary analogues

Following these broad algorithmic classes, the basis of our thesis is that compositional evolutionary mechanisms can provide adaptation akin to divide and conquer problem decomposition under certain circumstances. As is the case for D&C methods in general, the adaptive advantage of compositional mechanisms will depend on the structure of interdependencies in the problem domain - in particular the presence of a modular interdependency structure. If such a structure is present, then compositional mechanisms may provide a form of adaptation that lies in between, yet is fundamentally distinct from, either of the extreme scenarios mentioned above. The possibility of compositional evolution, and the fundamental distinction from accretive change, shares underlying intuitions with some previous work (e.g. Bermudes & Margulis 1985, Margulis 1993a). But the association of these intuitive ideas with this algorithmic class has not been previously recognised, and formal analyses of the relevant combinatorics have not been developed in evolutionary theory. In computer science, the underling intuition here is basically that which underlies the Building block Hypothesis that has been present in evolutionary computation, more or less explicitly, since its inception (Holland 1975, 2000). But despite several attempts to demonstrate the validity of this hypothesis, (e.g. Mitchell et al 1995, Forrest & Mitchell 1993b), this has proved difficult. In large part, the difficulty in formalising the possible impact of compositional evolution in natural evolution, and the difficulty of demonstrating optimisation of a compositional style in artificial evolution, stems from an inadequate understanding of modularity. In the next subsection, we describe a class of complex systems that is sufficient to properly exemplify the difference in the adaptive capacities of accretive and compositional mechanisms.

1.3.2 Modular interdependency

In a given system of variables, the configuration of a *subset* of variables that maximises the fitness of the system may depend on the setting of the *remaining* variables in the system. A system can be understood as *modular* if it can be described in terms of subsets of variables where the number of different configurations for a subset that could give maximal fitness (given all possible configurations of variables in the remainder of the system) is low. If there is only one configuration for a module that could be maximal then the module may be optimised entirely independently of the remainder of the system. However, this kind of complete independence is a simplistic conception of modularity. In contrast, if a system can be described as a number of modules where the number of possibly maximal configurations for each module is *low but greater than one*, then the dimensionality of the system can be reduced, perhaps greatly, even though the modules are not independent. We call this *modular interdependency*. In systems exhibiting modular interdependency, this property means that the configuration space that must be considered to find high-fitness configurations can be significantly reduced by an adaptive mechanism that is able to discover and manipulate modules effectively. However, a module that has been optimised in one context may nonetheless be far from optimal (configurationally and fitness-wise) in another context. Thus accretive mechanisms that are unable to manipulate modules effectively are ineffective in this class of systems.

In Chapter 4 we will be detailing a particular class of system exhibiting modular interdependency and we will build upon this to illustrate the distinction between accretive and compositional evolution. We start with a basic pairwise relationship between variables: specifically, we define the fitness interactions between two variables such that a 'fitness saddle' is created. This is illustrated in Figure 1-4.

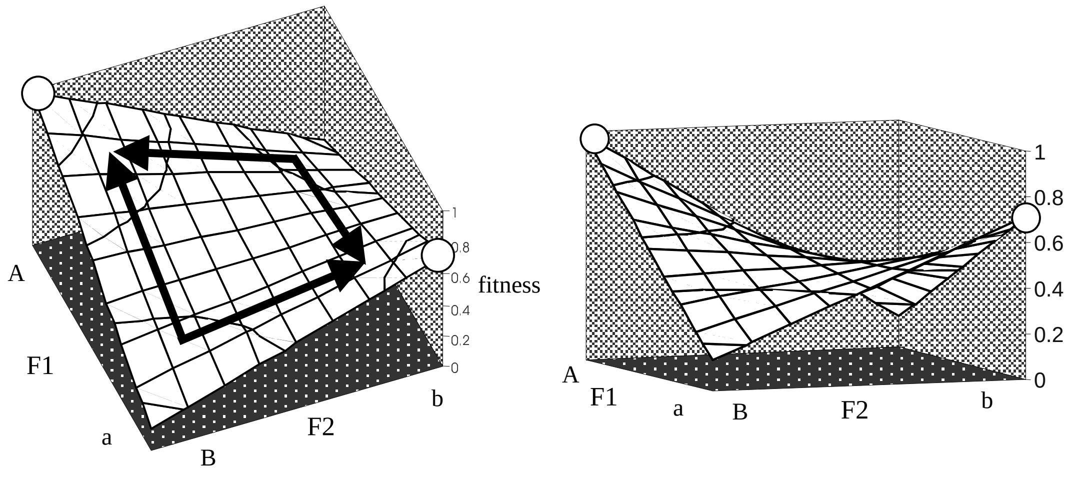


Figure 1-4: A fitness saddle created by interdependency between two variables.

A system of two variables, or features, F1 and F2, showing non-additive fitness interaction or 'epistasis'. Left) The surface shows a fitness landscape and the overlayed arrows show the paths of increasing fitness that can be followed by changing one feature at a time. Right) This particular kind of interaction creates a 'fitness saddle' which is shown more clearly on the rotated view of the same surface. This fitness saddle creates two optima indicated by the large dots located at "AB" and "ab". An algorithm that arrives at "ab" cannot cross the fitness saddle to "AB" without changing both features at once.

We then use this pairwise interdependency over a larger system of variables to define a system with modular interdependency. In Figure 1-5 below, we illustrate an example system exhibiting modular interdependency. The highly regular and symmetric construction used in this particular example enables us to see clearly the underlying structure that is important (though the regularities of this example are not, in fact, required). The dependency matrix on the left of the figure shows the strength of pairwise dependencies between eight problem variables, and the graph on the right depicts the same relationships. Sets of dependencies with equal strengths are indicated by appropriate borders in the table, and by the number of edges and spatial proximity of nodes in the graph. The fitness for a given configuration for the system of eight variables can be calculated from the sum of pairwise interactions (as per Figure 1-4) between each variable and every other variable weighted by the corresponding entry in the dependency matrix.

	1	2	3	4	5	6	7	8
1	-	16	4	4	1	1	1	1
2	16	-	4	4	1	1	1	1
3	4	4	-	16	1	1	1	1
4	4	4	16	-	1	1	1	1
5	1	1	1	1	-	16	4	4
6	1	1	1	1	16	-	4	4
7	1	1	1	1	4	4	-	16
8	1	1	1	1	4	4	16	-

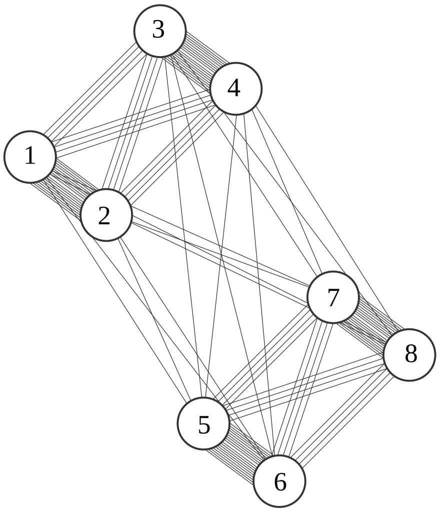
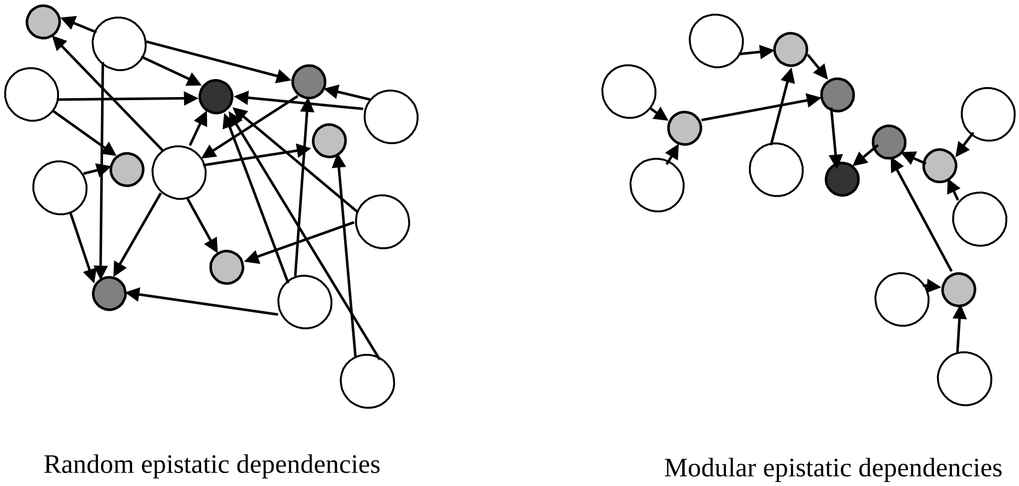


Figure 1-5: A system of variables exhibiting modular interdependency

The salient feature of this structure is that there are subsets of variables where the variables they contain are more strongly dependent on other variables within their own subset than they are on variables in other subsets. This creates a dependency matrix where the large values are diagonalisable. But importantly, the values of the matrix that are off the diagonal, representing the strength of dependencies of variables across modules, are non-zero. Clearly, the structure of dependency strengths in this example system are not just modular but also hierarchically modular - i.e. there are clusters and sub-clusters of more strongly interdependent variables. This potentially allows an adaptive mechanism that is capable of exploiting modularity to decompose the system recursively, or equivalently, to compose together subsolutions repeatedly. This kind of hierarchical decomposability is closely related to the notion of nearly decomposable systems discussed by Simon (1969).

This kind of system may be interpreted in different ways as appropriate for different domains. For example, we may interpret the graph on the right of Figure 1-5 as the network structure of a dynamical system (in the style of random Boolean networks, Kauffman 1993) where the future state of each variable is dependent on itself and the state of connected nodes. A biological example using a dynamical interpretation equates each variable with a gene, and interdependencies between variables may represent interactions that up-regulate or down-regulate one another. In this interpretation, the fitness of different configurations corresponds to the stability of attractors in the gene expression dynamics. In a different interpretation, we may use such a system to represent a modular epistasis structure in the genetic coding of phenotypic features - see Figure 1-6. The right hand side of Figure 1-6 indicates how the system of pairwise interdependencies acting between two large modules of Figure 1-5 can be abstracted into a single interdependency between two ‘aggregate’ variables, or understood as the interaction of the modules’ ‘aggregate effects’. This form of modular interdependency may be compactly represented by a simple recursive equation, and we will use this form in our simulation experiments.



Random epistatic dependencies

Modular epistatic dependencies

Figure 1-6: Alternate dependency structures.

A system of genes (large white nodes), contribute to a number of simple fitness-affecting features (light grey nodes), a smaller number of more complex features (mid grey), and, in these illustrations, just one feature that is a product of all genes. Genes may affect more than one feature (pleiotropy), shown by outgoing arrows. And several genes may have non-additive effects in the expression of a feature (epistasis), shown by incoming arrows. Left) In a random interdependency structure, the interdependency network of genes with features is arbitrary. Right) In a hierarchically modular interdependency structure, the expression of complex features can be understood as being dependent on (or resulting from the interaction of) the expression of a number of simpler features. A small number of complex features may thereby be indirectly dependent on epistatic interactions among a large number of genes.

To summarise this subsection, we suggest that the accretive view of evolution corresponds to the naïve class of optimisation methods that assume weak or few interdependencies between problem variables, and point out that this is not the only type of system that is amenable to efficient adaptation in principle. Specifically, in optimisation, problem domains that have a modular structure may be amenable to divide and conquer problem decomposition, and we suggest that compositional mechanisms may provide evolutionary adaptation of this form in certain circumstances. In this subsection we have also briefly described an example system exhibiting what we term ‘modular interdependency’ that we will use to exemplify the distinction in the adaptive capacity of accretive and compositional evolution. This class of system exhibits significant epistatic interdependency between all variables, but these dependencies have a clustered, or modular, structure. This kind of modularity defines formally an intuitive form of modularity, as described by Simon (1969), that has subsets of variables that are more tightly coupled internally than externally. In certain cases, such as that which we describe in our example, the optimal configuration for a module is dependent on the configuration of other modules but only in an aggregate way. This reduces the dimensionality of the search space for an algorithm that is able to exploit such modularity. In the next section we will outline the properties of this class of system that make it unevolvable for normal accretive evolutionary mechanisms.

1.4 The (un)evolvability of systems with modular interdependecy

Conventionally, evolution by natural selection is almost inseparable from the notion of accumulating slight modifications: “If it could be demonstrated that any complex organ existed which could not possibly have been formed by numerous, successive, slight modifications, my theory would absolutely break down”, (Darwin 1859). The notion of evolution as a gradual process that ascends local gradients in an adaptive landscape, has significantly shaped our conception of evolutionary change and evolutionary difficulty. For example, Wright (1931) states that “the central problem of evolution” is the (in)ability of a population to cross fitness-saddles between local optima in the fitness landscape, and Behe’s notion of ‘irreducible complexity’ (1996) is intended to characterise systems which are problematic for evolution because they do not admit evolution by accretion of small modifications. Accordingly, in the 140 years since Darwin a good deal of research effort has been directed at hypothesising smooth paths of slight successive modifications to account for problematic complex adaptations, and in identifying ‘missing links’, intermediate precursors of a complex adaptation.

Following the accretive assumptions of evolutionary change, there are several concepts of evolutionary difficulty that are pervasive in our understanding of evolutionary processes:

- **Ruggedness and the number of local optima.**

When the fitness contribution of one variable is dependent on the setting of another variable, the resultant fitness landscape may be rugged. Rugged landscapes create difficulty for accretive evolution. In particular, a configuration of a system where no small change in the system can produce a higher-fitness configuration is known as a ‘local optima’, or a local adaptive peak, and fitness landscape with many optima can prevent an accretive mechanism from finding high-fitness configurations.

- **Width of fitness saddles**

A second important concept of evolutionary difficulty concerns the width of fitness saddles - that is, the distance between a configuration and the nearest configuration with equal or higher fitness. For accretive mechanisms we may measure distance in terms of the number of variables that need to changed ‘in one go’ - this is the Hamming distance, for binary variables.

- **Irreducibility**

“Irreducible complexity” is defined by Behe (1996) as follows:

> *By irreducibly complex I mean a single system composed of several well-matched, interacting parts that contribute to the basic function, wherein the removal of any one of the parts causes the system to effectively cease functioning. An irreducibly complex system cannot be produced directly (that is, by continuously improving the initial function, which continues to work by the same mechanism) by slight, successive modifications of a precursor system, because any precursor to an irreducibly complex system that is missing a part is by definition nonfunctional. An irreducibly complex biological system, if there is such a thing, would be a powerful challenge to Darwinian evolution. (p. 39)*

These characteristics together tell us about whether there exists a path of monotonically increasing fitness from all configurations to the global optima, and more generally, the likelihood of finding such a path starting out from a random configuration. The supposed requirement of a ‘smooth path to the optimum’ typifies the normal understanding of evolvability, but our point is that this understanding of evolutionary difficulty is dependent on the assumption of accretive processes.

Let us now examine these notions of evolutionary difficulty for the hierarchical modular interdependency system we described above. First, the difficult pairwise interdependency acting between all variables creates a highly rugged fitness landscape with an exponential number of local optima. Sub-optima are created when variables within a module have maximised their fitness dependencies but inter-module dependencies are unresolved. Global optima correspond to configurations where all dependencies, within and across modules, are maximised. In the example system we will use in our simulations, there are two equally fit global optima which correspond to the two mutually exclusive ways to maximise all interdependencies (which in turn derive from the two mutually exclusive ways to resolve the pairwise interactions (see “AB” and “ab” in Figure 1-4). Second, the number of variables that need to be changed in order to escape from one local optimum to the next best configuration increases as larger modules are optimised. It thus becomes increasingly difficult for an accretive mechanism to cross fitness saddles between local optima in this problem class. Third, high fitness configurations of this system appear to be irreducibly complex because any small change in such a configuration is catastrophically deleterious.

An intuition for these properties of the system can be gained by examining a particular cross-section through the fitness landscape. In particular, a cross-section running from one global optimum to the other shown in Figure 5-2 indicates the large number of local optima, the separation of the two global optima shown at opposite extremes of the curve, and the width of fitness saddles in general. It also shows that the high fitness points are adjacent to low fitness points such that any small change in these configurations is catastrophically deleterious.

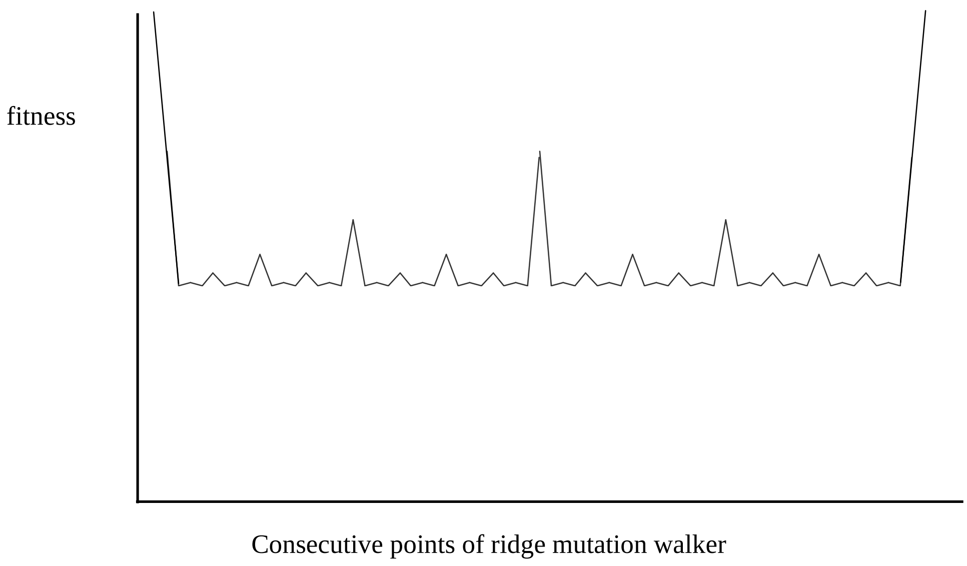


Figure 1-7: A particular cross-section through a fitness landscape derived from a system with modular interdependency.

See Section 5.1 for details of this figure.

Finally, there is no guaranteed path of monotonically increasing fitness leading to high-fitness configurations in this system. In conclusion, this class of system exhibits all of the properties listed above that are generally associated with evolutionary difficulty.

Given that the system of modular interdependency we have defined can be used to define an adaptive domain that exhibits all the normal features of evolutionary difficulty discussed above, it is not surprising that an accretive mechanism performs poorly. We prove that an accretive mechanism cannot be guaranteed to succeed in time less than exponential in N, the number of problem variables, in this class of system. This proof is formed by considering the expected time to escape from a high-fitness local optimum to either of the maximum fitness global optima. Our simulations illustrate the behaviour of accretive mechanisms by using mutation-only mechanisms on this fitness landscape.

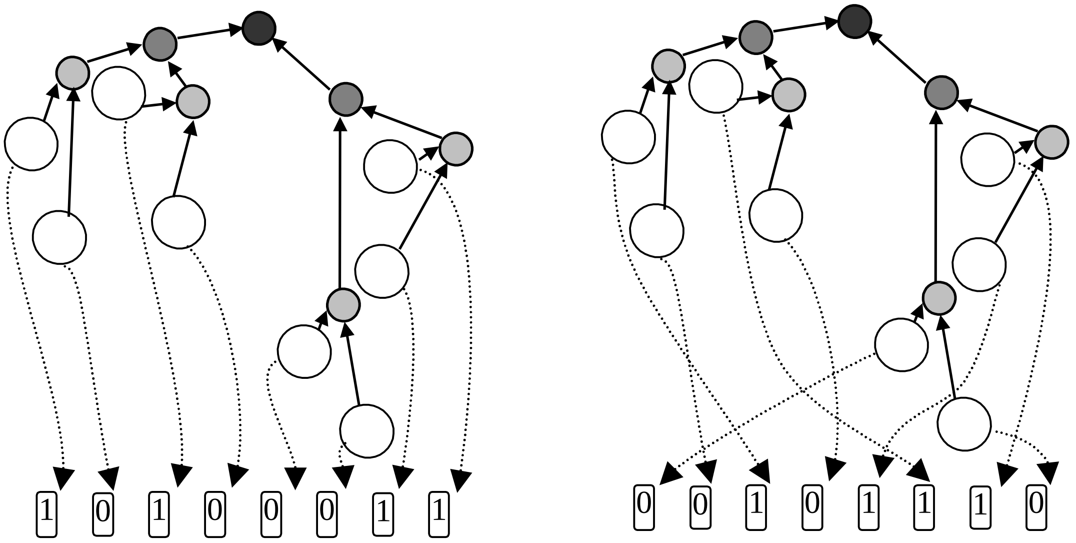
1.5 Conditions for the adaptive advantage of composition

We provide two highly abstract computational models to illustrate a sufficient set of mechanisms and conditions for compositional mechanisms that are able to provide efficient adaptation easily in systems such as the one we have described with hierarchical modular interdependency. The first model is based on sexual recombination (See Figure 1-2 left); the second is based on symbiotic encapsulation - the hierarchical encapsulation of symbiotic groups inspired by endosymbiosis and the major evolutionary transitions (See Figure 1-2 right).

1.5.1 Composition based on sexual recombination

We have suggested that compositional mechanisms may be able to exploit the modular structure of this class of problem domain and provide adaptation akin to divide and conquer problem decomposition. We can prove that this is possible in principle for a mechanism based on sexual recombination using crossover between two complementary strings. Although this exploits the *a priori* knowledge that the global optima are built from mutually exclusive complementary configurations, it serves to illustrate the path that a recombinative algorithm may follow if appropriate population diversity can be maintained. This analysis shows that a recombinative algorithm using sexual recombination can find the global optima in time polynomial in N given appropriate population diversity. We then show that a population-based model, a form of evolutionary algorithm using a particular mechanism of population subdivision, is able to quickly and reliably find the global optima in this problem class - in this case, without the *a priori* knowledge that the optima are complementary.

However, the ability of sexual recombination to search combinations of modules is strongly dependent on assumptions about ‘genetic linkage’ coming from the placement or ordering of genes on the chromosome (see Figure 1-8). If genes are not favourably ordered then, even though solutions to large modules may have been discovered in different members of the population, the amount of extra work (expected waiting time - represented by the term ‘c’ in Table 1-1) - is exponential in N, and this precludes efficient overall solution times.



‘Tight’ genetic linkage

Random genetic linkage

Figure 1-8: Alternate genetic linkage possibilities.

‘Genetic linkage’ refers to the tendency of alleles of different genes to be inherited together, or more exactly, to the deviation from allelic distributions that would be expected under ‘free recombination’ (6.4.1). In linear chromosomes, a system of genes must be mapped to positions on the chromosome, and the distance between genes in this ordering affects the likelihood that they will co-occur in an offspring produced by recombination with another string.[1] Left) Tight genetic linkage: Ideally, genes which are epistatically related, i.e. which contribute to the same features (solid arrows), will be next to each other on the chromosome. In this case, sexual recombination (in some circumstances) will be able to recombine subsystems effectively. Right) But without such a favourable situation, epistatically dependent subsets of genes may be arbitrarily positioned on the chromosome. We call this ‘random genetic linkage’. In this case, sexual recombination is unable to recombine subsystems effectively. In biological systems, the position of genes might be adaptively re-ordered, but in engineering domains where the interdependencies of variables are not known *a priori*, we cannot assume a favourable ordering of genes on the chromosome.

[1] In evolutionary computation, ‘genetic linkage’ is generally used to mean ‘genetic linkage caused by the positions of genes on a linear chromosome’ – biologically, the term is more general. The term ‘linkage’ is also sometimes used to mean epistatic dependence, but this is in principle a separate issue as we see in this figure. We will continue to use the term ‘tight linkage’ as a special term to refer specifically to an ordering of genes that corresponds favourably with the structure of epistatic dependencies between genes.

1.5.2 Composition based on symbiotic encapsulation

However, we are able to show that the ability of compositional mechanisms to exploit the structure of problems with modular interdependency is not necessarily dependent on assumptions about the favourable positioning of genes on the chromosome. To show this we use an integration mechanism based on symbiotic encapsulation. This is derived from a mechanism of symbiogenesis or endosymbiosis rather than sexual recombination. The salient distinction between sexual recombination and symbiotic encapsulation for our purposes is that whereas sexual recombination produces an offspring by taking, on average, half the genetic material from one parent and half the genetic material from a second parent, symbiotic encapsulation may simply take the sum of genetic material from both ‘parents’. Naturally, this is most productive when the expression of genetic material from the two parents is not mutually exclusive. Accordingly, our model of this mechanism allows for the possibility of entities that specify complementary characteristics rather than ‘overlapping’ or mutually exclusive genetic material.

The idea is that different specialist entities will evolve to cover different complementary parts of the problem domain (much like Cooperative Coevolution, Potter 1997). Then over time the dependencies between specialists will be resolved by forming unions between specialists (much like the ‘splice’ operator of the ‘Messy Genetic Algorithm’, Goldberg et al. 1989) to create a succession of new reproductive entities at a higher levels of organisation. This process repeats incrementally discovering larger alliances until a complete solution is found - a generalist that solves all modules and inter-module dependencies in the hierarchical problem structure.

This model must accommodate entities of increasing size as entities of many different levels of organisation are modelled together and thus requires a means to prevent large sub-optimal generalists from competitively excluding optimal specialists. This proved to be a critical aspect of the model that required us to step outside frameworks previously familiar in evolutionary computation. The model we develop, the “Symbiogenic Evolutionary Adaptation Model”, or “SEAM” (to invoke the notion of joins), uses a selection scheme that explicitly respects the *context sensitivity* of an entity’s fitness. That is, a proposed solution to a module may be good in one context and not in another, in the same way that the fitness of an organism is strongly dependent on environment. Selection in a varied environment may prevent competitive exclusion of one type of entity by another if the niches they inhabit are distinct, but in the case where one entity is fitter than another in all environments we may assume that the former will competitively exclude the latter. We model this explicitly in SEAM by retaining multi-dimensional measures of fitness coming from the performance of a specialist in different contexts, and applying selection conservatively using the dominance criterion. This provides an abstract form of automatic niching selection in an ecosystem of different species and replaces the normal single-dimensional fitness selection familiar in single-species models.

SEAM is able to properly identify and optimise the modules in this class of problem automatically and assemble them together to find larger modules repeatedly. It is thus able to quickly and reliably find globally optimal configurations of the problem variables in this class of hierarchical modular interdependency systems. We prove that if the discovery of conflicting module configurations does not become too unbalanced, the expected time to find globally optimal configurations in this class of system using SEAM is polynomial in N. The behaviour of SEAM is entirely insensitive to the ordering of genes on the chromosome. Thus we show that the algorithmic advantage of compositional mechanisms in this class of problem is not dependent on assumptions of favourable gene ordering.

In summary, the abstract models of sexual recombination and symbiotic encapsulation illustrate sufficient sets of conditions under which the adaptive capacity of compositional mechanisms is fundamentally different from that of accretive mechanisms. Specifically, we show that these mechanisms are able to exploit modular interdependency structure and provide efficient adaptation in cases where accretive mechanisms are not able to do so. The models that we use illustrate that the adaptive potential of sexual crossover is sensitive to gene ordering on the chromosome, but this is not a principled limitation of compositional mechanisms since a model based on symbiotic encapsulation is insensitive to gene ordering.

Simulation results from these models and algebraic analysis provide an existence proof for the thesis claim - i.e.

> Certain kinds of complex systems, considered unevolvable under normal accretive change, are, in principle and under certain circumstances, easily evolvable under compositional mechanisms.

1.6 Impact for evolutionary biology and evolutionary computation

Taken together, these models and results provide important insight into the broader questions we highlighted at the outset. Specifically, our simulations and analyses show that the existence of a succession of gradually changing proto-systems is not, in principle, essential to explain the evolution of a complex adaptation. Although some intuitions to this effect are already familiar in some fields, this dissertation enables us to clarify the fundamental algorithmic distinction between compositional and accretive mechanisms, to define the class of complex system that is at issue, to analyse the properties of such systems that make them difficult for accretive evolution, and provide a sufficient set of circumstances under which they are easily evolvable via compositional mechanisms.

Together, the models we provide illustrate the following points.

- **The 'accretive' conception of evolution can be misleading**

  The accretive model of evolution, and the corresponding conceptions of evolutionary difficulty, are inadequate in some circumstances. To wit, there is a class of adaptive landscape that is pathological for accretive evolution, exhibiting a high degree of ruggedness, wide fitness-saddles, and irreducibly complex optima, yet this class affords adaptation easily for alternate adaptive mechanisms.

- **The 'Compositional' model affords different adaptive capacities**

  We illustrate two mechanisms that enable the manipulation of relatively large feature complexes evolved in parallel lineages. These mechanisms permit adaptation in landscapes where accretive mechanisms do not. Specifically:

  - Sexual recombination, or crossover, under certain circumstances, provides adaptive change that is very different from the accumulation of mutational changes in a single population.

  - Mechanisms of symbiotic encapsulation, derived from symbiogenesis, encapsulating simple entities into a larger complex (as in several of the major evolutionary transitions), provides a second example. In terms of optimisation techniques, this mechanism provides a method of automatic module acquisition - encapsulating sub-solutions into a larger composite.

    - Both sexual recombination/crossover and our mechanisms of symbiotic encapsulation can be understood as forms of 'compositional evolution' — evolution based on variation arising from the assembly of pre-adapted sub-systems into a new arrangement.

- **The structure of dependencies is important**

  The structure (not just number or strength) of dependencies between problem variables, or epistatic dependencies between genes, can have significant effect on the evolvability of different adaptive mechanisms. In particular, a clustered dependency structure, exhibiting semi-independent subsets of interdependent variables, exemplifies the differences in the adaptive capacity of accretive and compositional mechanisms. Such a modular structure has a significant impact on evolvability, the likelihood of (near-)optimal solutions, and the feasibility of continued improvement in EAs, and on the likelihood of complex adaptations, and the feasibility of continued adaptation in natural evolution.

In general, the models provide an existence proof that certain kinds of complex systems, considered unevolvable under normal accretive change, are, in principle and under certain circumstances, easily evolvable under compositional mechanisms.

However, the existence of this certain kind of complex system, built on a modular epistasis structure, in natural adaptive domains, and the particular conditions required for adaptive composition events in nature, are a separate issue outside our claims. We provide these models as an illustration of the important potential that these mechanisms afford *in principle*, and to encourage analysis of evolutionary processes such as these that go beyond the usual accretive conception of evolutionary change. Nonetheless, the mechanisms and conditions that we explore are not arbitrary excursions into possible biologies. First, abstract computational models, such as these, are relevant to natural evolution at least in the sense that we demonstrate that it is possible for non-teleological adaptive processes, i.e. based only on fitness feedback, to behave in this manner. There is nothing in our models that is 'unevolutionary' in this sense. Second, mechanisms of sexual recombination, horizontal gene transfer, and symbiogenesis, for example, are biological fact. Third, some of the other concepts that we are exploring have a rich, albeit controversial, background in biological thought—and we discuss, without prejudice, the possibility that the necessary conditions may be present in natural evolutionary systems. In summary, our contribution to evolutionary biology lies in illustrating the adaptive consequences of such mechanisms *if* the necessary conditions are met in natural systems, and in providing a broader conceptual framework of possible adaptive processes by indicating parallels with different algorithmic classes.

In computational terms, we can be more specific about our central claim. i.e.

> Certain kinds of modular complex systems take time exponential in the size of the system[2] to evolve under accretive change. Yet these same systems may, in principle and under certain circumstances, be evolved in polynomial time under compositional mechanisms.

In addition to this claim and the broad points listed above, there are a number of points pertaining more directly to evolutionary algorithms/algorithmic theory. For example:

- **EAs can sometimes enable divide-and-conquer problem decomposition (as BBH suggests)**

  A variety of EA can, under certain circumstances, on a particular problem class, perform search more closely allied to divide-and-conquer problem decomposition than hill-climbing. Specifically, it is able to exploit the modular structure of a problem and combine sub-solutions to find a whole solution. This, we suggest, demonstrates the intuition of the Building Block Hypothesis for a particular class of algorithm and problem domain.

- **Crossover is not (always) equivalent to 'macro-mutation'.**

  A macro-mutation is simply a large mutation, a large random change, and some researchers suggest that the large genetic changes produced by crossover are nothing more than macro-mutations. But, the set of variants produced under crossover with other members of a population produces quite a different distribution from unbiased random change, and in particular, the distribution of variants by crossover is biased by prior selection on the population.

[2] Number of variables - e.g. primitive components, characteristics, or genes.

- **Mutation-difficult and GA-easy**

  There is a class of problem that is easy for a variety of GA with crossover, but pathologically difficult for a GA without crossover. We will show proofs for restricted cases in addition to empirical studies and reasoned argument for general cases. In short; certain kinds of modular complex systems take time exponential in the size of the system to evolve under mutation—Yet these same systems may, in principle and under certain circumstances, be evolved in polynomial time under crossover.

In addition to theoretic understanding, and better understanding of existing EA methods, we also provide some new algorithmic components for implementing artificial evolutionary algorithms.

For example, we develop a diversity maintenance technique that encourages a population to spread-out to cover different niches in an adaptive landscape; this can be understood as a method to automatically identify semi-independent sub-problems in an optimisation problem, as is demonstrated in the modular problem we use in our experiments. Furthermore, in the symbiotic encapsulation model, we also develop an encapsulation method that can be understood as a method for automatic module acquisition.

The abstract models we develop, applied to our synthetic modular problem domain, provide proof of concept for these algorithmic techniques, and the mechanisms of the adaptive processes we illustrate are readily implementable. However, the applicability of these principles in practical engineering optimisation is a separate issue outside our claims. We provide these models as an illustration of the important potential that these mechanisms afford *in principle*, and to encourage analysis of artificial evolutionary processes such as these that go beyond the usual accretive conception of evolutionary change implemented in normal evolutionary algorithms. However, (as was the case for the natural evolution counterparts), the artificial mechanisms and conditions that we explore are not arbitrary excursions into possible algorithms/problems. The algorithmic mechanisms we develop include some that are in common usage in artificial evolution practise, and address difficulties that are broadly accepted as important research issues in the field. Other issues involved in these models have a rich, albeit controversial, background in the theory of evolutionary algorithms.

1.7 Related issues

There are a number of other issues raised by these experiments and results. At this stage, we do not attempt to give full explanations. Our intent is to acknowledge the existence of important related issues and give some idea of the connections to related issues that the reader may have in mind. Chapters 2 and 3 introduce relevant background in evolutionary biology and evolutionary computation, and discuss those concepts that are necessary for grounding the main models that we will introduce in the body of the dissertation. The remaining issues will be left until discussion - Chapters 8 and 9.

1.7.1 Related issues in Evolutionary Biology

- **Shifting Balance Theory**

  Shifting Balance Theory, SBT, (Wright 1977), addresses the adaptation of a population when sub-divided into a number of subsets or ‘demes’. Wright suggests that different demes may converge on different ‘interaction systems’, sets of interacting alleles. We discuss the relationship of SBT to the evolution of modules in the mechanisms we illustrate, and the importance of mechanisms to promote or maintain diversity in a population.

- **Coevolutionary dynamics**

  In using a model that involves the multiple interacting lineages we necessarily develop a *coevolutionary* system. In contrast to the more common adversarial relationships seen in many coevolutionary models (e.g. Hillis 1992, Reynolds 1994, Pollack & Blair 1998), the coevolutionary relationships we are most interested in here occur between entities having complementary adaptive abilities (see Sigmund 1998). We devise a novel abstract model of ecosystem niching that permits different entities to diversify into different environments (each environment being created by neighbouring entities in the ecosystem). This segregates competition and avoids degenerate dynamics where one species competitively excludes all others.

- **Hierarchical selection**

  The notion of prior or parallel adaptation selecting for useful modules, and the idea of mechanisms that encapsulate a group of entities into a new whole, suggest that selection needs to act at more than one level - on primitive features and on modules, on modules and on individuals. Issues of group selection and hierarchical selection are poorly understood and highly controversial in evolutionary biology. The models we employ can be interpreted as implementing selection at different levels in a very abstract and subtle manner. Yet we are careful to consider the selfish interests of the parties involved in a symbiotic encapsulation, as required by a microscopic, 'Selfish Gene' style model (Dawkins 1976). Although our experiments do not attest to the biological validity of hierarchical selection, they do illustrate the fundamental adaptive impact that selection over several scales affords if present. This gives us good cause to examine the empirical questions in this subject, and suggests that some reinterpretation of common mechanisms such as sexual recombination may be informative.

- **Evolution of Cooperation**

  By proposing a coevolutionary model of co-adapting entities that are composed together by symbiogenic mechanisms, we necessarily enter the subject of evolved cooperation. However, the notion of whether a symbiotic pair cooperates because they are encapsulated, or whether they are encapsulated because they are cooperative, is not distinguished in our abstract models. However we make some preliminary observations about the relationship of the models we present to Selfish Gene theory.

- **Modularity in the organism and modularity in the adaptive domain**

  In this work we are primarily concerned with modularity in the adaptive domain. We suppose that, given appropriate variation mechanisms and conditions to exploit this structure, organisms can adapt to it more readily than is possible in a non-modular domain built on a random interdependency structure. This is related to modularity within an organism in two ways. 1) If we take the structure of gene dependencies as given, then if these gene dependencies are modular (see Figure 1-6), then the problem of finding fit configurations of alleles for these genes constitutes a modular problem domain. 2) The modularity of a problem domain is expected to induce modular internal structure in the entities that evolve in that domain. (There are several plausible reasons that may produce some degree of internal modularity: for example, reorganisation of epistatic linkage resulting from implicit selective pressure to increase robustness to genetic variation.)

1.7.2 Related issues in Evolutionary Computation

- **Modules are not the same as building blocks - they do not necessarily have tight-linkage**

  The modules in the problem class we define, and the modules that crossover and the symbiogenic mechanism are able to manipulate, are a different in important respects from Holland's notion of building blocks (1975).

  - For the cases where the GA with crossover can operate successfully, they are indeed 'short low-order schemata with above average fitness'. But this is not the definition that we find useful or accurate for 'modules'. Importantly, there is a sense in which the modules are meaningful entities regardless of which arrangement is used for the genes on the chromosome. In other words, the modules are meaningful entities even when they do not have short defining length—although crossover is admittedly unable to manipulate them when they have long defining length.
  - It is the more general sense of module, independent of defining length and gene ordering, that is successfully manipulated by the mechanism of encapsulation based on symbiogenesis.[3]

- **Modules do not have to be 'separable'**

  Importantly, whether we are talking about building blocks for the GA or the general kind of module for the encapsulation mechanism, it should not be assumed that the notion of module implies a problem that can be decomposed into *separable* sub-problems. This is a gross error. We will discuss how this assumption has been involved in the controversy around the BBH and the operation of the GA. A module, in the sense that is useful to us, is a subset of variables for which the interdependency interface to the remainder of the problem is low dimensional. That is, the state of the module, for the purposes of its interaction to the rest of the problem, can be adequately described with less unique states than the total number of possible internal states (we will define these terms later 4.3.1). There are additional constraints required in order for these modules to be

[3] Briefly, we use "module" to refer to a subset of variables, whereas a "schema" requires a configuration for that subset, and a "building block" is a short, low-order schema (Figure 3-5).

easily identifiable by a search algorithm. But, we stress that there may be strong non-linear interactions between modules, i.e. modules do not imply that a problem is *separable.* We use the word *decomposable*, in preference over separable, to indicate that a problem has modular structure.

- **Hierarchically decomposable problems**

  The interdependencies between modules that we describe in 1.3.2 show clusters of modules, creating the same kind of clustered interdependency between modules as there is between the primitive problem variables within each module. This allows for a hierarchically consistent notion of modules, and 'scale-invariant' problem structure. These notions of hierarchical modularity, and hierarchical decomposability, relate to Herbert Simon's ideas (Simon 1969), and the wealth of design/engineering and social systems he refers to. However, we show that a system of hierarchical interdependency does not depend on the idea of 'nearly-decomposable' modules with weak inter-module dependencies.

- **Validity of the Building Block Hypothesis**

  One of the primary EC motives for this thesis research is to clarify whether the EA can in some circumstances provide an algorithmic advantage akin to divide and conquer problem decomposition as the intuition of the Building Block Hypothesis suggests. With respect to the controversy around the Building Block Hypothesis, it is worth noting that in the endeavour to demonstrate problem decomposition by an EA, we are quite prepared to use non-standard varieties of EA—our objective is not to identify whether the 'standard' or 'simple' Genetic Algorithm (a particular form of evolutionary algorithm) provides this algorithmic capacity, but rather to determine whether some kind of evolutionary algorithm can. We also note that for any algorithm, the algorithmic advantage of the method is dependent on the class of problem to which it is applied (Wolpert and Macready 1997). Accordingly, it is not our intent to show that an EA, standard or otherwise, is advantageous in all problem classes. On the contrary, we invest a good deal of this dissertation identifying and illustrating the properties of a particular problem domain—specifically, the class of problems with modular interdependency. Thus, any failure of the Building Block Hypothesis to explain the operation of the simple GA in other problem domains is simply not relevant to our conclusions.

- **The Schema Theorem**

  A ‘schema’ is a partial specification of a candidate solution, or, more formally, the set of all candidate solutions that contain that partial specification. The Schema Theorem (Holland 1975) concerns the increase in copies of schemata with above average fitness in a population, and historically, it is supposedly responsible for the supply of building blocks for the Building Block Hypothesis. In the models we develop we find the notion of schemata, and schema combination, very useful, but we have no need to invoke or defend the Schema Theorem since our selection schemes are quite different. Our selection schemes also make conventional disruption analysis ineffective for our analytic purposes.

- **Automatic module acquisition**

  The model of symbiotic encapsulation we develop is conceptually allied to mechanisms of automatic module acquisition that explicitly ‘encapsulate’ subsets of features for subsequent re-use during the search process - for example, ‘Automatic module acquisition’ (Angeline & Pollack 1993), ‘automatically defined functions’ (Koza 1994), and ‘adaptive representation’ (Rosca 1997). However, our model operates on the simple string representation of chromosomes, and therefore assists us in seeing the utility of modules and their relationship to schemata more clearly.

- **Representational recoding**

  In enabling search to move from the combination of genes to the combination of schemata of increasingly higher-order, the encapsulation mechanism we illustrate constitutes a mechanism of representational recoding. That is, the representation of solution space is ‘re-coded’ at a higher level of organisation. Having been encapsulated as a module, alternate solutions to sub-problems may be treated as ‘alleles’ of higher-level variables within the problem, and the original high-dimensional problem is explicitly reduced to a lower-dimensional one.

- **Other methods using selection on parts and wholes**

  There are many other algorithmic methods related to notions of selection over modules/parts and wholes: classifier systems (compare Michigan and Pitt approaches) (Holland & Reitman 1978, De Jong 1988), SANE (Moriarty 1997), MIL (Juille 1999), immune systems (Dasgupta & Forrest 1999), and Ant Algorithms (Dorigo et al 1999). We outline the similarities of these methods with each other and our own methods, and provide a conceptual framework in which they can be understood more clearly.

- **Other models of interdependencies**

  Perhaps the best known model of genetic dependencies is Kauffman's 'NK-landscapes' (1993). When the number of variables involved in each dependency, K, is large they produce very rugged landscapes that are very difficult for accretive evolution. However, these landscapes have no structural modularity and are therefore also difficult for compositional evolution. We will see that two landscapes with the same *number* of dependencies can be very different in difficulty because of differences in the *structure* of those dependencies. We also discuss the interesting properties and inadequacies of 'NKC landscapes'.

- **Linkage Learning**

  Since the encapsulation mechanism we develop addresses the limitations of the tight-linkage requirements of crossover, we must also discuss other existing methods of 'linkage-learning' (see 3.4.2). However, it is not our goal to outperform alternate methods—and we do not compare with them in our experimental work. It is our intent to illustrate the concept of modules in a way that is independent of gene ordering, which our mechanism does. However, we do discuss some theoretical limitations associated with some linkage learning methods that assume modules are separable.

- **Diversity maintenance and Pareto dominance**

  The method of diversity maintenance that we develop for the encapsulation mechanism is not like other existing methods. It is based on the notion of Pareto dominance from multi-objective optimisation techniques. More specifically, we do not allow one schemata to competitively exclude another unless it *dominates* it. That is, for two schemata, A and B, if A is fitter in some context than B, and B is fitter in some other context than A, then we allow the two schemata to coexist. This model of ecosystem diversity has interesting connections to niche overlap in natural ecosystems. But computationally, it is an almost direct calculation of whether a schemata ‘could’ be useful for building higher-level systems, or alternatively, whether it is ‘good for nothing’ and can be discarded from future search.

- **Cooperative Coevolution**

  ‘Cooperative Coevolution’ (Potter & De Jong 1994, 2000, Potter 1997) explicitly uses the notions of divide and conquer problem decomposition and collective problem solving by coevolved parallel lineages. Our model basically applies this idea hierarchically.

- **‘No Free Lunch’**

  The NFL theorem (Wolpert and Macready 1997) basically says that ‘you cannot get something for nothing’—specifically, you cannot have an algorithm that is better than random search unless you also restrict the class of problems to which it is applied. In this dissertation we take considerable care to define the class of problems for which compositional mechanisms are superior - specifically, problems with modular interdependency.

1.8 Summary

1.8.1 Motives, research questions, and claims

Motives and goals:

- To foster and capitalise on two-way interdisciplinary exchange between Evolutionary Biology and Evolutionary Computation.
- To provide a conceptual framework that incorporates a broader view of evolutionary mechanisms than the adaptation of single population in a single niche. In particular, to understand the adaptive role, if any of symbiosis and symbiogenesis.
- To better understand the competence of the Genetic Algorithm, and in particular the utility, if any, of crossover and the Building Block Hypothesis.

Research questions:

- What are the limitations and capacities of alternate adaptive mechanisms, such as mutation, sexual recombination, and symbiogenesis in natural and artificial evolution?
- What are the consequences and affordances of alternate dependency structures in an adaptive domain, in particular, modular structures?

Claims:

- ‘Accretive evolution’, i.e. evolution based on the accretion of small successive modifications, is not the only possible mechanism of change in evolution, and accompanying concepts of evolutionary difficulty, are sometimes inadequate and misleading.
- ‘Compositional evolution’, i.e. evolution based on the assembly of sub-systems pre-adapted by prior adaptation in parallel lineages, is different from the accretive view of evolution in important respects.
- Certain kinds of complex systems, considered unevolvable under normal accretive change, are, in principle and under certain circumstances, easily evolvable under compositional mechanisms.

  More specifically, certain kinds of modular complex systems take time exponential in the size of the system to evolve under accretive change. Yet these same systems may, in principle and under certain circumstances, be evolved in polynomial time under compositional mechanisms.

1.8.2 Approach and outline of models

Approach

Our approach to investigating these research questions and supporting these claims is to develop conceptual argument and formal analyses supported by computational illustrations and mathematical proofs where possible.

Outline of models

Our main arguments concern three kinds of models, given below. Together these models illustrate important characteristics in different types of problems/landscapes, and the fundamental distinctions between accretive evolution and compositional evolution.

- **Models of the problem class/ adaptive landscape:**

  We define a problem class with a modular interdependency structure that creates modular sub-problems at many different scales. This problem class is designed to represent an interdependency structure that illustrates the distinction between the accretive and compositional mechanisms in the following models. The modules in the problem have significant interdependencies that prevent the problem being optimised by incrementally solving individual modules. The corresponding adaptive landscape typifies one that is difficult for accretive models of evolution: e.g. it is highly-rugged, and high-fitness points appear to be irreducibly complex. Nonetheless, the modular dependencies in the problem are, in principle, amenable to problem decomposition.

- **Models of accretive evolution**

  Here we use existing algorithms to illustrate the difficulty of our defined problem class for this type of adaptive mechanism. We use a mutation hill-climber, and various types of genetic algorithm, e.g. with crossover but with no genetic linkage (i.e. free recombination). We also use a large range of mutation rates to illustrate that large random changes do not help in this problem class. These experiments provide illustrations for reasoned argument and related mathematical proofs.

- **Models of compositional evolution**

  Here we use two particular models. First we use a Genetic Algorithm using sexual recombination/crossover. We show that this is able to solve problems with modular interdependency in limited circumstances. We then provide a novel model of evolutionary adaptation based on the encapsulation of coadapted simple entities into a new whole. This model is based on natural mechanisms of encapsulating genetic material from different entities evolved in parallel, in particular, symbiogenesis. This mechanism, together with supporting methods for maintaining appropriate population/ecosystem diversity, is able to find optimal fitness points in the fitness landscape easily. These experiments provide illustrations for reasoned argument and related mathematical proofs.

1.8.3 Contributions

The main contributions for evolutionary biology are:

- Clarifying the structure of epistatic dependencies that produce different kinds of evolutionary difficulty.
- Demonstrating the conceptual contrasts of accretive evolution and compositional evolution, broadening our interpretation of evolutionary processes.
- Providing a conceptual framework for evolution that begins to accommodate adaptive roles for symbiosis and symbiogenesis.
- Providing specific models for fragments of evolutionary processes such as genetic linkage, population/ecosystem diversity, population sub-division, and Shifting Balance Theory.

The main contributions for evolutionary computation are:

- Clarifying the structure of dependencies between variables in modular problem domains, in particular, hierarchically modular problems. (These are related to, but resolve important limitations of, ‘building block’ problems in GA theory).
- Demonstrating the conceptual contrast of hill-climbing and divide-and-conquer problem decomposition mechanisms in evolutionary algorithms. (This is related to, but less specific than, the notions of the Building Block Hypothesis).
- Providing a concrete illustration of mechanisms that permit automatic module acquisition and assembly in evolutionary algorithms, and a conceptual framework that accommodates cooperative coevolution and function optimisation; together addressing fundamental issues such as ‘representational re-coding’ and open-ended scalability in evolutionary search.

- Providing specific models for fragments of algorithmic methods such as methods for maintaining appropriate schema diversity, and the use of partially-specified genotypes to represent epistatic dependency in a way that is insensitive to gene ordering but affords appropriate recombination.

1.8.4 Scope

It is important to be clear about the scope and boundaries of this dissertation. The work in this dissertation is theoretic and conceptual. As indicated in the motives, approach, and contributions above, our studies address issues of principle, not empirical observations about natural or engineering domains or processes. Accordingly:

- Our claims do not purport that compositional mechanisms provide a general purpose problem solving algorithm. However, the thesis work is useful in expanding possible computational methods, and understanding existing computational methods. It also provides some specific algorithmic components, and specialised algorithms for the purposes of illustrating algorithmic possibilities.

- An empirical survey of general evolutionary methods and their performance in general problem classes is not required to support our claim. However, we will discuss some principled limitations and affordances of different algorithmic approaches when explanatorily useful.

- Our claims do not require that the problem class we define be equivalent to any particular problem class in engineering domains, or any particular adaptive landscape in natural domains. However, our models of problem classes are useful for expanding our notions of evolutionary difficulty, and problem difficulty in general. We also discuss principled limitations and affordances of different general problem classes.

- Conclusions derived from our abstract models should be transferred to natural evolution only with extremely careful qualification. Abstract computational models, such as those we use, are relevant to natural evolution at least in the sense that we demonstrate that it is possible for non-teleological adaptive processes, based only on fitness feedback, to behave in this manner. There is nothing in our models at this abstract level that is ‘unevolutionary’. Moreover, the experiments that we explore are not arbitrary excursions into possible biologies. Mechanisms of sexual recombination and symbiogenesis are biological fact. The other concepts that we are exploring have a rich, albeit controversial, background in biological thought—and we shall do our best to draw the relevant connections to the biological literature, and provide appropriate biological interpretations of models and results. In short, we do not claim the availability of the necessary conditions for compositional evolution in nature—this is an empirical matter. However, our models assist us in identifying what those conditions might be and provide a description of the features that may be examined in natural systems. In the meantime, the conceptual and theoretic principles of the thesis stand independently.

1.8.5 Dissertation structure

Overview of dissertation structure

In the following background chapters we introduce the relevant background in evolutionary biology and computation. The middle chapters (4, 5, 6 and 7) describe the models and experiments.

Chapter 4 describes a class of systems built on *modular interdependency*. This is designed to distinguish the adaptive capacities of accretive and compositional mechanisms in the other main chapters.

An obvious division of the mechanisms investigated might have been provided by dividing them into simulations of accretive mechanisms, and simulations of compositional mechanisms. However, it transpires that sexual recombination can behave either accretively or compositionally and does not therefore fit neatly into either category. Accordingly, we divided the simulations by three main classes of genetic operator- i.e.:

Chapter 5 - mutation (accretive),

Chapter 6 - sexual recombination (either accretive or compositional), and

Chapter 7 - symbiotic encapsulation (compositional).

The models themselves are quite simple: the fitness landscape can be defined in a simple equation, and each algorithm can be defined in a few lines of pseudocode. However, since the purpose of the models is an illustration of deeper conceptual issues, they require a fair amount of set-up in order to convey why the models are the way they are, and what the results do and do not mean.

In the discussion chapters we indicate the implications of the thesis work with related issues in the EB and EC literature.

Chapter-by-chapter Outline

Chapter 1 - **Accretive and compositional change in natural and artificial evolution**. Lays out the basic ideas and arguments and scope of the thesis.

Chapter 2 - **Accretion and Composition in Evolutionary Biology**. Describes the relevant biology that motivates the ideas of the thesis and the controversial concepts in evolutionary biology that are at issue.

Chapter 3 - **Accretion and Composition in Evolutionary Computation**. Describes the relevant models in artificial evolution and the controversial concepts that effect evolvability and the utility of evolutionary algorithms in contrast to other problem solving methods.

Chapter 4 - **Defining Modular Interdependency**. Describes the various models that have been proposed, both to capture the qualitative nature of landscape structures in natural evolution, and to exemplify the utility of evolutionary algorithms (or not). Clarifies the concepts of decomposability and separability in the interdependency of problem variables, and introduces a new problem class with hierarchically clustered modular structure.

Chapter 5 - **Mutation on Modular Interdependency**. Describes the main characteristics of the landscapes defined in the previous chapter, and illustrates the difficulty of this landscape for simple accretive mechanisms, specifically mutation.

Chapter 6 - **Sexual Recombination on Modular Interdependency**. Illustrates the operation and limitations of evolution using sexual recombination, represented by the Genetic Algorithm with crossover, on a problem with modular interdependency.

Chapter 7 - **Symbiotic Encapsulation on Modular Interdependency**. Introduces a complete, though simple, computational abstraction of symbiotic encapsulation, and illustrates its operation on a problem with modular interdependency.

Chapter 8 - **Implications for Evolutionary Biology**. Outlines the implications of these models and results for EB, and discusses related issues and future research.

Chapter 9 - **Implications for Evolutionary Computation**. Outlines the implications of these models and results for EC, and discusses related issues and future research.

Chapter 10 - **Summary and Conclusions**. Summarises the arguments, experiments and contributions of the dissertation, and concludes.

Chapter 2 - Accretion and Composition in Evolutionary Biology

In this, and the corresponding EC chapter that follows, we will use the following outline:

- First we will talk about accretive evolution, and corresponding notions of evolutionary difficulty.
- Then we introduce some mechanisms that step outside this standard model, towards mechanisms of compositional change. And, where available, we highlight corresponding notions of what might be easy or difficult for evolution under these mechanisms.

2.1 The accretive model

2.1.1 Successive slight modifications/ gradualism

Conventionally, evolution by natural selection is almost inseparable from the notion of accumulating slight modifications. This is evident from the beginning: "If it could be demonstrated that any complex organ existed which could not possibly have been formed by numerous, successive, slight modifications, my theory would absolutely break down", (Darwin 1859). In fact, this statement is "clearly invalid" (Gould 1982a, p.84) because if, somehow, large fitness-positive[4] changes are available then natural selection can act on them. Nonetheless the supposed necessity of gradualism is quite pervasive, as we will discuss.

The notion of slight modifications perhaps originates from observing character variation under artificial selection, and by observing spectra of small quantitative variation in natural organisms. Of course, observed slight modifications might not be representative of evolutionary changes in general, and even if they were, it does not categorically exclude the possibility of large changes. But there is a more fundamental reason to support this notion. Even before knowing the genetic basis of variation, the idea that

[4] Or, not appreciably fitness deleterious.

evolution must proceed by successive slight modifications, SSMs, follows from the idea that changes are random and unguided. That is, large random unguided changes are generally considered unlikely to be beneficial. This follows from the assumption that a small change from an adaptive peak will likely remain somewhere on the adaptive peak, whereas a large change will probably not - then, given that high fitness points are rare, a large change is more likely to arrive at a lower fitness configuration than the peak we started on. Accordingly, we arrive at the conclusion that it is likely to be the case that beneficial changes are more likely to be small than large.

When the genetic basis of variation is understood, the idea of SSMs is further reinforced. If genetic mutations are the source of variation, and these are small and rare, then SSM seems to be the only option. It is clear that we should separate genotypic variation from phenotypic variation—a small change in genotype does not necessarily mean a small change in phenotype. But, still the general concept of changes being unguided might suggest that even if a genetic mutation did induce a large phenotypic change, it would be unlikely to be beneficial. In short, large changes would have to rely on 'hopeful monsters' (Goldschmidt 1940, Gould 1982b).[5]

The assumption that evolutionary change must be based on slight modifications leads to the perception that evolutionary change as a whole may be characterised merely as the accumulation of successive slight modifications. This is the model we refer to as 'accretive evolution'.

In summary, the notion of accretive evolution, evolution via the accumulation of successive slight modifications, is supported in three ways: a) by the observation that organismic variation in phenotypic characters, both those occurring naturally and those occurring under artificial selection, are generally slight, b) by considering the *prima facie* likelihood of large random unguided variations being beneficial, and c) by the existence of mechanistic sources of small variations, i.e. mutation.

The notion of accretive evolution has become embedded in evolutionary thinking in many different ways. For example, as population genetics took shape (i.e. with Fisher, Haldane, and Wright in the 1930s), the

[5] It will be clear to the reader that this argument is ignoring the possibility that although the genetic mutation may be random, the large phenotypic consequences of a mutation are non-random. We will shortly discuss this and the possibility that large non-random phenotypic changes may also be 'guided' by prior selection, i.e. more likely to be adaptive than a random change of the same scale.

assumption of evolution by SSMs became entrenched. Goldschmidt’s theory of ‘saltations’[6] was rejected, and evolution by small changes was also supported in more subtle ways. Specifically, if we assume that evolutionary change involves small modifications then the only fitness effects that matter are those that can be accounted for by additive effects of small changes. Put another way, evolution by SSMs can only respond to the fitness effects of small changes. So, although the fitness effect of a large change might not be equal to the sum of fitness effects from the small changes of which it is composed, we need not consider the influence of this possibility on evolutionary trajectories. The upshot of this, is that population genetics models generally consider only additive effects of genes and generally ignore *epistasis* (which for now, we will just define as non-additive fitness effects of allele combinations).

This leads us to consider the main influence of accretion on evolutionary thinking that we want to discuss—the impact of accretive models on notions of evolutionary difficulty.

2.1.2 Evolutionary difficulty under accretion

Fitness landscapes

Wright (1967) introduced a highly pervasive tool in thinking about evolutionary processes, the *fitness landscape.* A fitness landscape, is a surface where the vertical position at each point is given by the fitness of a corresponding genotype, and the neighbourhood of points on the surface is given by the neighbourhood of corresponding genotypes under genetic variation. That is, genotypes that are adjacent under the operators of genetic change, correspond to adjacent points in the fitness landscape, and the height of each point is given by the fitness of the corresponding genotype.

[6] ‘saltation’: evolutionary change within a species originated by sudden, drastic changes that reorganize the whole genome. Although most such reoganizations are deleterious, a few of those “hopeful monsters” would be progenitors of new groups, Goldschmidt argued. (The viability of ‘hopeful monsters’ was later resurrected by Gould 1970, drawing on the observation that changes in phenotype are strongly non-random, as we shall briefly discuss.)

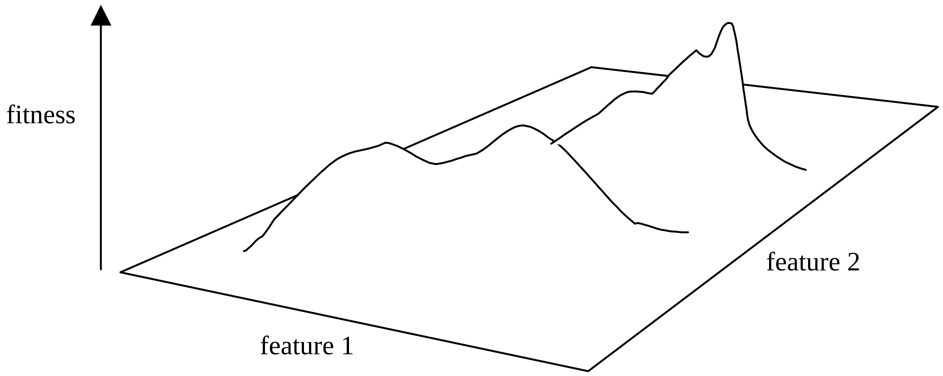


Figure 2-1: A fitness landscape.

In principle, the fitness landscape can be defined the same way regardless of the genetic operators involved (Jones 1995). For example, we could define adjacent points in the landscape using k-point mutation instead of single point mutation. In general, for the fitness landscape to be intuitive, we want it to be the case that the closer two points are to each other in the landscape, the higher the probability of transitioning between them under genetic variation. This is natural and intuitive when the distance metric is mutational distance (or Hamming distance—the sum of differences in all dimensions of change). We will discuss non-mutational neighbourhood metrics later, but the metric of small modifications (genetic or phenotypic) is intuitive and by far the most pervasive.

Armed with the fitness landscape, we can now understand evolutionary processes as a hill-climbing process. That is, evolutionary change can be seen to ascend local gradients in the fitness landscape by making small changes in the neighbourhood of the population mean, and ‘moving towards’, or selecting for, variants that are fitter or ‘higher’. Then, since evolutionary change allows movement to only local points on the landscape (by definition), evolution must ascend local gradients to arrive at local optima, local peaks, in the landscape.

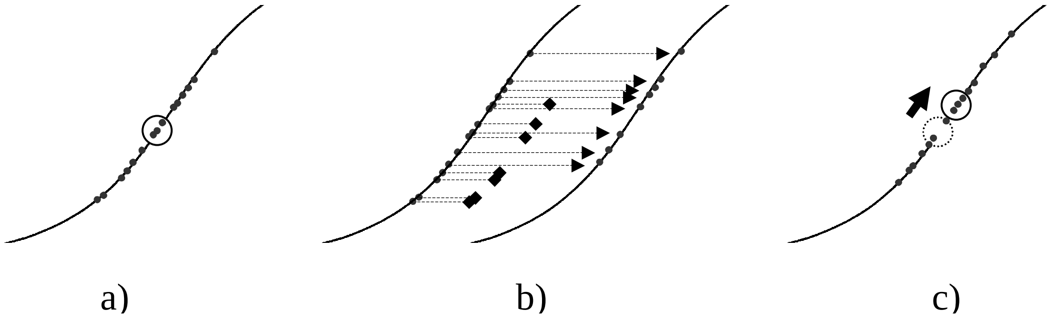


Figure 2-2: Population mean climbing a local fitness peak.

The curve represents the local fitness landscape, the points represent individuals of a population, the circle represents the approximate population mean position on the landscape. a) a population of individuals can be represented by its mean. b) selection removes less fit individuals but allows more fit individuals to 'pass through' c) the remaining individuals have a new population mean, as does the next generation of new individuals generated from them. In the process the population mean in (c) has climbed the local fitness gradient with respect to (a).

Epistasis, ruggedness, local optima and fitness saddles

The problem arises when features have epistatic interactions. That is, when the change in one feature changes the fitness effect of changes in another feature. Or at the genetic level, epistasis occurs when the allele at one locus changes the fitness effects of alleles at another locus. If genetic changes are epistatic then the corresponding fitness landscape is non-planar.

Intuitively, the *ruggedness* of a surface is the extent to which it deviates from a flat plane. Evolutionarily, the problem with ruggedness is that it may create *local optima*—local peaks on the landscape where no neighbouring points have higher-fitness. From this, one of the predominant notions of evolutionary difficulty follows.

> *the central problem of evolution .. is that of a trial and error mechanism by which the locus of a population may be carried across a saddle from one peak to another and perhaps higher one*.
>
> Wright, (1935) p.264.

Thus, the idea that evolution proceeds by the accumulation of slight modifications gives rise to the intuitive notion of a population climbing a local fitness gradient in a fitness landscape. And from here, the idea that the population might become trapped in a local optimum, a point that has no neighbours of higher fitness, creates a powerful notion of evolutionary difficulty. Specifically, we must explain how evolution is able to cross a 'fitness saddle', the region of depressed fitness between one local optimum and another.

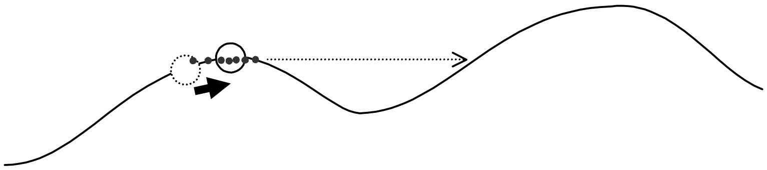


Figure 2-3: A fitness saddle.

Wright characterises evolutionary difficulty as the problem of crossing a fitness saddle. i.e. a population climbs a local gradient to the local fitness peak, and is unable to cross to higher fitness points on other fitness peaks.

We note that there are several alternative scenarios that could enable successful adaptation in problematic cases. For example, one possibility is that large amounts of genetic material may accumulate without being expressed in the phenotype thus allowing neutral evolution (Kimura 1983) and permitting variants to travel along neutral pathways (Huynen et al. 1996, Reidys et al. 1997) that 'tunnel' their way to fitter parts of feature space. This possibility enables a valley crossing by utilizing large amounts of random mutation that is unguided by selection, but unlike the hopeful monster, this need not occur in a single mutational event. (see Section 5.3).

Irreducible complexity

Another, less sophisticated yet still emotive, notion of evolutionary difficulty is also based on violating the assumption of accumulating small changes. "Irreducible complexity" is defined by Behe (1996) as follows:

> *By irreducibly complex I mean a single system composed of several well-matched, interacting parts that contribute to the basic function, wherein the removal of any one of the parts causes the system to effectively cease functioning. An irreducibly complex system cannot be produced directly (that is, by continuously improving the initial function, which continues to work by the same mechanism) by slight, successive modifications of a precursor system, because any precursor to an irreducibly complex system that is missing a part is by definition nonfunctional. An irreducibly complex biological system, if there is such a thing, would be a powerful challenge to Darwinian evolution.*
>
> *(p. 39)*

Behe then indicates that many biological systems are irreducibly complex at the molecular level. The notion of an irreducibly complex system sounds a lot like the notion of a designed system, such as the infamous watch, from Paley almost 200 years earlier (1802). But Behe incorporates the anti-Darwinian notion that not only is the system complex and specific in its structure, but that it also cannot be formed by accumulating small modifications. Behe is right that an irreducibly complex system is a problem for (accretive) evolution.

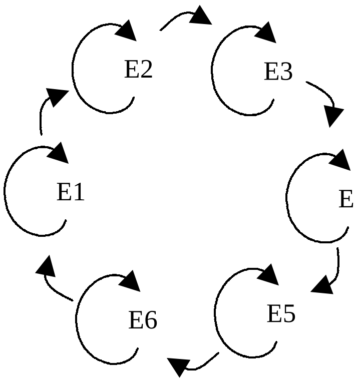


Figure 2-4: A hypothetical irreducibly complex system.

A system of autocatalytic enzymes forming a hypercycle. If any one enzyme is removed the whole cycle is broken.

Following the assumption that accretive evolution is the only possibility for natural evolution there has been, in the 140 years since Darwin, a great deal of research effort spent hypothesising about plausible fine-scale paths of monotonically increasing fitness approaching problematic adaptations (e.g. Dawkins 1996), (and similarly, a good deal of effort spent identifying systems that do not appear to have smooth paths of approach as typified by Behe).

This effort is expended with good cause. If there are paths of small changes approaching all adaptations in nature then there is no evolutionary difficulty. But we have taken pains to emphasise that the perceived need for such paths arises from the assumption of slight successive modifications.

2.1.3 Consequences of the accretive model

In summary, the assumption of accretion is embedded in many notions of evolutionary change and evolutionary difficulty. For example, the assumption of accretive change is embedded in…:

- …the assumption that there must always be some path of successive slight modifications conferring monotonically increasing fitness approaching any evolvable feature or adaptation.
- …the assumption that there is a meaningful neighbourhood metric on which a fitness landscape may be based, giving a meaningful notion of local adaptation and local hill-climbing for evolutionary change.
- …the idea that ruggedness in a fitness landscapes, and the existence of local optima, corresponds to evolutionary difficulty.
- …the idea that a population will, for the most part, be more-or-less converged around a local fitness peak, i.e. stuck on a local fitness optimum.
- …the belief that a system where the removal of any one part causes non-function (irreducibility) is difficult for evolution.
- …the assumption that, although epistasis is not necessarily absent, natural selection will act primarily on the additive effects of gene substitutions.
- …the assumption that large adaptive changes should be relegated to ‘hopeful monsters’.

All of these assumptions and notions of evolutionary change and evolutionary difficulty are based, directly or indirectly, on the assumption of accretive change.

2.2 Compositional mechanisms

Despite the pervasiveness of accretive notions in evolutionary thinking, it is clear to many that the assumption of successive slight modifications is a simplification. Perhaps the most obvious mechanism effecting large changes are mechanisms that enable large changes in phenotype from small changes in genotype. Other mechanisms enabling large modifications include mechanisms that re-use pre-adapted sections of chromosome, such as gene-duplication. But the mechanisms we want to address first are compositional mechanisms.

2.2.1 A spectrum of compositional mechanisms

There are many mechanisms that manipulate subsets of pre-adapted genetic material in nature:

- Sexual recombination combines genetic material from members of the same species.
- Natural hybridization may occur between very similar organisms.
- Allopolyploidy (having chromosome sets from different species) (Werth et al. 1985) generally occurs between related species.
- Horizontal or lateral gene transfer (Mazodier & Davies 1991, Smith et al. 1992, Davies 1996, Ochman et al. 2000) can occur between both similar and dissimilar organisms.
- Endosymbiosis can encapsulate the genetic material of wholly unrelated microbial species (Margulis 1993a).

Figure 2-5: Mechanisms of genetic composition

All of these mechanisms manipulate pre-adapted gene complexes between individuals and sometimes between species. In this list we begin to emphasise a spectrum of mechanisms: Specifically, the mechanisms near the beginning of this list re-use complexes that are pre-adapted in similar organisms (same species); whereas the mechanisms toward the end of the list may manipulate genetic material pre-adapted in relatively unrelated parallel lineages (different species).

The two mechanisms that we will examine in our models are taken from either end of the spectrum above. The first is sexual recombination. Although sex is usually considered to occur *within* a lineage, or at least within a population, sexual recombination manipulates feature complexes by exchanging them between *different* individuals. Indeed, to the extent that organisms are similar, sexual recombination between them has no effect, so it is the diversity of internal lineages within a population that makes sexual recombination productive (if it is).

The second mechanism we will examine subsequently is one which can combine gene complexes from unrelated parallel lineages, i.e. entirely different species. Specifically, we examine a mechanism of interspecific genetic integration based on Endosymbiosis, which we call 'symbiotic encapsulation'.

Margulis (1992) suggests that sexual recombination and endosymbiosis can be thought of as forms of the same process but with different degrees of relatedness between the entities involved: "symbiosis and meiotic sexuality entail the formation of new individuals that carry genes from more than a single parent. In organisms that develop from sexual fusion of cells from two parents, these parents share very recent common ancestors; partners in symbioses have more distant ancestors."

The following sections overview these two mechanisms.

2.2.2 Sexual recombination

To an optimistic adaptationalist, sexual recombination offers the possibility that it might take the good parts from two parent individuals and bring them together in the offspring by crossover (Figure 2-6). But not every adaptationalist is optimistic. In fact, the adaptive utility of sex, if any, is very controversial. We are not going to try and unravel the details of all the various arguments that have been proposed for and against the adaptive potential of sexual recombination in this dissertation. In particular, we are not going to address the benefit or deficit of sex to the individual - but rather the adaptive capacity of the population afforded by sexual recombination. In general, we will focus on the issues that we can address with the models that we will introduce later.

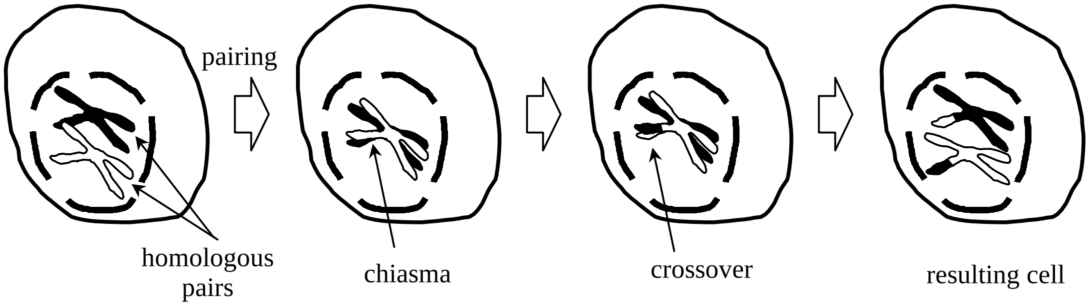


Figure 2-6: Crossover

Sexual recombination certainly has the potential to make new combinations of alleles by bringing together genes from two different donor individuals. However, it also has the potential to break-up combinations of alleles that were resident in either parent. The observation that allele combinations are not stable, not reliably reproducible, under sexual recombination is the motivating tenet of ‘the selfish gene’ (Dawkins 1976). Specifically, if gene complexes cannot be reliably reproduced because they are broken-up by sexual crossover, then the only unit remaining that is reliably reproduced is very short sections of chromosome—on the order of a gene. On the basis of this, it is concluded that all adaptive change in sexual populations is driven by the selfish interests of genes, not individuals, (because evolution is driven by the ‘selfish’ interests of whatever unit of reproduction is in operation).[7]

The break-up of allele combinations by sexual reproduction is another part of the picture in population genetics models that places focus on the additive effects of alleles. In a large panmictic[8] polymorphic[9] population “combinations [of alleles] are broken-up too rapidly to permit effective selection among interaction systems[10] (unless the loci are so closely linked that alleles at different loci behave almost as if alleles of each other).” And thus selection is “restricted to the net effects of alleles” (Wright 1977, p.470).

[7] Note that we, like Dawkins, are thus far assuming that the mechanism of reproduction is a given—as though it were not subject to evolutionary change. The evolution of sex is a more complicated subject altogether, and in a framework that acknowledges the possibility of alternate reproductive mechanisms, the units of reproduction are as much a *product* of the evolutionary process as they are the *actors* of the evolutionary process.

[8] ‘panmictic’, unstructured or freely mixed. Wright is introducing the assumptions behind ‘Shifting Balance Theory’, (which we will discuss later), where in contrast, he addresses the action of small sub-populations within a spatially sub-divided population.

[9] ‘polymorphic’, containing different alleles (in different individuals).

[10] Wright uses “interaction system” to refer to a set of coadapted interdependent alleles.

This introduces another important concept in sexual reproduction—'linkage'. 'Linkage' or 'genetic linkage' refers to the tendency of alleles of different genes to be inherited together, or more exactly, to the deviation from allelic distributions that would be expected under 'free recombination' (6.4.1). In linear chromosomes, the distance between genes on a chromosome affects the likelihood that they will co-occur in an offspring produced by recombination with another string. Wright observes in the above quote, in passing, that linkage *may* enable selection on sets of closely linked genes under sexual reproduction.

Thus far, we see that:

- Sexual recombination has the *potential* to create new combinations of alleles.
- Sexual recombination also has the potential to break-up combinations of alleles.
- Sexual recombination introduces units of reproduction smaller than the whole chromosome.
- These units may be as small as the individual gene, in which case a model need only consider the additive effects of alleles.
- Genetic linkage may enable selection on sets of genes if they are close together on the chromosome.

We emphasise the point that the potential of sexual recombination to manipulate gene complexes effectively, if at all, will be dependent on appropriate genetic linkage coming from the ordering of genes on the chromsome. This will be a significant feature in some of our models.

We also note that if a population is completely converged, sexual recombination offers no new combinations of alleles. In the limit, if all members of a population are identical (no polymorphism at any locus) then sexual recombination has no effect. Accordingly, if we are interested in the idea of bringing together new combinations of alleles, we need also to consider issues of population diversity. This will also be a significant feature in some of our models.

We acknowledge that the idea of sexual recombination bringing together new combinations of alleles, and at the same time, not breaking-up existing combinations of alleles, seems problematic. Nonetheless, we will explore the possibility that sexual recombination can, with favourable genetic linkage, enable effective adaptation by bringing together sets of alleles that have been coadapted in different individuals of a diverse population.

Finally, there are many theoretical studies on sexual recombination, its limitations and affordances that we have not mentioned. Often these assume no genetic linkage, or no-epistatic effects between genes, or simple models of population diversity (or lack thereof). These are not unreasonable assumptions in some cases, but there are alternative scenarios. In the models that follow, we step outside these assumptions. Accordingly, since we use different assumptions in our epistasis model, our linkage model, and our diversity model, our conclusions complement rather than contradict the models in the literature. Thereby, we are able to show specific effects of sexual recombination that are radically adaptive, whilst not contradicting existing arguments that are generally to the contrary.

2.2.3 Interspecific genetic integration

Horizontal gene transfer

The exchange of genetic material between members of different species is more widespread than might be imagined given the normal definition of species. Recent nucleotide sequencing indicates that processes of interspecific genetic transfer account for the origin of significant amounts of both cytoplasmic and nuclear DNA in microorganisms.

> *The most reasonable explanation…* [for anomalies in the phylogenetic tree] *is that the pattern of evolution is not as linear and treelike as Darwin imagined it. Although genes are passed vertically from generation to generation, this vertical inheritance is not the only important process that has affected the evolution of cells. Rampant operation of a different process—lateral, or horizontal, gene transfer—has also affected the course of that evolution profoundly. Such transfer involves the delivery of single genes, or whole suites of them, not from parent to offspring but across species barriers.* (Doolittle, 2000)

There is also evidence that lateral gene transfer is possible between multi-cellular organisms:

> “DNA elements have been observed transferring into and between higher organisms, such as plants, fungi and metazoans, including humans.” (Syvanen & Kado, 1998).

Lateral transfer may involve a gene or a suite of genes, but there are occasionally more radical processes of interspecific genetic integration - as follows.

Endosymbiosis and symbiogenesis

Endosymbiosis is the symbiosis (living together) of two organisms where one lives inside the other. In some cases of endosymbiosis, it is very difficult to establish the independent status of the parties involved. Indeed, the endosymbiosis of prokaryotes (bacteria) with one another has to a large extent become irreversible, i.e. the parties involved have subsequently become dependent on one another and cannot live independently. Such endosymbiosis accounts for the origin of various organelles, e.g. mitochondria and chloroplasts, in eukaryote cells. And there is some evidence to suggest that other organelles, such as flagella, also resulted from the physical inclusion of symbiotic bacteria (Margulis 1993b).

In some cases, the symbiotic entities remain as physiologically distinct ‘organelles’ within the larger ‘host’ cell — e.g. plastids such as mitochondria and chloroplasts (Figure 2-7). But there are also mechanisms, such as transposons, that can transfer genetic material from plastids to the nucleus or vice versa, implying that symbiotic origins may account for nuclear DNA as well as cytoplasmic DNA, as Goodnight’s data concords.

Such evidence suggests a more general process—‘symbiogenesis’, the origin of new species from the integration of symbionts. In this view, endosymbiosis is not simply responsible for the origin of eukaryote organelles, but for the origin of the eukaryotes themselves (including all plants and animals). The idea that new organisms could be created in such a manner was introduced as early as 1890,

> “Complex organisms might arise “through the unification of elementary organisms into colonies, and the transformation of the aggregate of them into an entity of a higher order…”” (Khakhina 1992, p.33, quoting, Famintsyn 1890).

The symbiotic origin of organisms was further developed by Merehzkovsky (1909), through studying lichens, algae, and fungi, and by Kozo-Polyansky (1921) (see Khakhina 1992). Meanwhile in western biological thought, Wallin (e.g. 1927) had similar thoughts about the origin of mitochondria, but at that time the idea was ridiculed and dismissed. The symbiogenic origin of eukaryote organelles now accepted, the concept of symbiogenesis is most well known in modern western science through Margulis (1970s).

Margulis describes the process as follows:

> *...different bacteria form consortia that, under ecological pressures, associate and undergo metabolic and genetic change such that their tightly integrated communities result in individuality at a more complex level of organization.* (Margulis 1995).

Moreover, evidence suggests that secondary endosymbiosis has also taken place; that is, endosymbiosis involving the inclusion of an endosymbiont (see Figure 2-7). This process of creating new species from symbiotic relationships is known as *symbiogenesis,* and is the basis of *Serial Endosymbiosis Theory* (Margulis 1970, Margulis 1993a). This theory proposes that endosymbiosis has been a recurring source of innovation in the evolution of complex cells.

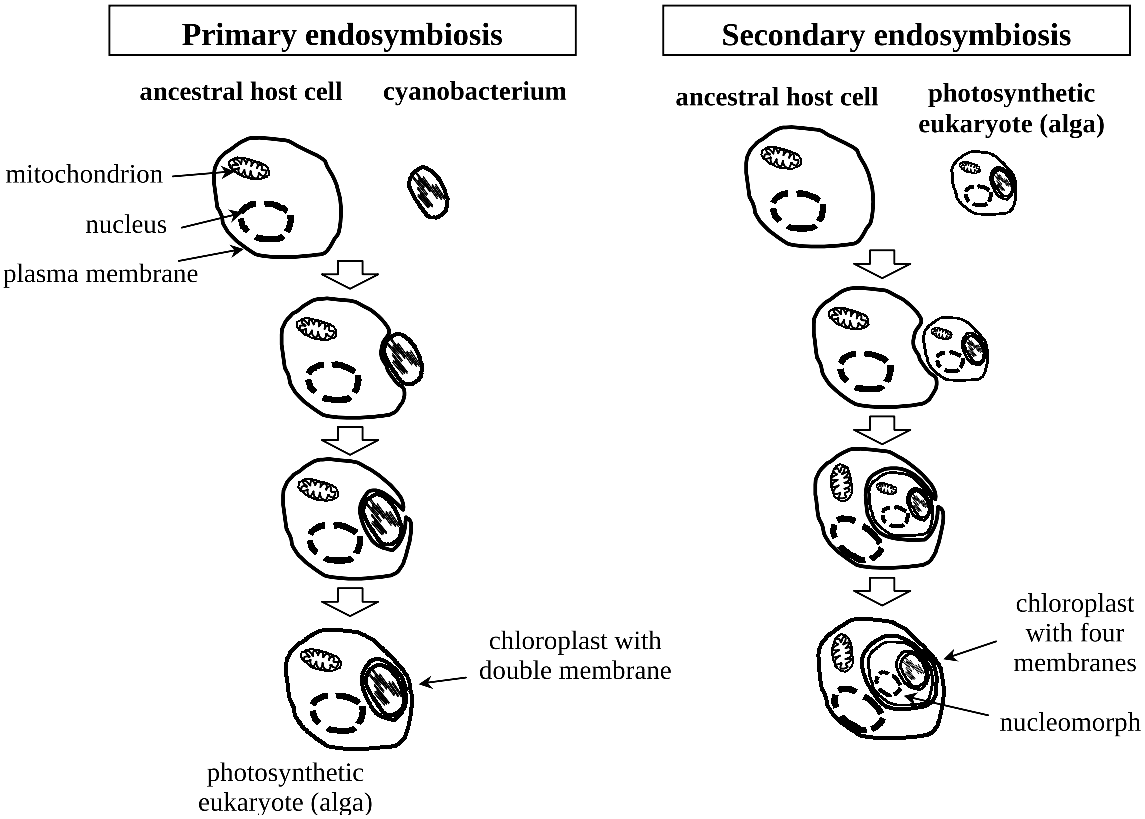


Figure 2-7: Endosymbiosis.

Left) Primary endosymbiosis involving host cell and photosynthetic cyanobacterium resulting in a photosynthetic eukaryote. Right) Secondary endosymbiosis involving host cell and eukaryote.

Accordingly, though its action may be rare in evolutionary history, the impact of symbiogenesis has been profound (eukaryotes include all plants and animals). Certainly, the genetic impact of symbiogenesis is, to say the least, unconventional. Symbiogenesis involves the encapsulation of genetic material from parallel lineages into a new whole—a new entity replicating as an individual. Certainly, from the point of view of either entity involved, it affords large non-random jumps in genotype (and phenotype) that introduce large amounts of pre-adapted DNA. And moreover, the idea that the organisms involved were symbionts before they were encapsulated, even if their relationship was not mutually beneficial, supports the idea that the combinations of organisms so created are favourably biased by prior co-adaptation.

The adaptive utility of symbiogenesis is under researched and poorly understood. Its potential to provide large adaptive jumps is of primary interest in this thesis. Moreover, the notions of serial endosymbiosis suggest an idea of repeated inclusion and hierarchical encapsulation, the consequences of which are even more poorly understood. Hierarchical encapsulation is one of the features common in the "the major transitions in evolution".

The Major Transitions in Evolution

The major evolutionary transitions (Buss 1987, Maynard Smith & Szathmary 1995, Michod 1999) involve the creation of new higher-level complexes of simpler entities. Summarised by Michod for example, they include the transitions "from individual genes to networks of genes, from gene networks to bacteria-like cells, from bacteria-like cells to eukaryotic cells with organelles, from cells to multicellular organisms, and from solitary organisms to societies". There are many good reasons to be interested in the evolutionary transitions: they challenge the Modern Synthesis preoccupation with the individual as the unit of selection, they involve the adoption of new modes of transmitting information, and they address fundamental questions about individuality, cooperation, fitness, and not least, the origins of life (Buss 1987, Maynard Smith & Szathmary 1995, Michod 1999).

In several of the transitions:

> *entities that were capable of independent replication before the transition can replicate only as part of a larger whole after it* (Maynard Smith & Szathmary 1995).

Although Maynard Smith and Szathmary identify several transitions which do not fit what they describe as "symbiosis followed by compartmentation and synchronised replication",[11] several of the transitions including the origin of eukaryotes from prokaryotes (Margulis 1993a & 1993b), and the origin of chromosomes from independent genes (Maynard Smith & Szathmary 1993), do involve the quite literal union of previously free-living entities into a new whole.

This form of change has some obvious contrasts with how we normally understand the mechanisms of neo-Darwinist evolution. The ordinary (non-transitional) view of evolutionary change involves the accumulation of relatively small random genetic variations within an entity, whereas innovation by composition involves the union of two different entities, each contributing relatively large amounts of genetic material to the new composite entity. Moreover, the result is quite different from a random variation applied to either donor entity since each contributes highly non-random genetic material by virtue of being independently pre-adapted as entities in their own right, if not in their symbiotic role.

2.2.4 Compositional evolution

We consider all the mechanisms listed above (Section 2.2.1) as compositional mechanisms. As mentioned we will focus on the first and last of these, sexual recombination and interspecific integration, because they represent the extremes of a spectrum: exchange of genetic material between very similar organisms (within a species) and exchange of genetic material between potentially unrelated species.

In the latter mechanism, we will use the idea that entities at one level of organisation are encapsulated into a new individual at a higher level of organisation. We shall refer to this mechanism as 'symbiotic encapsulation'. Symbiotic encapsulation is our term for the union of previously free-living, and independently adapted entities, into a new whole.

Symbiotic encapsulation might be seen as a *transitional* change in the sense of the major evolutionary transitions, whereas sexual recombination is an intra-transition form of composition.

In general, we define compositional mechanisms as follows:

[11] A notable exception in Michod's list above is the transition to multi-cellularity—although the existence of slime moulds that spend part of their lifecycle as colonies of single-celled organisms, and part as multi-cellular fruiting bodies, is of interest here.

- Compositional variation mechanisms combine together systems or subsystems of genetic material, or feature complexes, that have been semi-independently pre-adapted in parallel in different lineages.

And we include both sexual recombination between members of a diverse population (in some circumstances) and symbiotic encapsulation of members from different species as examples of compositional mechanisms.

2.2.5 Is gradualism necessary and/or sufficient?

Our interest in compositional mechanisms is as a source of large adaptive changes, in contrast to the accretive model of successive slight modifications. This possibility, and the possible contrast with gradualism, is recognised in the evolutionary biology literature (e.g. "Symbiosis as a Source of Evolutionary Innovation" (Margulis & Fester 1991). However, there is some disagreement about whether symbiotic encapsulation contradicts the supposed need for gradualism, and sufficiency of gradualism, in evolutionary change.

For example, Ridley (1985), having specifically mentioned symbiosis as a possible source of increases in complexity, states that, "The Darwinian denies (as he must) that the complex co-adaptation arose in a single chance event." (p.35) - "In all these cases, including symbiosis, complex organs have been built up in small stages." (p.41). So each adaptation "…appears at first impossible to build up in a series of small but advantageous stages. But in fact they probably were." (p.41). And thus Ridley concludes that "Complex adaptations must have evolved by the natural selection of a large number of small mutations over a long period of time." (p.42).

Similarly, on the subject of the apparent contradiction with Darwinian change, Maynard Smith (1991) states: "Symbiosis may give rise rather suddenly to evolutionary novelty; it is therefore seen as presenting a challenge to Darwinian gradualism. I think this is to misunderstand the reason why Darwin was a gradualist: essentially, it was because the origin of a complex adaptation would be miraculous." But, Maynard Smith points out that in light of mechanisms that exchange genetic material across lineages, we see that a complex adaptation acquired in such a manner is not 'miraculous', and - "There is, therefore, no contradiction between Darwin's belief that complex adaptation arise by the natural selection of numerous intermediates, and the possibility that new evolutionary potentialities may arise suddenly if genetic material that has been programmed by selection in different ancestral lineages is brought together by symbiosis.” (Maynard Smith 1991, p.37)

However, let us take a moment to break apart this reasoning, because there are some points that need to be clarified. First, as Ridley says, the *components* (or ‘organs’) that are brought together in a compositional event may well have been evolved gradually, or in any case, are the result of prior adaptation and are not ‘pulled out of thin air’, as it were. Thus, in agreement with Maynard Smith, there are no ‘miraculous’ large changes in effect in compositional events. But we should be careful with what we conclude from this observation. Although, there are no *miraculous* large changes - there *are* large changes.

What does this say about the necessity and sufficiency of gradualism in evolutionary change? If the *components* were evolved gradually then it appears that gradualism is necessary in order to explain the evolution of the new entity. But is gradualism sufficient? Consider the new entity rather than the components - it should be clear that, even if the components were evolved gradually, the new entity was not. Specifically, there need not necessarily be a path of small changes of monotonically increasing fitness approaching the configuration of features exhibited in this new entity. If there is not, then gradualism is not sufficient to explain its evolution - and even if there is such a path, no such path was followed in this historical case.

A more fundamental question arises from these considerations: specifically - Is it in principle possible that an entity may be evolved despite the fact that it *could not* be evolved only by accumulating small mutations?

It might appear that this is not possible. Consider a new entity AB that happens to have been composed together from two independently evolved components A and B. We might suppose that if A is evolvable from some start point, through appropriate intermediates, to a fully formed A, and likewise B is evolvable from some start point, through appropriate intermediates, to a fully formed B, then surely, even if AB was contingently the result of composition, AB *could* in principle have been evolved from nothing through appropriate intermediates to a full-formed AB without using adaptation of the parts in parallel lineages and without composition. However, we will show that this is not necessarily the case.

Specifically, we will show that it may be the case that although A and B are each independently evolvable gradually, AB is not evolvable gradually as a whole. This will be the case when A and B correspond to *interdependent modules* in the sense that we define in Chapter 4. As we will show, in systems with modular interdependency, each module is independently evolvable but the two modules cannot be evolved together accretively because of dependencies acting between the modules.[12]

Accordingly, we conduct simulations and analysis for non-compositional, purely accretive mechanisms, and show that accretive mechanisms are not sufficient to provide complex adaptations in an adaptive landscape arising from this kind of epistatic dependency. Moreover, for the sake of making a point, we also show that gradualism is not even necessary in this kind of landscape. Specifically, although the components that are composed together are the result of prior adaptation (and are not therefore miraculous), they need not necessarily be the result of accretive changes, but may themselves be the result of selection on prior compositional events. This notion of 'composition all the way down' (in the style of 'serial endosymbiosis theory' Margulis 1993) - as if there were no variational changes except compositional ones - is used in our simulation models to make the point: specifically, that in principle, there is a class of adaptive changes for which gradualism is neither necessary nor sufficient.

However, we by no means intend to imply that accretive change is not actually in effect in natural scenarios - of course it is. Yet, we will discuss the issue, outside our central claim, of whether accretive change is in principle possible of affording continued open-ended adaptive innovation, and contrast this with the possibility of continued innovation in the presence of compositional mechanisms.

2.3 Other related mechanisms

Mechanisms such as ontogenic processes and gene duplication provide a different source of large changes. The distinction between these mechanisms and compositional mechanisms is based on where the bias for

[12] We should qualify that 'can be evolved' and 'cannot be evolved' is more exactly understood as 'is easy to evolve' and 'is hard to evolve' - but, in the framework of combinatorial optimisation, these terms are not as subjective as they might appear. Specifically, in the former case the expected time to evolve the adaptation is polynomial in the size (number of elementary variables) of the adaptation, and in the latter, the expected time to evolve the adaptation is exponential in the size of the adaptation.

large changes comes from. Specifically, whether it comes from prior adaptation within the same lineage or from prior adaptation in parallel lineages. This section discusses the implications of these two possibilities and the consequences of our choice to focus on the latter.

2.3.1 Large phenotypic changes from small genotypic changes: ontogenesis

Ontogenic mechanisms, ‘translating’ the genotype into the phenotype, may permit large non-random changes in more than one way. For example, a small genotypic change may cause a complex of features to be expressed more than once in the phenotype, as in an additional repetition of a body segment. They may also allow subsequent modifications in the genotype to produce changes in all phenotypic repeats of a body segment, for example. “Small changes early in embryology accumulate through growth to yield profound differences among adults...Indeed, if we do not invoke discontinuous change by small alterations in rates of development, I do not see how most major evolutionary transitions can be accomplished at all.” (Gould 1982b)

There are perhaps other ways in which ontogenic processes may assist in increasing the viability of large modifications. For example, we might imagine a process of ‘ontogenic repair’—a local modification process that adjusts the physiological characteristics of a developing organism towards those that are more likely to be viable. Such ontogenic mechanisms may help ‘protect’ against non-viable variations and allow large changes that are viable with higher probability than large random changes. “Monsters may be hopeful because the regulative properties of development tend to channel perturbations along viable (if discontinuous) routes.” (Gould 1982a p.89). Some claim that genetic repair is biased so as to prevent deleterious changes (Smuts (1926) termed this “internal selection”, see also Arthur 1997).

Assuming that ontogenic processes are, at least to some extent, subject to adaptation and not entirely the result of pre-determined biological causation, then such favourable ontogenic processes would presumably be selected for if they should occur.

Other processes potentially involved in large non-random changes include *exaptation* (Gould & Vrba 1982). Exaptation refers to cases where a collection of features adapted for some purpose is co-opted for some other purpose or function. With respect to the function of interest, a large set of phenotypic features is introduced simultaneously. And since this set of features has been subject to prior selection (albeit for a different function) it is a non-random set of features. The utility of this scenario requires that the feature of interest be variationally close to a structure that, although functionally different, is also well-adapted for some purpose.

2.3.2 Mechanisms for large genetic changes within a lineage: Gene duplication

Any genetic mechanism that *re-uses* a subset of genetic material that has been pre-adapted by prior adaptation potentially enables a large non-random change.

Gene-duplication (Ohno 1970) provides a case in point. Genetic errors in replication can produce repeated segments of genetic material. Such mechanisms are responsible for the observation that many genes appear to be derived from similar sequences, i.e. they are duplicated and subsequently modified by mutation. The new section of chromosome so introduced, are of course quite different from random genetic material because they are sequences that have been arrived at through prior adaptation. Of course, they may cause some disruption, but it is easy to see that they may permit a large non-random change that has a higher likelihood of being non-deleterious than a random change would have.

Other genetic mechanisms manipulating sections of chromosome within a single chromosome include translocation and inversion.

2.3.3 Large changes within a lineage and self-similarity

Mechanisms like gene duplication have some further similarities with ontogenic mechanisms of large changes. That is, they both utilise biases coming from prior adaptation within a single lineage. It can also be seen that this bias is largely related to the repetition or duplication of feature complexes. This introduces a notion of repeated modules and the implication that the adaptive advantage of these mechanisms, if any, may depend on those modules being similar or at least being derived from similar beginnings. Put another way, this implies that these mechanisms will be useful when the adaptive domain is such that *self-similarity* in subsets of features is a useful heuristic.

2.3.4 Within a lineage vs between parallel lineages

In contrast to the mechanisms discussed above, mechanisms that exchange subsets of genetic material *between parallel lineages* (outlined previously), do not necessarily involve repetition or duplication of feature complexes, and do not invoke any implication that the newly introduced features are similar to existing features. Rather, from the point of view of each of the entities involved, the features introduced from a different lineage are not a rearrangement or duplication of *something it already has*, but *something new*.

In this thesis, we focus on mechanisms manipulating subsets of genetic material pre-adapted in parallel lineages. Accordingly, we will not pursue further mechanisms of ontogenic change or genetic duplication—mechanisms utilising bias from repeated or duplicated structure. We make this choice for three main reasons:

1) We wanted to illustrate the possibility of large, adaptive changes with as few assumptions about the problem domain as possible. In particular, we discovered that there are means by which large adaptive changes may occur that do not rely on the heuristic of self-similarity between modules in the problem domain.

2) The sexual exchange of genetic material between members of a species does not involve duplicating sections of chromosome, and we wanted to understand how, if at all, sexual recombination may allow large changes.

3) Mechanisms combining genetic material from parallel lineages are involved in several of the 'Major Transitions in evolution'.

We note that many natural complex systems do involve self-similar structure (Mandelbrot 1982), so this may be a very good heuristic for adaptive bias, and these processes also warrant further investigation (e.g. Hornby 2002)—but we choose not to focus on this issue in this thesis. We list them here for contrast and to explain our chosen focus.

2.4 Evolutionary difficulty under composition

In this section we discuss what we know about what is easy and what is hard for compositional mechanisms in evolution. We intend to move towards an understanding of what kind of complex adaptations are evolvable in compositional evolution that are unevolvable under accretive mechanisms.

2.4.1 Transferring concepts of accretive difficulty (where possible)

Let us examine some of the concepts of evolutionary difficulty previously outlined for accretive evolution.

Fitness landscapes under compositional mechanisms

We mentioned earlier that, in principle, a fitness landscape could be defined using whatever neighbourhood metric is given by the genetic variation operators (Jones 1995). One might argue then, that in the new neighbourhood metric given by compositional variation operators like sexual crossover, the evolutionary process may still be described as one that moves the population mean up the local fitness gradient—selecting among neighbouring variants based on fitness—but where ‘neighbours’ are defined using a different distance metric a different neighbourhood metric from that defined by mutation.

This is reasonable and useful in some circumstances. However, there are some problems with this interpretation. Specifically, the way in which a genotype can be changed under crossover is dependent on the composition of other members in the population. Since the make-up of the population is altered over evolutionary time, the neighbourhood metric is not constant. Accordingly, the idea of moving around in a fitness landscape is confusing, and most of the intuition of the landscape metaphor is lost, because the proximity of points in the landscape is constantly changed by the movement of the population.

An alternative is to discuss the movement of the population in the space of possible populations (e.g. Nix & Vose 1992, Wright & Rowe 2001). For a given operator, even a population-sensitive operator, the neighbourhood metric is then constant. That is, the new population configurations that can be reached from a given population configuration is fixed. However, this is quite a radical departure from the intuitive notion of a fitness landscape. First, each point in the landscape now represents a population, not a genotype. Second, what should the height at each point be now? In other words, what is the fitness of a population? If the population is diverse (and to the extent that it is not diverse, the landscape of a population with sexual crossover degenerates to the mutational neighbourhood) then taking an average of the fitnesses for the members of the population loses a lot of information. And besides, it is not clear that this appropriately describes the gradient that a population would follow.

Another possibility is to consider the neighbourhood between pairs of individuals (Jones 1995, Gitchoff &. Wagner 1996). Again, it is not clear what the fitness of a pair is or whether the movement of a population can be understood from the neighbourhood of pairs to one another under crossover. However, in restricted circumstances, this approach can be very useful (6.6.3).

In the meantime, we conclude that the notion of a fitness landscape does not easily accommodate variation operators that are population-sensitive such as sexual crossover.[13] However, we feel that the spatial intuition behind fitness landscapes is too powerful to discard completely. One partial resolution is as follows. We suggest that the mutational neighbourhood metric provides a natural default metric for the fitness landscape. In the neighbourhood defined by this metric, variations under crossover are seen as jumps. Thus the movement of the population mean cannot be described by local hill-climbing. However, the landscape still has some useful intuitions. For example, the size of the jump enabled by a crossover event is proportional to the polymorphism between the two parents. That is, by the introduction of new genetic material from Parent-2, the offspring of Parent-1 may be distant from Parent-1.[14] Also, we can see that since crossover cannot introduce alleles that are not already present in one parent or the other, the offspring of a crossover event must lie somewhere in the hypervolume whose opposite corners are defined by the parents. Accordingly, we suggest that the fitness landscape defined by the mutational distance metric is still useful in intuiting the action of crossover in some respects.

Additionally, our intent in this thesis is in large part to compare the operation of mutation and recombination and accordingly it will be useful to use the fitness landscape based on the mutational neighbourhood metric as one of those that we examine in detail.

[13] Although, in some of our analysis of recombination (see 6.5) we will consider properties (such as the absence of local optima) of the fitness landscape defined by the recombinative neighbourhood metric by using ‘invariants’ of the population.

[14] We find it useful to imagine the crossover operation on a pair of strings, C(A,B), as a ‘curried’ function on one string, C_A(B), describing the transformations that may occur between B and its offspring produced by ‘crossing with A’. (see discussion of multiple inheritance and single parent inheritance in Section 2.5.1).

Mechanisms like symbiotic encapsulation are even more difficult to accommodate into the notion of a population climbing the local gradient of a fitness landscape. For example:

- Mechanisms of encapsulation involve changes in the number of features exhibited by an entity, or changes in the number of genes specified by an entity. Accordingly, the space of genotypic possibilities does not have a fixed number of dimensions.

- Since the genotype of two different entities may not overlap completely, if at all, in the set of genes they specify, it is reasonable to suppose that they exist in different fitness landscapes. In what landscape then, would their composite exist?

- The notion of symbiotic encapsulation, by introducing the notion of symbiotic relationships, makes explicit the fact that the fitness of one organism is significantly dependent on the biotic (as well as abiotic) environment in which it is placed. In short, the fitness on an entity is undefined in isolation (Lewontin 1982, 1983), and if we want to discuss models involving symbiotic encapsulation this cannot be ignored.

These considerations challenge the utility of discussing a fitness landscape under encapsulation mechanisms. But, let us see what utility, if any, can be recovered.

- Let us first consider the union of all features, all genes, that an entity or any possible entities may exhibit. Let us take this to be a large, but fixed, dimensional space. We suggest, for the same reasons given above, that the mutational neighbourhood metric provides a useful default for a fitness landscape in this space.

- Any entity that specifies some value for all features, or allele for all genes, can be at a unique position in this space. But in general, entities will exhibit some subset of these features, and accordingly, they will occupy a hypervolume in this space.[15] The encapsulation of two entities is then properly defined as the intersection of these two volumes.

[15] This corresponds to the notion of *schema* in GA theory, as we will discuss.

- We also see that, at any one time, any particular entity is resident at a particular location in this volume of possibilities and has a particular fitness, even though its own genotype does not uniquely define that position. That is, its environment, biotic and abiotic, defines a context in which its own features reside, and this context is required to define its specific location in the volume.

When talking about symbiotic encapsulation, we will continue to talk about fitness landscapes, and in particular, we will continue to use the mutational neighbourhood metric as the default. But the landscape must be interpreted as described above, and the notion of a population mean climbing the local gradient must be discarded.

However, ruggedness and local optima produced by epistasis still provide meaningful concepts of evolutionary difficulty for accretive evolution in this landscape. And thus it provides a way to think about some of the differences between the adaptive capacity of accretive and compositional mechanisms.

Crossing fitness saddles

The introduction of a large subset of genetic material into an organism constitutes a large ‘jump’ in genotype. A large jump in genotype may conceivably enable an adaptive process to cross a fitness saddle (in the original, mutation-based landscape). But on the other hand, it could just take it somewhere random. We have argued that, by virtue of the fact that the new material introduced has been pre-adapted in a parallel lineage, the large changes afforded by compositional mechanisms may be more likely to produce adaptive change than a large undirected mutation. However, the hope that such a change might take us across a fitness saddle seems overly optimistic. If we are to understand how large jumps enabled by compositional mechanisms enable movement in the fitness landscape then we will need to understand the adaptive relationships between one local peak and another. We will need to understand the structure of the fitness landscape in a way that incorporates non-local relationships.

Irreducible complexity

The definition of irreducible complexity that Behe gives (see 2.1.2) assumes explicitly that modifications must be small: “the removal of any one of the parts causes the system to effectively cease functioning”. Behe is assuming that if the removal of a small part is a catastrophically deleterious, then evolution of that system as a whole is problematic: this in turn is based on the assumption that the only way to arrive at the whole system must be by the addition of a small part to a ‘proto-system’ that is independently viable. Since compositional mechanisms may introduce large parts, or many small parts, simultaneously, they may in principle enable the evolution of irreducibly complex systems, by Behe’s definition.

However, the idea of irreducibility could be defined more generally. For example, we might say that a system is irreducible if the largest subset of the system that is independently viable is distant from the whole system by many variational steps. This would be a problem for compositional mechanisms (as well as accretive mechanisms).

So what really matters is whether there are any subparts of the system that are independently viable, and whether we can get from them to the system of interest in a small number of variational operations. Behe assumes that if the largest viable subsystem is more than a small modification away from the final system then it is unevolvable. However, if it is a large modification away, it may still be evolvable *if* the large modification can be described as the addition of a subsystem that is available in some other individual. On the face of it, this seems unlikely, but if that subsystem is itself independently viable then the idea of combining two sub-systems to make the final system is not inconceivable.

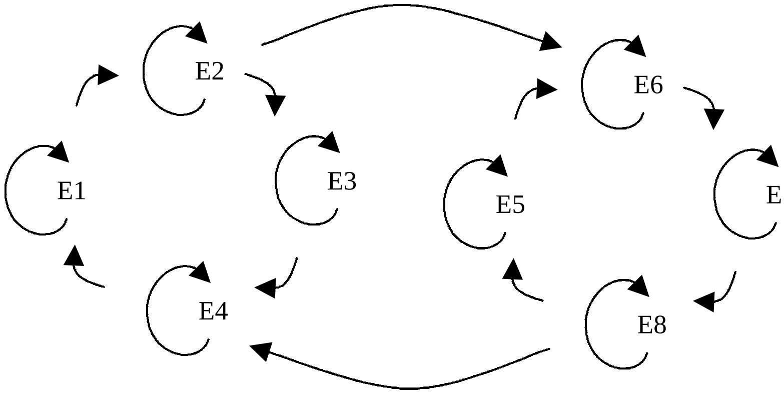


Figure 2-8: Hypercycles with multi-level structure.

A system that appears to be irreducible by Behe’s definition (i.e. the system function is broken you remove any one part) but nonetheless, it may be composed of useful/functional subsystems. A system of autocatalytic enzymes form two hypercycles, with catalytic interaction between the two hypercycles. If any one part of the system is removed then its subsystem collapses and therefore the entire system breaks. However, it is nonetheless plausible that each subsystem is stable independently, and the whole system could be created from the assembly of the two subsystems in one variational step. Thus, under compositional

> mechanisms, such a modular system is still conceivably evolvable, though apparently irreducible by Behe’s definition.

The revised notion of (ir)reducibility introduces the idea that a system might be composed of a small number of large, relatively self-contained, subsystems. This emphasizes the need for us to discuss and define ‘modular’ systems. And the idea of compositional mechanisms that assemble together pre-adapted feature complexes suggests mechanisms that can manipulate modules.

We are going to discuss modularity at length in Chapter 4. For now, let us just note that modularity is going to be an important part of the models that we will examine. Importantly, these modules are not separable sub-systems having no interaction; the interaction between modules will be important in understanding what the manipulation of modules under compositional mechanisms can enable that large mutation rates under accretion cannot. We also stress that the modularity we are going to discuss does not depend on notions of modules that are self-similar.

2.5 Discussion

The issues discussed above suggest that we should broaden our perspectives of evolutionary processes in several respects, covered in the following subsections.

2.5.1 Single inheritance and multiple-inheritance

Non-random changes, directed by *prior* selection

It should be clear that the mechanisms we discussed above are capable of producing large modifications (in phenotype in some cases, and in genotype in others). The important issue that we are moving towards, however, is whether such changes are likely to be adaptive, and more specifically, whether they are more likely to be adaptive than random changes.

In order to talk about the possibility of non-random changes, we need to define what we mean by random changes. Before doing that, it is useful to choose a definition of a random distribution of strings (or configurations of variables). A common default definition is a distribution of strings where all alleles at all loci are equally represented, and in fact all combinations of alleles at different loci are equally represented.

This is a uniform distribution of strings in the space of possible strings based on the metric of Hamming distance.

Now a random *change* necessarily requires the notion of a point of origination. For our purposes, this is a ‘parent’ string. Then a natural definition for a distribution of random changes is a uniform distribution of strings in Hamming space around the parent. We may then picture a random change as the addition of a vector (to the parent vector) drawn with uniform probability from a distribution of vectors that have no net directional component. More specifically, the set of possible offspring that may occur a given distance from the parent occur with equal probability (and uniformly distributed in direction).

This notion of random changes matches exactly with our normal notion of mutational change[16] - specifically, a mechanism that introduces random genetic material to a string with a uniform distribution over the loci and uniform distribution over the alleles of each locus. One-point, k-point, and per-locus methods of mutation are all valid examples - though they will have different distributions over the allowable distances that an offspring may be from the parent.

Now clearly, the changes produced by recombination (or composition) are not like this. Specifically, the genetic material of possible offspring is partly biased by the material of the first parent and partly biased by the material of the second parent. From the point of view of each parent, the new genetic material that is introduced into the chromosome of the offspring is not (necessarily) random. Specifically, in mutational change, genetic material is either inherited from one parent or is introduced stochastically, and in recombination, genetic material is inherited from two parents. Thus, in any case where the population from which the second parent is drawn is non-random, the changes between each parent and the offspring are also non-random.[17]

[16] Other definitions of random strings, and random changes may be derived from different neighbourhood metrics and different notions of uniformity in the space of possible strings. For our purposes, we may as well define random change to be whatever kind of changes mutation produces. But the definitions given here reflect quite natural meanings for ‘random string’ and ‘random change’.

[17] This is simply a consequence of defining a ‘change’ to be a movement from a single point of origination. Note that we could, if we wished, define a change with respect to two strings - for example, a random change might then be one drawn uniformly from the possible results of recombination between two parents - in which case, mutational change would be ‘non-random’. The point is that …

This is a very trivial observation - but it is worth clarifying: non-random variation is not ‘unevolutionary’ in any way. Now, immediately we must clarify that recombination only produces a non-random distribution of variants when the population from which the parents are drawn is non-random. Thus in the absence of selection, neither repeated mutation nor repeated recombination produce consistently directed change. However, when selection *is* applied this does not affect the distribution of changes produced by mutation - but it does affect the distribution of variants produced by recombination. Under recombination, it is not only the selection that acts on the variants produced at some given generation that directs evolutionary change, but also the biases in the distribution of variants produced coming from *prior* selection on the population.

The context sensitivity of genetic material

By the reasoning above, mechanisms that exchange subsets of genetic material between individuals are a source of large, non-random genetic changes. The next obvious question is: When will this kind of bias, coming from recombinative variation, be better than the ‘unbiased’ variation of mutation?

Of course, this is not a new question, or one that is easy to answer. But the above reasoning provides some intuition that may be useful. Specifically, we have been thinking about both mutational and recombinative variation in terms of a distribution of variants produced using inheritance from at least one parent. The difference between these mechanisms is that under recombinative change some of the new genetic material that arrives in the offspring (i.e. that which was not inherited from the first parent) comes from a second parent, whereas under mutation all the new genetic material in the offspring is random. Note that if both parents of a recombinative offspring have been subject to exactly the same selection, then the new genetic material introduced from the second parent is not likely to be significantly different in its adaptive affordances from that which is inherited from the first parent. In contrast, if the two parents have been subject to different selective pressures, the following notion arises:

> To the extent that pre-adapted genetic material, when re-located to a different genetic context (e.g. a different individual), is more likely to be adaptive than random genetic material placed in that

whichever operation we use to define random change - mutational change and recombinative change are different - and they cannot both be random in the same terms.

> context, mechanisms that exchange sections of genetic material from existing individuals will produce a distribution of offspring that are fitter on average than the distribution produced by random mutational changes.

If, on the other hand, genetic material adapted in one context is, when placed in some other context, no fitter on average than random genetic material, then mechanisms that exchange sections of genetic material from existing individuals will not be better than random mutational changes. The advantage of compositional mechanisms may thus be said to depend on the extent to which genetic material is 'context sensitive' (or perhaps, more exactly, 'context *in*-sensitive').

It does not seem unreasonable in some cases to suppose that genetic material that is adaptive in one context, a protein-encoding gene, for example, will have a better chance of being adaptive in some other context than random genetic material. If the gene still transcribes into the same protein, then for sure, the protein may be useful in some contexts and not in others, but in contrast, nearly all completely random sequences of genetic material do not transcribe to a protein at all. But ultimately, the context insensitivity of genetic material is an empirical matter.

In this thesis, we will discuss and define adaptive landscapes with limited context-sensitivity - which we call 'modular interdependency'. This will help us in understanding when such compositional genetic mechanisms do and do not provide variation that is fitter on average than mutational change.

2.5.2 The rest of the genotype as a 'backdrop' vs important epistasis

In order to study the selection effects for one allele in comparison to another allele of the same gene we might consider the 'net effect' of each allele as Wright suggests. However, if there is any epistasis between this gene and other genes then the fitness contribution of each allele will be different in different contexts. In single-loci population genetics models, which are not uncommon, epistatic effects are ignored and the remainder of the genome is treated as though it were a static backdrop against which the alleles in question are evolved.

As Wagner (1997) states, the absence of epistasis in population genetics models does not come about because population geneticists believe that there are no epistatic effects between genes. It is because they believe that selection cannot act on non-additive effects. At least, this is the case for mutation, for example. Indeed, Wright's conception of evolutionary difficulty is based on local optima created by epistatic effects. In fact, it is precisely because evolution by small modifications operates on additive effects that epistasis forms the basis of evolutionary difficulty in the accretive model. Population genetics models that do address epistatic interactions often model the epistatic interaction of only two loci.

Now, by changing our assumptions about the ability of variational mechanisms to manipulate subsets of genetic material (rather than individual genes), two things must change:

- Epistatic interactions with the rest of the genotype can no longer be treated as a 'backdrop' and we must address them explicitly,
- Our notions of evolutionary difficulty must scale-up—instead of pair-wise epistatic interactions between individual loci, we need to address non-additive interactions between larger subsets of genetic material.

In short, by modelling genetic mechanisms that manipulate modules, we will have to consider epistatic interactions within and between modules (of all scales that are manipulated) and selection on modules. Next we introduce an important additional piece of background on this topic.

2.5.3 Selection on interaction systems: Shifting Balance Theory

Wright considered selection on modules to be vitally important and he suggested mechanisms to enable it. Wright's 'Shifting Balance Theory' (SBT) describes a mechanism to enable the evolution of "interaction systems", i.e. sets of coadapted alleles.

Shifting Balance Theory (SBT) describes the adaptation of a population when sub-divided into a number of semi-isolated 'demes'. Wright suggests that the cumulative effect of a sub-divided population is quite different from the effect of adaptation in a single panmictic (unstructured) population. Specifically, sub-division permits different sub-populations to differentiate by genetic drift (non-adaptive genetic change arising from stochastic selection processes in small populations) and arrive at different genetic combinations (see also Barton 1993, Slatkin 1981, Wakeley 2000). This parallel exploration of many demes increases the possibility that at least one sub-population may discover a higher adaptive peak, and subsequent migration of the superior variants to other demes permits the whole population to shift to the superior peak. Alternatively, a few demes may each discover different high fitness adaptive peaks, and the mating of migrants from these sub-types may result in a superior integration of their characteristics.

Wright thus compares two possibilities for the ‘progress’ of evolution: a) selection among mutations, b) selection among ‘interaction systems’. In a large panmictic polymorphic population “combinations [of alleles] are broken-up too rapidly to permit effective selection among interaction systems (unless the loci are so closely linked that alleles at different loci behave almost as if alleles of each other).” And thus selection is “restricted to the net effects of alleles” (Wright 1977, p.470). Accordingly, in the panmictic case, “progress of mass selection under a constant set of conditions is limited by the extremely rare occurrence of novel favourable mutations.”

But, although a favourable mutation might be rare, “The fixation of a gene by mass selection tends, however, to be followed by a succession of favourable modifiers that build up a favourable interaction system”. So, Wright suggests that finding fit combinations of alleles, ‘interaction systems’, (together with the occasional new mutation) is more likely than the occurrence of mutations whose net effect is positive. Then, since a panmictic population provides no means to select among interaction systems, Wright employs the notions of sub-division and drift to enable localized adaptation where each deme can discover and focus on a different interaction system. Thus Wright states that “inferior interaction systems are replaced by superior ones” if a population is “divided into small populations, sufficiently isolated to permit wide stochastic deviations in numerous loci but not so isolated as to prevent excess diffusion from those centers that happen to have acquired the most adaptive interaction systems…” (p. 471).

In SBT we see some important features that will appear in our own models:

- Selection on interaction systems (modules)
- Semi-independent, parallel adaptation of many subsets of genetic material. (An emphasis on diversity.)

- Significantly localized selection (i.e. competition is predominantly between members within a subset).
- Significant among-subset variance in the effect of an allele/feature. Different demes contain different genetic backgrounds and therefore, through epistatic effects, produce different effects on a given allele.
- Possible 'assembly' of "complex adaptations whose constituent parts arise via peak shifts in different demes" (Coyne et al. 2000).

Wright’s work on SBT provides an example of a population genetics model that explicitly addresses the evolution of interaction systems or modules, the importance of mechanisms that can enable selection on interaction systems, and finally the idea that the rest of the genotype cannot be treated as a static backdrop to the adaptation of an individual locus.

2.5.4 Other species as a ‘backdrop’ vs important interspecific relationships

When approaching the task of modelling a system as complex as natural evolution it is necessary to make many simplifications. A common simplification is to examine the adaptation of a single species as though it were isolate - as though the adaptation of other species were a kind of ‘backdrop’ merely defining the niche for the species of interest. Although changes in the biotic environment may affect the niche of the species in question, and may in some cases produce quite tightly coupled co-adaptive dynamics, there is a deeply rooted assumption that interspecific relationships are not an integral part of evolutionary mechanisms - at least in the following sense. Changes in the genetic make-up of other populations may affect what is selected for in the given population, but they do not affect the genetic make-up of the given population directly. Specifically, it seems reasonable to study the evolution of a population as though it were independent from other populations because the mechanisms of genetic variation, specifically mutation and sexual recombination, act only within a population not across populations.

What then are we to make of events that involve the integration of genetic material across populations (2.2.3)? The existence of horizontal gene transfer, for example, means that modelling the evolution of a single species in isolation is not easily justified. But further, since several of the major evolutionary transitions involve the quite literal union of previously free-living entities into a new whole, clearly, evolutionary models addressing the adaptation of a single population are not capturing the whole process.

The phenomenon we focus on in this thesis cannot be properly understood with a single-species single-niche model. By looking explicitly at multi-species multi-niche models, many assumptions need to be revised. For example, competitive exclusion is not the only possible dynamic, and convergence of the population/ecosystem to a single type is not the only stable state.

2.5.5 Intragenomic dependencies and intergenomic dependencies

The previous subsections highlight the need to consider the interactions between one locus and another and between one organism and another. In short, they suggest that we must develop a model of both intragenomic interactions and intergenomic interactions.

But further, in models of symbiotic encapsulation it will be clear that these categories will change. That is, when two entities reside in parallel lineages, any fitness dependencies between them are intergenomic. But if those two entities should become encapsulated into a new organism then those same interactions are, for all intents and purposes, now intragenomic.

Accordingly, it will be important that the model of genetic dependencies that we develop be able to encompass both intra- and inter- genomic dependencies in a unified fashion.

2.6 Summary

This chapter has defined the distinction between accretive and compositional mechanisms of change. We discussed the normal (accretive) notions of evolutionary difficulty: e.g. saddle crossing and irreducibility, and their dependence on the assumption of small successive variations. We discussed a spectrum of compositional mechanisms and identified sexual recombination and symbiotic encapsulation to represent this spectrum for our upcoming models. Finally, we suggest that these mechanisms are not just a different kind of accretive change but that they cause us to revise important assumptions regarding for example, the meaning of a fitness landscape and the movement of a population on it, selection on modules, and the role of intragenomic and intergenomic interactions.

Chapter 3 - Accretion and Composition in Evolutionary Computation

In this, as in the previous EB chapter, we will use the following outline:

- First we will talk about accretive processes in artificial evolution, and corresponding notions of evolutionary difficulty. This starts with an overview of simple EA models.
- Then we introduce some mechanisms that step outside this standard model, towards mechanisms of compositional change. And, where available, we highlight corresponding notions of what might be easy or hard for artificial evolution under these mechanisms.

3.1 Evolutionary Algorithms

3.1.1 Evolutionary algorithm basics

Evolutionary Computation is, in general, the field encompassing computational techniques based on, or inspired by, evolution. In practise, the term is essentially synonymous with the class of problem solving and optimisation techniques known as Evolutionary Algorithms (EAs), (see De Jong 2002). Evolutionary Algorithms, EAs, are a form of computational optimisation or design technique based loosely on Darwinian evolution (see De Jong 2002 for an overview). Any attempt to overview the components of an EA that are common to all varieties will not do justice to all emphases. But, without being specific to procedural details, an EA generally has the following components.

The components of a simple EA	The biological metaphor/inspiration
An EA maintains a (large) number of candidate solutions or designs for a problem.	A population of individuals.
All candidate solutions are evaluated with respect to some quality or performance criterion.	Every individual has a “fitness”.
Those candidate solutions that are higher quality are used for generating new solutions with higher probability than those that are low quality. Low quality solutions may be discarded.	Differential reproduction. Differential survival.
New candidate solutions are created by random variations of the existing candidate solutions. And those variations may propagate to future candidate solutions.	Descent with modification. Heritable variation.

Table 3-1: Components of an evolutionary algorithm

The basic form of the algorithm uses repeated cycles of *evaluation*, *selection*, and *reproduction with variation* to shape a population of candidate solutions for some problem. A number of random candidate solutions, individuals, are generated to initialise the population. Evaluation is the process of assessing the value or quality of a candidate solution with respect to some objective measure of quality or performance. Selection is applied to (probabilistically) focus search around those solutions that are superior. Reproduction produces new candidate solutions by random variations of selected individuals. Successive repetitions of evaluation, selection, and reproduction (with variation) change the composition of the population by promoting candidate solutions that are superior and exploring the space of possible solutions in the vicinity of superior solutions.

The principle of an EA is that by making successive random variations of candidate solutions that are good so far, the algorithm is able to bias exploration of solution space (with respect to random search) to invest more exploration in those parts that are ‘most promising’. If, by virtue of the fact that ‘offspring’ may inherit some of their ‘parent’s’ desirable characteristics (i.e. fit individuals have a better chance of producing fit variants than unfit individuals do), then repeated selection and variation should exploit this bias appropriately and accumulate successive adaptive modifications. Accordingly, an EA is often considered to proceed something like a hill-climbing process (Muhlenbein 1992), except that an EA is usually based on a population of solutions rather than a single solution.

3.1.2 The Simple Genetic Algorithm

Evolutionary Algorithms, EAs, come in many flavours, for example: Genetic Algorithms, (Holland 1975), Evolutionary Strategies (Back et al. 1991), Evolutionary Programming (Fogel 1966), and Genetic Programming (Koza 1992). The differences between these methods largely depend on the default choice of representation (e.g. bit strings or real-valued vectors), the choice of variation operator (e.g. mutation and/or crossover), and the default selection method (e.g. fitness-proportionate selection, or truncation selection).

For most of the work that follows in this dissertation, the genetic algorithm, GA, is the best archetype (see Mitchell 1996). A GA usually uses a binary encoding over large linear chromosomes (fixed length bit strings), a large population (on the order of 100 to 1000), both point-mutation and crossover for variation, and fitness proportionate selection by default. The overall operation is often ‘generational’ meaning that the entire population of individuals is evaluated, selected and reproduced as a batch before the next cycle commences. Something like this is what is meant when referring to ‘the Simple GA’, SGA. A procedural outline of the Simple GA is given below (Figure 3-1).

1. Initialise population with random individuals, i.e. random bit strings.
2. Repeat until stopping criterion (e.g. threshold quality reached, or population no longer changing significantly).
    a. Evaluate the fitness of all individuals in population using an ‘objective fitness function’ giving the quality of each string.
    b. With selection probability of each individual proportionate to their fitness, select (with replacement) individuals from population to be ‘parents’.
    c. For each parent, p1, generate an ‘offspring’ individual by:
        i. Pick another parent to be p2, create a new string that has some bits from p1 and some bit/geness from p2. - ‘crossover’
        ii. Randomly change some of the bits in the string. i.e. apply point-mutation with low per-locus probability.
    d. Replace original population with new population of offspring.

Figure 3-1: A procedural outline for a generational ‘Simple GA’, SGA.

Step i) may be omitted for a ‘mutation-only’ version, in which case step ii) mutates a copy of the parent. There are a few implementation details that are not spelled-out in this description; we will give more precise descriptions for the algorithms that we implement.

This description includes mutation (Figure 3-2) and crossover as variation operators. However, some kinds of EA do not include crossover, and the basic intuition for many researchers (but no means all) about how EAs work considers crossover to be optional. Further, some researchers consider the operation of crossover to be useless or, at least, nothing more than a ‘macro-mutation’ equivalent to a large number of point mutations. Whether this is necessarily true or not is one of the research questions we want to answer. Accordingly, we will start by considering the operation of the GA without crossover, i.e. a mutation-only GA. Following sections will discuss the impact of crossover in detail.

101010001010010100 - parent

1010**0**000101**11**10100 - offspring with point mutations (bold)

Figure 3-2: Point mutations on a bit-string individual.

There are many different varieties of GA (as there are for other types of EA). For example, there are varieties that use different selection methods, different variation operators, different representations, and all sorts of sophisticated add-ons such as diversity maintenance techniques. There are also parametric choices to be made: the size of the population, whether or not to use crossover (or for what proportion of offspring to use crossover), how much mutation to use, etc. These differences make it very difficult to make thorough comparisons between one type of GA and another. In this thesis we are not going to try to make comprehensive comparisons. The particular variants of algorithms that we define, and the experiments that we perform, are provided for illustration. But we will see from analytic argument that different kinds of algorithm have profound differences in their behaviour and the kind of problems they can solve in principle.

3.1.3 EAs as Evolutionary Models

Evolutionary algorithms in computer science EAs are *inspired* by natural processes, but are not generally intended to be accurate models of biological evolution. EAs, as noted, are an optimisation and design technique, and most researchers feel free to introduce new features as required to address computational needs. Having said that, it is common to see researchers taking inspiration from evolutionary biology to enhance performance, e.g. island models to promote diversity (Starkweather et al. 1991), and also to see researchers appealing to biological plausibility as a guide for making choices in certain design decisions (usually in cases where computational preference is undecided).

Although evolutionary algorithms are generally not required to follow biological models closely, the basic framework of the EA outlined above adheres to some quite strong assumptions in common with early population genetic models - for example, the preoccupation with the genetic change of a single population (i.e. the coevolution of multiple species is often handled simplistically as purely adversarial interactions and generally does not involve genetic exchange between populations), and the assumption of a single fixed niche (i.e. all individuals are evaluated with a single given fitness function). The compositional phenomenon we address in this thesis require that we expand our models and relax our assumptions both in EC and in EB (as discussed in 2.5).

These assumptions restrict the kind of dynamics that may occur. In particular, starting with the assumptions that we are modelling the adaptation of a single species in a single niche. That is, all individuals in the population are candidate solutions for the same problem. This has several important consequences: First, all individuals are in a competitive relationship - superior solutions are promoted by competitively excluding inferior solutions from the population. In the absence of any additional measures to the contrary, this introduces an inherent pressure for the population to converge to a single type - around the current best solution. Second, it is therefore not possible that some subset of the population may diversify into a different niche and coexist without competing with the remainder of the population. Third, it follows that there is no possibility for different types to specialise diversely in parallel, or coexist, let alone enter into collective behaviours or cooperative relationships. Although there are a few exceptions and qualifications (e.g. fitness sharing and cooperative coevolution) that we will detail later, these limitations are true of the basic Simple GA and underlie the assumptions of EAs in general.

Some assumptions in the basic EA framework include:

- The model is based on the adaptation of a single population in a fixed niche. i.e. all individuals are assessed with respect to the same criteria - the same fitness function.
- Every individual has an identifiable fitness (objective performance metric) and all individuals can be ranked in this single dimension of quality.
- The only type of dynamic between variants is competitive exclusion. i.e. all individuals are in the same niche and compete to be members of the next generation.
- There is no change in the mechanisms of selection, variation, reproduction, or evaluation over evolutionary time.
- (At least in versions without crossover), the unit of selection, the individual, is considered to be unambiguous. i.e. parts do not reproduce without wholes, and wholes do not reproduce in groups.

3.2 Evolutionary difficulty for the Simple EA

The basic intuition of an EA is the same as the basic intuition for natural evolution: by accumulating small random variations that incrementally improve fitness we can arrive at progressively fit adaptations.

Accordingly, many of the same notions of evolutionary difficulty apply for artificial evolution as they do for natural evolution.

Fitness landscape

The fitness landscape is a surface defined over the genotype space where the height of each point is the fitness of that genotype and adjacent points are genotypes that can be reached by point mutation. Accordingly, the evolutionary process can be characterised as the ascent of the population mean up local gradients in the fitness landscape.

Interdependencies, ruggedness, and local optima

Difficulty in a combinatorial optimisation problem is largely created by the ‘frustration’ of problem variables. One variable is said to be dependent on another when the effect of changing the variable is dependent on the state of the other. Following the biological terminology, this is often called epistasis. Epistasis creates non-planar fitness landscapes, ruggedness, and local optima. A local optima is a point in the fitness landscape where all neighbouring points are of lower fitness, i.e. a point corresponding to a candidate solution that cannot be improved by any small change i.e. a mutation (we are still assuming that we are talking about algorithms without crossover, for now.)

Although GAs generally operate in a high-dimensional binary space, it is still popular to visualise fitness landscapes as two, or sometimes one dimensional, metric spaces. Intuitively, these are often imagined as a ‘walk’ through adjacent points in the high-dimensional space—i.e. adjacent points in the 1-D landscape are adjacent points in the high-dimensional space. The ruggedness of this landscape is then taken to indicate the correlation between the fitness of mutation neighbours in the search space. The figure shows some landscapes of increasing ruggedness, and increasing number of local optima, that are taken to indicate a spectrum of evolutionary difficulty.

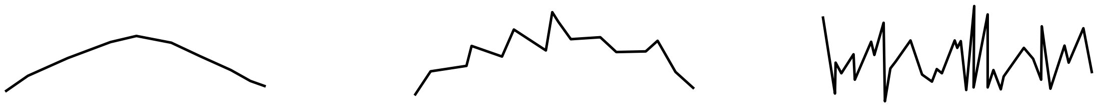


Figure 3-3: smoothness and local optima

a) a smooth unimodal landscape with no local optima—considered easy for EAs. b) a landscape with some ruggedness. c) an 'impossibly hard' highly-rugged landscape with many local optima. (see 'NK landscapes', Kauffman 1993).

Paths of monotonically increasing fitness (absence of ~)

In general, the evolvability of a solution for an evolutionary process is understood to be connected to the presence of a path of monotonically increasing fitness to that solution.[18] In general, in non-trivial problems, we do not expect there to be such paths from most points to the global optimum of a landscape. The absence of such a path corresponds to the notion of irreducible systems in evolutionary biology literature.

Convergence and diversity

In the Simple GA, one common explanation for failure is *premature convergence*, where the population loses diversity and becomes genetically similar before finding the global optimum. Thus it becomes stuck on a local optimum.

We will see that operators like recombination require us to revise our notions of evolutionary difficulty, and that these 'classical' notions of evolutionary difficulty are dependent on the underlying assumption of accretive change.

3.3 Compositional mechanisms in EAs

In the discussion above we have been considering the operation of the GA *without* crossover, i.e. a mutation-only SGA. *If* we view crossover as merely a macro-mutation then the difficulties discussed would apply equally to the operation of the simple GA *with* crossover. One of the main points we want to

[18] In fact, as the 'long path' problems attest (Horn et al. 1994), we also need this path to be short, as it is actually possible to construct problem spaces that have exponentially long paths of monotonically increasing fitness to the global optima.

elucidate in this work is that there are some circumstances where crossover cannot be accurately characterised as a merely a source of large mutations, and that the operation of an EA with crossover is fundamentally different in those circumstances from the operation of a mutation-only EA. For this reason, we have discussed the operation of the mutation-only GA, but for those that characterise the operation of crossover as macro-mutations, the discussion of evolutionary difficulty would end there. In short, the discussion thus far presents the accretive view of EAs (whether or not they use crossover). But, now we will discuss the operation of the EA from a compositional framework, in particular, the compositional interpretation of crossover.

Crossover potentially allows genetic material pre-adapted in different individuals to be combined together into a new offspring. Accordingly, crossover is a mechanism that potentially allows large non-random modifications and introduces the first of what we call 'compositional' operators for evolutionary algorithms.

3.3.1 The operation of crossover

Crossover, a simple abstraction of sexual recombination, creates an offspring individual by taking some genes (bits) from parent 1 and some genes from parent 2. Three types of crossover, commonly used in GAs are shown in Figure 3-4.

	one-point crossover	**two-point crossover**	**uniform crossover**
parent 1	**101010010010100**	**101010010010100**	**101010010010100**
parent 2	100101010011110	100101010011110	100101010011110
offspring	**1010100100**11110	**10101**1010011**100**	**1**0**1**10**0**0**1**00**101**10

Figure 3-4: Crossover operations on bit-string individuals.

Crossover points are indicated by the short vertical tick marks. Possible crossover points for uniform crossover are indicated by the dotted marks. In GAs we normally consider only a haploid chromosome, i.e. crossover is applied as soon as the two haploid chromosomes coming from the parents are brought together and before evaluation/selection is applied.

In one-point crossover, a crossover point is chosen, and all alleles on the left of this point are taken from parent 1, all alleles on the right are taken from parent 2, or vice versa. In two-point, or more generally n-point crossover (as is common in drosophilla), contiguous blocks of alleles between n randomly positioned crossover points are taken from each parent, swapping donors at each crossover point, as shown.[19]

Under uniform crossover (Syswerda 1989), alleles (bits) are taken from either parent with equal probability on an independent per locus basis. The most obvious difference between uniform crossover and n-point crossover is that under uniform crossover there is no correlation between adjacency on the chromosome and the probability of genes being transferred to the offspring. Put another way, the combination of p1 and p2 genes that end-up in the offspring would be the same (statistically) regardless of how those are ordered or positioned on the chromosome. In the terminology of population genetics, it would be said that there is no *genetic linkage* between genes under uniform crossover, but there is for one and two point crossover. In biological terms, uniform crossover corresponds to a case where all the genes of interest reside on different chromosomes, or cases where there are a very high number of crossovers per reproduction, or cases where several (one-point) recombinative events occur before selection has an appreciable effect on its results.

Lastly, note that for binary strings, uniform crossover has the same effect as producing an offspring with alleles the same as the parents' alleles where the alleles of the parents agree, and randomly assigned alleles at all other loci (Watson & Pollack 2000b). From this it is easy to see why uniform crossover, in particular, is considered to be nothing more than a macro-mutation (albeit one that preserves similar parts of the parents - Chen 1999).

Whether because they assume the use of uniform crossover, or because (quite reasonably) they do not want to make any assumptions about the positions of genes on the chromosome, for the many researchers that consider crossover to behave like a macro-mutation, the same ideas about evolutionary difficulty hold for GAs *with* crossover as for crossover without. In this point of view, it is merely the case that crossover allows large non-random changes—more specifically, changes that are no more likely to be adaptive than an equivalent number of point mutations. However, there has been a long-standing battle in GA theory

[19] We can see that one-point crossover is the same as two-point crossover where one of the two points is always positioned before the first (or after the last) locus. Accordingly, one-point crossover treats the end-points of the string as special points whereas two-point crossover does not.

about the adaptive utility of crossover (if any). Primarily, this revolves around the notions of the *building block hypothesis*.

3.3.2 The Building block Hypothesis, BBH

Following the notions outlined by Holland (1975), Goldberg (1989) says the following about the operation of the GA under the heading “The building block hypothesis”:

> *Short, low-order, and highly fit schemata are sampled, recombined, and resampled to form strings of potentially higher fitness. In a way, by working with these particular schemata (the building blocks), we have reduced the complexity of our problem; instead of building high-performance strings by trying every conceivable combination, we construct better and better strings from the best partial solutions of past samplings.*

Forrest and Mitchell (1993a) restate these ideas as follows:

> *The Building block Hypothesis states that the GA works well when short, low-order, highly-fit schemas recombine to form even more highly fit higher-order schemas.*

More recently, Holland (2000) says the following:

> *The ‘BB thesis’ holds that most of what we know about the world pivots on descriptions and mechanisms constructed from elementary building blocks…Once a computer scientist starts thinking about building blocks as a source of innovation, the next obvious step is to look for algorithms that can discover and exploit building blocks. It is my claim that genetic algorithms (GA’s) are particularly qualified for this task…. What purpose does crossover serve then? As is well known to most practitioners it serves two purposes: First of all, it recombines building blocks residing on separate strings into combination on a single string. The second effect is more subtle. Crossover uncovers new building blocks by placing alleles in new contexts. Moreover, …building blocks are often combinations of more elementary building blocks, so recombination works at this level too.*

Put simply, the BBH suggests that the GA will perform well when short sections of chromosome that confer above average fitness contributions (‘building blocks’) can be recombined to discover longer sections of chromosome that are still fitter. The intuition is that different members of the population will discover different building blocks, and that sexual recombination will then search combinations of small building blocks to find larger building blocks, and so on.

Before we delve into the ideas in this hypothesis, we stress that it concerns the operation of the Simple GA (with crossover) and the ability of the hypothesis to explain the operation of the simple GA is quite a different issue from whether or not there is some useful intuition about how a ‘GA-like’ algorithm *could* work. More specifically, we are going to argue that the idea described here, if it can be made to work (for some type of EA on some class of problems), would be a valuable process—even if it does not explain the operation of the Simple GA on problems in general.

Goldberg (above) describes building blocks as “short, low-order, and highly fit schemata”. As stated, a schema is a partial specification of the problem variables, or the set of all candidate solutions containing that subset of values. A schema is often depicted as a tertiary string as in the figure, where “*” can be read as “any allele can go here”, or “this gene is not specified by the schema”. The *order* of a schema is the number of specified genes, and so a *low-order* schema is one with a small number of specified genes. Finally, the *length* of a schema, or *defining length*, is the number of loci between the first and last specified loci (inclusive) (Figure 3-5).

```
a) ***1**10*******0*0*

b) ***11*10*0*********

c) 1**0010100*01010000
```

Figure 3-5: Schemata of various orders and defining lengths

a) a low-order schema (order 5, length 15). b) a short, low-order schema (order 5, length 7). c) a high-order schema (order 16, length 19).

Underlying the claim in the BBH is the idea that the GA promotes copies of short, low-order schemata of high fitness in the population. This claim is based on what is known as the *Schema Theorem* (Goldberg p.33) which says “short, low-order, above-average[-fitness] schemata receive an exponentially increasing number of trials in subsequent generations”. (See also Altenberg 1995). This is intuitive enough; individuals containing above-average fitness schemata will tend to be above-average fitness individuals;

these individuals will produce more offspring than other individuals under fitness proportionate selection; and when the schemata have short defining length, there is a good chance that they will appear intact in their offspring. However, whether the Schema Theorem has anything to do with explaining the operation of the GA is questionable (Vose 1999).[20]

Nevertheless, it is worth examining what crossover can achieve (if anything) by searching combinations of schemata, regardless of how they come to be in the parents, and regardless of whether or not they are *building blocks* as Goldberg defines them, i.e. we suggest that the Building block Hypothesis warrants some discussion regardless of the validity of the Schema theorem. The intuition from the BBH that we want to examine is that low-order schemata can be combined together to find higher-order schemata of higher fitness—and we separate this idea from the specific hypothesis that this is the way the Simple GA operates.

Goldberg indicates the notion of *problem decomposition* that is implied. Specifically, that the search space is reduced to a process of searching combinations of schemata rather than searching combinations of the original problem variables. In this manner, the GA should be able to move from searching combinations of the original problem variables to searching combinations of modules, and this process should continue through combinations of modules of successively larger scale. If it were possible for an EA to discover and manipulate useful modules effectively, it would be fundamentally valuable: 1) the EA would reduce the dimensionality of the search space, effectively making a new problem over a smaller number of variables; 2) introducing a useful module into a candidate solution introduces a large pre-adapted set of features simultaneously. (The latter is really a different interpretation of the former.) Algorithmically, this, if it can

[20] One reason for this is because it is difficult to believe that selection in the GA can accurately respond to 'schema fitness'. Schema fitness is the average fitness of points in the schema hypervolume, and is supposed to indicate the fitness contribution coming from that schema. However, in practise, the GA can only assess the fitness of individuals in the population which provides a very impoverished sample of the schema hypervolume. For example, consider a very low-order schema, say with one defined allele. This schema contains $2^{(N-1)}$ strings, where N is the number of bits in an individual. The members of the population can hardly be expected to give a reasonable sampling of the schema fitness therefore. (However, a possible exception to this is implied by the assumption that the dependencies between loci in this schema and loci outside the schema is weak, low, or limited in some sense, as we shall discuss. In this case, accurate schema fitness could be assessed from a small number of individuals.) The question then, is whether a sub-string representing a schema of interest is exchanged from one genetic context to another fast enough to test enough contexts such that it is really schema fitness that is being rewarded. If not, it is likely to be the *individuals* (containing fit schemata and some unfit schemata, or 'hitch-hikers', residing in the same individual) that get propagated, not the schemata *per se*.

be made to work, is an intuition worth pursuing. In order to better understand the fundamental issues involved here, let us first take a step back to consider the placement of EAs in the larger picture of optimisation algorithms in general.

3.3.3 Algorithmic Principles - Divide-and-Conquer Problem Decomposition

Algorithms for combinatorial optimization can be divided into different classes depending on the different assumptions they entail about the nature and structure of interdependencies in a problem domain. At one extreme we might assume that the interdependencies of the variables are arbitrarily difficult or random. In this case, the appropriate class of optimization methods are those that make no assumptions about the structure of the fitness landscape. These include exhaustive search, which explores the space of possibilities in its entirety, and random search, which explores the space of possibilities with uniform probability. Of course, these are impractical in all but the smallest of problems since the number of possibilities is subject to ‘combinatorial explosion’ as the number of problem variables increases.

At the other extreme, we might suppose that the interdependencies between variables are weak or few. In this case, the appropriate class of algorithms uses this assumption to sample points non-uniformly, directed by prior samplings using the heuristic that effects of variables are approximately additive. These include ‘greedy’ ‘hill-climbing’ algorithms, which accept small variations to a candidate solution if they are superior, and gradient ascent methods. The effectiveness of this class depends on the implicit assumption of low interdependency between the variables in the problem because its effectiveness requires that locally beneficial variations may accumulate without mutual interference.

These two classes represent the naïve extremes of algorithmic possibilities: whereas exhaustive algorithms can deal with arbitrarily difficult interdependency of the problem variables, hill-climbing algorithms are only guaranteed optimal when the interdependency of the variables is weak or unimportant.

In between these two extremes there are many other possibilities. One possibility is the class of ‘divide and conquer’ algorithms (see, for example, Cormen et al. 1991) which broadly include algorithms that utilize modular structure in the interdependency of problem variables. Divide and conquer, D&C, techniques exploit decomposable problems by dividing them into manageable sub-problems. Their appropriateness depends on the assumption that the problem variables are neither close to independent (as for hill-climbers) nor arbitrarily interdependent (as for exhaustive methods) but that the problem variables have some intermediate modular structure of semi-independent subsets.

There are a number of different D&C methods. In design and engineering domains, top-down functional decomposition of a problem into semi-independent sub-problems is familiar and intuitive. In combinatorial optimisation, dynamic programming methods are an important class of D&C technique that utilize structured dependency bottom-up by caching all partial solutions, without *a priori* knowledge of how to decompose the problem. Either top down or bottom up, divide and conquer strategies are fundamentally important because they exploit modular structure in a problem by re-using solutions to semi-independent sub-problems and they thereby avoid the ‘combinatorial explosion’ associated with high-dimensional problem spaces. See Figure 3-6.

Consider optimising a function, ***f***, over a set of N variables each of arity *k*. i.e. optimise ***f***$(n_1,n_2,\ldots,n_N)$.

Random Search

If we make no assumptions about the interdependencies of these variables then the size of the search space that must be covered to find the optimal solution is (at most) $S_R=k^N$.

Hill-climbing

If the variables are individually separable, that is, if the contribution of one variable is unaffected by the values of other variables, then ***f*** can be described using the summation of N functions over one variable each. i.e. the function $\boldsymbol{f}(n_1,n_2,\ldots,n_N)= \boldsymbol{f}_1(n_1)+ \boldsymbol{f}_2(n_2) +\ldots+ \boldsymbol{f}_N(n_N)$. In this case, the size of the search space that must be covered to optimise ***f*** is at most $S_{HC}=k$N.

These are the two naïve extremes: no assumptions about interdependency gives a search space of k^N, the assumption that all variables are independent of one another gives a search space of *k*N. The former is intractable for all but very small N, and the latter gives a problem that is trivially easy.

Divide and Conquer problem decomposition

In between these two extremes, there are a lot of other possibilities. Consider a case where the variables have some interdependency but their interdependencies are formed over two disjoint subsets of the variables. For example, $\boldsymbol{f}(n_1,n_2,\ldots,n_N)= \boldsymbol{f}_1(n_1,n_2,\ldots,n_{N/2}) + \boldsymbol{f}_2(n_{N/2+1}, n_{N/2+2},\ldots,n_N)$.

Here, where the problem can be divided into two separable sub-problems (e.g. see Royal Roads and Concatenated trap functions Section 3.5.1), we can maximise the function ***f*** by maximising the function $\boldsymbol{f}_1$ and the function $\boldsymbol{f}_2$ separately (later, we will define a class of problems that is decomposable but not separable (4.3.1)). So, if this decomposition is known, then the total search space that must be covered is at most $S_{DC}=2\cdot k^{N/2}$. In general of course, the decomposition is not known and discovering the decomposition adds some overhead to the method. $2k^{N/2}$ is much better than k^N, of course, and forms the basis of the divide and conquer algorithmic advantage—when the overhead of finding the decomposition is low. Conceivably, a problem could be divisible into many small sub-problems, and/or each sub-problem may be further divisible into sub-sub-problems, and so on.

Figure 3-6: Outline of algorithmic advantage from problem decomposition

In human design and engineering problems, we use top-down knowledge of the problem structure to divide a problem into sub-problems. For example, we might divide the problem of designing a vehicle into the problems of drive power, steering and control, load bearing, etc. Or we may divide the problem of timetabling a rail network into the many sub-timetables within each region. However, notice that although it is familiar and intuitive to approach design problems this way, and we gain significant leverage by doing so, we are cognisant of the fact that the sub-problems are almost never entirely independent. For example, the details of the drive power system, if we choose a heavy engine for example, places constraints on the design of both the control system and the load bearing system. Similarly, trains do travel between regions, as well as within regions, so the timetable of one region is not independent of that in other regions. Thus the division of a problem into non-overlapping sub-functions depicted above is unrealistic. More likely, the sub-functions are not completely independent. We will discuss this further in later sections, and define the notion of a *decomposable* function in contrast to this kind of *separable* function. However, despite sub-problem interdependencies, design problems are often fruitfully approached by working on the sub-problems, finding some reasonable solutions for the sub-problems, and then resolving the interdependencies between alternate sub-solutions.

So, this divide-and-conquer problem decomposition is very advantageous algorithmically, when applicable. And it is an intuitive and fruitful approach for human design and optimisation problems. Moreover, as mentioned, there are bottom-up methods of gaining this algorithmic advantage that do not require the decomposition of the problem to be known *a priori*. (The need for top-down knowledge must be avoided if we are interested in biological analogues.)

Clearly, if some form of problem decomposition were available to an EA, it would be quite different to the simple ‘hill-climbing’ notion of gradual improvement through random variation and selection. The intuition underlying the Building Block Hypothesis is just this - that the GA with crossover, by manipulating building blocks, exploits a form of problem decomposition. Our research motives are not specifically wedded to the Simple GA, but this question provides one of the primary EC motives for this thesis research: to clarify whether some form of EA can in some circumstances provide an algorithmic advantage akin to divide and conquer problem decomposition.

It is not clear that it is available in a Simple GA, but our later experiments will show that it is available in a particular variety of GA when applied to a suitable class of modular problem. In the next chapter we will discuss carefully the kind of modularity that is in principle amenable to decomposition, or more to the point, composition, by mechanisms like crossover. And later in this chapter we will discuss some of the conditions that must be met for crossover to successfully manipulate and recombine modules.

3.3.4 Mechanisms of encapsulation in EAs

In addition to the operation of sexual recombination, there are also other variation mechanisms employed in artificial evolution methods that enable forms of composition.

For example, modularity is addressed implicitly by the use of variable-length, moving-locus, non-linear, and generative encodings - for example, Messy GA (Goldberg et al. 1989), Linkage learning GA (Harik & Goldberg 1996), Genetic Programming (Koza 1992), and cellular encoding (Gruau 1994). And modularity is addressed explicitly in mechanisms that 'encapsulate' subsets of features for subsequent re-use during the search process - for example, 'automatic module acquisition' (Angeline & Pollack 1993), 'automatically defined functions' (Koza 1994), and 'adaptive representation' (Rosca 1997). The advantage of these explicit methods is that "the modularization of representational components and their protection from mutation [/internal variation] can be viewed as removing unnecessary dimension[s] from the search space…" (Angeline & Pollack 1993).

Methods explicitly addressing problem decomposition (the division of labour) include Learning Classifier Systems (Holland & Reitman 1978), Cooperative Coevolution (Potter 1997), Evolutionary Divide and Conquer (Rosca 1997), MIL (Juille 1999), 'arbiters' (Husbands 1993), 'Multi-level Cooperative Search' (Valenzuela & Jones 1994), and SANE (Moriarty 1997), as well as techniques embedded in the modularity methods listed above.

We address some aspects of some of these models in the following subsections. However, we will find that Cooperative Coevolution and the Messy GA encompass most of the important concepts in a straightforward manner and we will refer to these repeatedly as we develop the models that follow in later chapters.

3.3.5 Other related divide and conquer methods

Adaptive representation and hierarchical learning

Rosca, (1997), develops techniques covering most of the fundamental ideas we are developing. Specifically, the work explicitly addresses divide and conquer problem decomposition where individuals specify for a sub-domain of a function, and Rosca explicitly pursues hierarchical assembly of specialists into complete solutions. Individual specialists are evaluated collectively. Diversity is maintained by techniques similar to restricted mating – where mating is restricted to individuals in the same sub-domain cluster. The role of an individual in the collective task is specified by the 'sub-domain' part of an individual, and the behaviour of the individual is specified by a function within that sub-domain (analogous to the "condition" and "output" parts of individual classifier rules). The sub-division of the problem domain is not given *a priori*, (see also Juille & Pollack 1996). The substrate is genetic programming s-expressions and the problem domain is symbolic regression. Crossover of s-expressions can enable composition. The main limitation in this work is that although a) there are diverse specialists covering the problem space, b) there is a combination operator (GP crossover) that could put specialists together into a generalist, - in fact, the crossover operator cannot put diverse specialists together into a composite because the only way they were maintained as diverse specialists in the first place is by restricted mating. So, in fact, there is no clear hierarchical assembly of coadapted specialists into composites. Additionally, since this work is in the GP domain we cannot connect it to the BBH and string-based crossover. Also when applied to arbitrary symbolic regression problems that may or may not be amenable to decomposition, it is difficult to assess exactly how it works when it does and analyse its behaviour. However, this work suggests a lot of potential to bring the ideas we develop in this thesis into the GP domain.

Cooperative Coevolution

Potter (1997, Potter & De Jong 2000) presents a general divide and conquer approach to evolutionary algorithms he calls "Cooperative Coevolution". In general, individual specialists are coevolved to collectively solve a problem and the notions of divide and conquer problem decomposition are very explicit in this work. In earlier work, (1997), there are explicit fixed roles for the individuals, later methods (Potter & De Jong 2000) show some ability for individual species to adaptively cover the problem domain and vary in the number of specialist species required. Potter also explicitly uses the idea of restricting competition to act only between specialists of the same type – specialists of different types cannot competitively exclude one another because they are held in different populations.

However, the ‘specialisation’ demonstrated in the ‘string covering problem’ (Potter & De Jong 2000) is of a particular type. Specifically, a collection of individuals are each evaluated against a test string and the best one is used. This means that if some other member of the collection contradicts the specifications required for the test case then it does not affect the fitness of the group. In a straightforward function optimisation scenario this is not the case – the collection of specialists must internally resolve what single value will be used for each parameter. In short, this example does not show automatic discovery of the roles of specialists in a way that we can use. The way that specialists are used in some other examples of the method is quite different and more appropriate for our purposes – specifically, each specialist covers an explicit subset of the problem parameters. However, there is no automatic method to determine which subset should be covered in Cooperative Coevolution.

Also, there is no notion in Cooperative Coevolution of assembling specialists together in a hierarchical fashion. Representatives of specialists species are evaluated together in a group but a successful group is never encapsulated so that we can form meta groups. This means that if there are two alternate ways to be a particular kind of specialist and one kind works well in some contexts and the other kind works well in other contexts then there is no way to maintain both competing types in a way that respects the preferred contexts of each. Rather there is only one group that can be reliably maintained – the group formed by the representatives of each sub-species where each sub-species is converged. In short, alternate assemblies of specialists cannot be maintained and selected for, and the problem decomposition is necessarily only a single level of decomposition.

In summary, Cooperative Coevolution does not have a composition operator or a way of assembling specialists into composites recursively; also Potter does not provide a formal description of the class of problems for which Cooperative Coevolution is well suited. However, this work explicitly uses the notion of divide and conquer problem decomposition in many different substrates including string-based function optimisation, the notion that competition should occur between specialists within a type, and specialists across types should be allowed to coexist and collectively cover the problem domain.

Classifier systems

Classifier systems provide an interesting spectrum of divide and conquer optimisation at different scales. ‘Michigan-style’ classifiers (Holland & Reitman 1978) only optimise at the small-scale (selection is for individual rules). This is efficient when rule interdependency is insignificant or of the easy kind (see 4.2), but will produce sub-optimal results otherwise. In contrast, the ‘Pitt’ approach to classifiers (De Jong 1988) can in principle find optimal configurations of rules by selecting for whole rule sets - but in general suffers from the combinatorics introduced by the much larger search space this defines. Hierarchical classifiers (see Barry 1996) potentially offer the advantage of both approaches. Some of the mechanisms we develop in this thesis, for selecting for complementary sets of schemata and for determining when a collection of schemata makes a valuable composite, may be useful in classifier domains.

Other problem decomposition approaches

Immune systems (Dasgupta & Forrest 1999), and Ant Algorithms (Dorigo et al. 1999) also gain their basic intuition from the algorithmic advantage of problem decomposition by selecting for individual antigens and individual ants that collectively provide solutions to a protection problem or graph routing problem, respectively.

3.3.6 Explicit symbiosis/symbiogenesis models

There are a number of models in the EC literature pertaining specifically to symbiosis and/or symbiogenesis. We review these below with particular emphasis on: a) the use of a composition operator that joins entities together into a new whole, b) multiple hierarchical levels of composition, c) the modularity of the problem domain, d) whether the roles of symbiotic specialists are predefined or are defined adaptively.

Barricelli, 1957, “Symbiogenetic Evolution Processes Realised by Artificial Methods”

Barricelli (1957) sets out to model symbiogenesis, and to use computer simulations to investigate the dynamics of symbiotic processes. This work predates some of the work often cited as the foundation of work in genetic algorithms (e.g. Holland 1975) and it is remarkable how many of the ideas we are discussing in this thesis are present in this work. For example, the idea of compositional mechanisms and their combinatorial advantage, the potential for open-ended evolution under compositional mechanisms and its superiority to monolithic approaches, and the close relationship of sexual recombination and symbiogenesis, are all present in this work. Additionally, Barricelli uses a substrate where the boundaries of organisms are flexible and allow for the possibility of encapsulation. He also discusses horizontal gene transfer, and the tendency of populations to converge without diversity maintenance mechanisms.

However, this work is performed in 1D and 2D cellular automata with fairly *ad hoc* transition rules. This is an interesting substrate and the implicit boundaries of replicating patterns in the automata is potentially a powerful means to develop complex composition results, but it is entirely impenetrable to analytic treatments. It is also very difficult to see how results in this domain impact our understanding of, what have now become, the traditional GA and related hypotheses like the BBH. The main problem in this respect is that there is no explicit goal or function to be optimised in this system, which is fine as an artificial life experiment, but does not assist us in identifying the adaptive capacity of different mechanisms in a measurable way.

Bull & Fogarty, 1995, "Artificial Endosymbiosis"

Bull and Fogarty (1995, Bull et al. 1995), present a computational model of endosymbiotic mechanisms. This and other models that these authors present, seem to be intended as an examination of the conditions under which endosymbiotic relationships will occur rather than an examination of when such mechanisms can provide superior optimisation. The authors provide two possibilities for evolving entities – a) a specification for the evolving genes is provided by two independently evolved half-specifications, b) a specification for the evolving genes is evolved as a whole. Pairs of individuals from the first case may be composed together to form individuals of the second case. The problem domain is a two-component NKC landscape and it is shown that when the dependency between the two clusters of variables in the problem is low case-a predominates, and when it is high case-b predominates. The roles for each of the entities involved is predetermined (each individual is assigned one of the clusters of variables in the problem), and there is no possibility that one kind of specialist may over-write a specialist of the other type. In summary, this work illustrates the idea of problem decomposition and composition of sub-solutions, and it also carefully examines the conditions of the problem domain that induce different results. However, the problem domain has only two modules and no hierarchical structure, so there is no possibility of repeated composition of entities. There is also no mechanism for automatically identifying the sub-domains in the problem. Later work, (1996), examines the alternatives of individuals that are either, a) a sexual host plus an endosymbiont, or b) as before. This work explicitly refers to the parallels of sexual recombination and symbiosis. However, it does not use a composition operator, and has the same limitations with respect to the number and identity of specialist's roles. Bull (1997) and (1999) apply similar ideas to other evolutionary transitions. Bull (1999) interestingly, mentions the possible involvement of the Baldwin effect which relates to some ideas that we develop in Chapter 7.

Tomlinson & Bull, 2001, "Symbiogenesis in Learning Classifier Systems"

Following ideas introduced by Wilson and Goldberg (1989), Tomlinson & Bull examine the composition of rule into sets of rules that are chained together. Rules that are composed together reproduce together as a group. They also utilise the notion of 'encapsulating' a set of rules with respect to their 'behaviour' – i.e. the opportunity for rule firing when multiple rules can be applied at a given time is resolved with reference to set membership. This work provides the opportunity for hierarchical compositions (implicitly), and the role of a rule is flexible not predetermined. This embodies many of the concepts we are examining in this thesis but in the domain of classifier systems rather than straightforward function optimisation.

Demeur, R, 1995, "Evolution Through Cooperation: The Symbiotic Algorithm"

Demeur presents a very interesting and unconventional style of GA. The method uses a tree of parameter values, holding the values of problem parameters. The ordering of leaves and sub-leaves in the tree controls the co-occurrence of problem parameters in a candidate solution. 'Symbiotic associations' are represented by co-occurrence of parameters on a sub-tree. The only application tested is to cover the peaks of a multi-modal function (a sinusoid) – it is not clear what the modular structure of this problem is in terms of the problem variables (i.e. in the binary variables that represent a candidate solution), if any. It may be the case that the mechanics of this approach are not all that different in spirit from those we develop in Chapter 7, but the approach is too radically different to see the parallels with this and other GA work clearly. In the representation that Demeur uses it is very difficult to make associations with the BBH, to identify the class of problems that it is well-suited for, or to ascertain its ability to maintain diversity of competing schemata, for example.

Numaoka, 1995, “Symbiosis and Coevolution in Animats”

Numaoka provides an ‘animat’ based model in which the sensing ability of an agent is evolved. Evolved behaviour, controlled by two 8-bit parameters of the sensors, is applied to a foraging task called the ‘blind hunger problem’. There is no explicit partial specification – i.e. specialisation of different individuals to different sub-parts of the problem domain – or concatenation of partial solutions into a whole. The genetic operator is more like horizontal gene transfer (see ‘Microbial GA’ Harvey 1996, also in Watson et al. 2000a) than symbiotic encapsulation.

Kvasnicka, 2000, “An Evolutionary Model of Symbiosis”

This work, (inspired by Watson & Pollack 1999d) includes explicit divide and conquer problem decomposition (in the style of Bull and Fogarty), and a concatenation operator that assembles partial solutions together. However the problem domain is a concatenation of several non-overlapping NK landscapes which makes sub-problems that are entirely separable, thus this problem would be amenable to a Macro Mutation Hill Climber (see 4.3.1, 4.6.4). Kvasnicka uses an ad hoc method to limit string bloat (see 9.3.2) – i.e. only if a certain fitness threshold is reached is the composition operator applied – in general, there is no means to know where this threshold should be placed on an unknown problem. The roles of independent symbionts are predefined, and tight genetic linkage (see 6.4) is assumed.

Hirasawa, et al., 2001, “Genetic Symbiosis Algorithm”

The algorithm presented here is the same as the Simple GA but with the addition of some ‘symbiosis parameters’ that modify fitness. These parameters for a pair of entities are derived from Euclidean distance between genotypes and the fitness difference of individuals, and define a matrix of association relations that control competition, exploitation, and mutual benefit. There is no composition operator, and no hierarchy in variation or in the problem domain. The method essentially provides an interesting form of fitness sharing for diversity maintenance. Mao et al., (2001) use similar ideas to handle interaction among objectives.

Tsujimura et al., 2001, ‘Symbiotic Genetic Algorithm’

This algorithm is specific to the domain of job shop scheduling. It applies selection based on two objectives (make span, and idle time) in a manner similar to implicit fitness sharing. A third objective, (total job waiting time), is applied to promote diversity. The algorithm has no composition operator, and no hierarchy.

3.3.7 Summary of existing methods

Table 3-2 summarises the features of this existing work that we are interested in.

	Compos-ition	**Hierarchy**	**Modules**	**Roles**
Barricelli, 1957, "Symbiogenetic Evolution Processes Realised by Artificial Methods"	Y (implicit)	Y/N (implicit)	Y (implicit)	Y (implicit)
Bull & Fogarty, 1995, "Artificial Endosymbiosis"	Y	N	N	N
Bull & Fogarty, 1996, "Horizontal Gene Transfer in Endosymbiosis"	N	N	N	N
Bull, 1999, "On the Evolution of Eukaryotes: Computational Models of Symbiogenesis and Multicellularity"	Y	N	N	N
Tomlinson, & Bull, 2001, "Symbiogenesis in Learning Classifier Systems"	Y	Y (implicit)	Y (implicit)	Y
Demeur, 1995, " The Symbiotic Algorithm"	Y	Y (implicit)	Y (implicit)	Y (implicit)
Numaoka, 1995, "Symbiosis and Coevolution in Animats"	Y	N	N	NA
Kvasnicka, 2000, "An Evolutionary Model of Symbiosis"	Y	N	Y	N
Hirasawa, et al. 2001, "Genetic Symbiosis Algorithm"	N	N	N	Y
Tsujimura et al. 2001, "Symbiotic Genetic Algorithm"	N	N	N	N
Goldberg et al., 1989, "Messy GA"	Y	Y/N	Y	Y
Potter, 1997, "Cooperative Coevolution"	N	N	Y	Y/N
Rosca, 1997, "Evolutionary Divide and Conquer"	Y (implicit)	N	N	Y
Angeline & Pollack, 1993, "automatic module acquisition"	Y	Y (implicit)	N	Y
Juille, 1999, "Modular Inductive Learning"	N	N	Y	Y

Table 3-2: Summary of features in existing methods

"Composition" - employs a composition operator that encapsulates specialist entities together into a new composite entity. "Hierarchy" - employs composition on multiple hierarchical levels (as opposed to one- or two- level decomposition). "Modularity" - number of modules in the problem domain is more than 2. "Roles" - roles of coadapted entities are defined adaptively (not *a priori* predefined roles)

In conclusion, only Rosca, Demur, Barricelli, Tomlinson, and Goldberg apply a composition operator over multiple levels. None have both a principled means to promote complementary specialists and a means to join specialist together into composites - except perhaps, Demur and Tomlinson. Tomlinson's work is in classifier systems domain. Demur does not show operation on a hierarchical problem (but perhaps it could provide this in principle). Thus none of the existing methods demonstrate all of the properties we are interested in but many of them provide important parts of the feature set we will develop.

3.4 Evolutionary difficulty under compositional mechanisms

In this subsection we will outline some of the numerous issues that arise in the implementation of EAs using compositional mechanisms.

3.4.1 Difficulties of building block processing under recombination

Two practical difficulties with BB crossover in GAs come from a lack of diversity in the population and 'disruption'. That is, when fitness proportionate selection is applied to a panmictic population it tends to become genetically converged very quickly and thus sexual recombination has no effect. And the problem of disruption of a BB in crossover refers to the fact that the stability of a building block, i.e. its likelihood of being recombined with other building blocks appropriately, is dependent on the co-location of constituent genes on the chromosome – i.e. 'tight' genetic linkage. Considerable research effort to improve the performance of EAs has been directed at issues of diversity maintenance and robust representations of functional sub-components that would allow appropriate manipulation of building blocks when discovered.

3.4.2 Schema disruption: genetic linkage, partial specification, and competing conventions.

Schema disruption

A long defining length (see Figure 3-5) increases the probability that a useful schema in a parent will be disrupted in reproduction with crossover. If disruption of schemata by crossover is high then even if a schema has a fitness that is above-average it may not receive an exponentially increasing number of trials in subsequent generations. Initial disruption analysis by Holland (1975) has been developed further by many others (e.g. De Jong 1975, De Jong & Spears 1990). The selection schemes that we use in our models are quite different from those where such an analysis is important, and instead we are more concerned with the likelihood of schema creation. However, issues of favourable genetic linkage, which have a strong effect on schema disruption, are also relevant to schema creation, as we will discuss (see below and 6.4).

Poor genetic linkage

Figure 3-7 shows the possibility of combining two schemata under one-point crossover assuming tight linkage. However, we also illustrate the difficulty of combining two schemata in cases of poor linkage.

tight linkage	poor linkage
10**1010**010010100	10**1**01001**0**010**10**0
10010101001**1110**	**1**00**1**0**1**010**0**11110
10**1010**01001**1110**	?

Figure 3-7: Building block combination and disruption

Left) Combining two low-order schemata using one-point crossover is possible when genetic linkage is tight, i.e. defining length is short, (see Figure 3-5). Right) when linkage is poor because the ordering of epistatically related genes is random, one-point crossover cannot create an offspring with both schemata.

In general, in cases with random genetic linkage an arbitrary number of crossover points must be placed at appropriate positions in order to guarantee that good schemata from each parent can be brought together in the offspring. Accordingly, we see that one of the weaknesses of crossover is that the modules that we wish to manipulate are only represented implicitly by virtue of the collocation of their genes on the chromosome.

It should be clear at this stage that the notion of what bits ‘should’ travel together in crossover events because they are epistatically dependent on one another and correspond to useful schemata, is quite different from the notion of which bits are actually likely to travel together under regular crossover because they are close to each other on the chromosome. We call the former ‘dependency’ or ‘epistatic dependency’, and we call the latter ‘linkage’ or ‘genetic linkage’. So one way to rectify the problem of poor linkage is to re-order the genes so that interdependent genes are close together - so that genetic linkage corresponds with epistatic dependency. Accordingly, several variants of the GA have proposed ‘moving-locus’ schemes, e.g. GA with inversion operator, Messy GAs (Goldberg et al. 1989), Linkage Learning GA (Harik 1997). Despite the validity of these approaches, it is important to remember that the need to rearrange genes is a result of using a recombination operator that is sensitive to gene positions. And it should also be noted that genetic linkage based on gene ordering on linear chromosomes cannot in principle represent epistatic dependencies that cannot be linearised.

Unfortunately, the notions of epistatic dependency and genetic linkage are sometimes conflated. For example, the term ‘linkage learning’ is a case in point. Linkage learning in some cases (e.g. Harik 1997) refers to the process of discovering and representing the epistatic dependency of variables in a problem by moving them around so that they are next to each other on the chromosome. In other cases, e.g. Kargupta 1997, linkage learning was used to mean simply the discovery and representation of epistatic dependency by whatever means, and not necessarily by the use of rearranging genetic linkage. More recently, explicit probabilistic models of epistatic dependencies between genes have been based on Bayesian nets (see 6.4.4).

A different method proposed to overcome the weaknesses of crossover is the use of ‘crossover masks’ - additional binary vectors, specified along with each chromosome - that specify which bits of the chromosome should travel together in a crossover event (Louis & Rawlins 1991). Other methods use a real-valued vector that represents the probability that each bit will be transferred to the offspring. In these methods there are various methods required to regulate how the values in the mask vectors are updated (e.g. see Vekaria & Clack 1998).

Similar to the notion of crossover masks, is the use of ‘partially specified representations’. That is, methods where each individual may represent a variable number of genes and may therefore leave some alleles for some genes unspecified. The Messy GA combines the use of partial specifications with moving-locus representations (Goldberg 1989). Previous work has discussed the relationship between the use of partial specifications and moving-locus representations, and the other properties of the Messy GA (Watson & Pollack 1999c).

One useful interpretation of the use of partial specification is that whereas under regular crossover it is problematic to determine which subset of genes in each parent should be inherited in the offspring, when combining together partially specified individuals the individuals themselves are already explicitly representing the desirable subset of genes.

0**1**010**0**1**1** - GA

((2,1), (6,0), (8,1)) - Messy GA, moving-locus

-1---0-1 - fixed-locus, partial-specification

Figure 3-8: Representation of genomes/schemata for GA, Messy GA, and 'partial specification'.

> The Messy GA, (MGA), uses a 'moving-locus' representation for genomes. Each gene is represented by a locus/allele pair. The bold genes (some desirable schema) in the GA representation are transferred to an example MGA representation in the second line of Figure 1. Moving the genes together in this way allows the MGA to transfer the schema intact during recombination.

Messy-GA-style recombination is simple a matter of creating an offspring C which is the sum of genes from both parents. This is the *splice* operation of the Messy GA (Goldberg 1989).

```
A: ((2,1), (6,0), (8,1))
B: ((1,1), (3,0), (5,0))
-------------------------------------------
C: ((2,1), (6,0), (8,1), (1,1), (3,0) (5,0))
```

Figure 3-9: 'Splice' recombination in the Messy GA.

> Two parents A and B represent partially specified strings using locus/allele pairs. The offspring, C, is simply the union of specified genes from both parents.

It should be noted that when the two parents involved in a splice operation are not overlapping in the genes that they specify this operation is in fact insensitive to the ordering of genes in the chromosome and accordingly may just as well be represented by the fixed-locus but partially specified representation shown in Figure 3-10.

```
A: -1---0-1
B: 1-0-0---
------------------------------
C: 110-00-1
```

Figure 3-10: An alternate representation of the splice operation in Figure 3-9.

> Here we represent unspecified genes, or *don't cares,* by "-" and the offspring is created by taking specified genes from either parent where available. This provides a fixed-locus, but partial-specification representation of individuals that effectively represents schemata explicitly. We shall shortly address methods to handle conflicts in specified alleles when they occur.

The representation of individuals and the combination operation shown in Figure 3-10, inspired by the splice operation of the Messy GA, is the basis of the combination operator we will detail in Chapter 7.

Competing conventions

Suppose, for the sake of argument, that we arrive at some method that can successfully evolve different useful schemata in different individuals, and moreover we have some means to represent where the schemata are in each individual so that they can be transferred as a coherent whole into the offspring. Even then we still have the issue of how to handle conflicting schemata. This is known, in general, as the ‘competing conventions’ problem. To take a scenario in a different domain, if we were evolving neural networks to control a robot, and both networks work reasonably well but by different methods (different conventions) - then if we take a subset of one network and a subset of another and put them together we are likely to end up with a network that does not function properly. If on the other hand, both networks worked by the same method but with only slight variations (same convention) - then it might plausibly be valuable to try a combination of partial networks.

The intuition that recombination between radically dissimilar individuals will likely produce non-functioning offspring, is in opposition to the intuition that recombination requires parents to be different in order to be worth doing at all - that is, to the extent that parents are similar, recombination between them has no effect, and to the extent that parents are dissimilar, recombination between them is likely to produce non-functioning offspring.

However, note that this apparent conflict is alleviated when the parents have the possibility of being ‘complementary’ - representing different roles within the problem as a whole - for example, by specifying for non-overlapping subsets of genes using a partially specified representation. This possibility is explicitly utilised in Cooperative Coevolution (Potter 1997). However, when individuals specify for different roles or tasks, some extra machinery is required to evaluate and compare them.

3.4.3 Evaluation of parts

In the Messy GA, individuals are partially specified subsets of genes. Accordingly, an objective fitness function that expects a completely specified subset of genes presents a problem. In some problems, partially specified individuals may be evaluated naturally, or by filling in unspecified values with some kind of default value. In other cases, especially in cases where there are important interdependencies between genes, this is not possible or does not give a useful indication of the value of a schema.

What we really want to know when considering the utility of a particular schema in a compositional system is whether it is useful in creating larger schemata or fully-specified strings. Accordingly, a schema should be evaluated in some context or ‘template’ of the remaining values. ‘Templating’ in the Messy GA simply means superimposing the schema of interest onto some other fully-specified string. In some versions of the Messy GA, this template was created from the assembly of other individuals in previous ‘rounds’ of evolution (Goldberg et al. 1993). In cooperative coevolution, an individual fulfilling one role within the problem as a whole, is evaluated in a ‘shared world model’ along with representatives of individuals fulfilling complementary roles.

These scenarios are reminiscent of the ‘rowers’ analogy in Dawkins (1976 (1989 edition, pp. 38-39)) - specifically, a set of independently replicating entities may thus become coadapted by virtue of evaluation in a shared domain. In cooperative coevolution this scenario is set up explicitly - but Cooperative Coevolution does not have any subsequent method to explicitly combine coadapted individuals together. In the Messy GA where the template has also been evolved, this scenario is also apparent but the opportunity for coadaptation is less explicit.

One interpretation of these scenarios then, is that individuals that cover different parts of the problem domain may become coadapted to one another and complement one another’s unspecified characteristics. This provides explicit opportunity for divide and conquer problem decomposition (see also Rosca 1997) and potentially in some cases, relief from the competing conventions problem.

3.4.4 Selection on parts: The credit assignment problem

The next issue is as follows: if a partial solution must be evaluated in some context - for example, an assembly of other partial solutions (a template or shared world model) - then how do we apportion fitness to the modules involved? This is known as the ‘credit assignment problem’. In genetic algorithms using crossover, (where the implicit modules are sections of chromosome and the context is the individual), the credit assignment problem is manifest as ‘hitch-hiking’ (Forrest & Mitchell 1993b). In the Messy GA, where the modules are partially specified individuals and the context is other ‘spliced-on’ individuals and the template, the credit assignment problem is manifest as ‘parasites’ (Goldberg et al. 1989). In

Cooperative Coevolution the credit assignment problem is avoided because different individuals are not joined together and so cannot ‘hitch-hike’ on the credit of others.

In general, there is no principled solution to hitch-hiking in the GA. In the Messy GA, ‘parasites’ are handled by the use of a two-phase model. In the first phase, individuals are not recombined (or sliced together) and are not allowed to grow in size - thus eliminating the possibility of parasites. And when this is relaxed in the second phase, it is assumed that only good schemata remain in the population.

Thus the credit assignment problem for coevolving modules in compositional processes is an issue that will demand a more general solution in our models.

3.4.5 Coevolution of parts: premature convergence/competitive exclusion and fitness sharing.

In general then, one way or another, compositional methods often involve the coevolution of complementary parts (and their assembly into wholes). In the Simple GA with crossover the parts are schemata, and the wholes are individuals. In the Messy GA the parts are the size-limited individuals of the first phase, and the wholes are the templates and/or the fully-specified individuals formed in the second phase. And in Cooperative Coevolution, the parts are represented by explicitly segregated species, and the wholes are represented in the shared world model.

In a general framework of coevolving modules, it is necessary that we allow individuals covering different parts of the problem domain to coexist - otherwise, we will not be able to search different combinations of them. Maintaining diversity in cooperative coevolution of modules presents a problem. If we apply the usual kind of selection that we use in competitive evolution then, under repeated selection, the type that is best on average will sooner or later competitively exclude all other types. In biological models, e.g. (Arthur 1987), competitive exclusion of one species by another, and one allele by another, (pp. 17-48), is in some cases counteracted by ‘frequency dependent fitness’ effects (p. 52). That is, in principle, as the frequency of one type increases this may have consequences for the fitness function of the species involved, and in some cases, may apply negative feedback on the frequency of that type. Under certain conditions this may enable stable coexistence of more than one type. In artificial evolutionary algorithms frequency dependent fitness effects are known as ‘fitness sharing’ (e.g. Goldberg & Richardson 1987). Fitness sharing comes in many varieties (e.g. ‘crowding’, Cavicchio 1970; De Jong 1975, ‘thresholding’ Goldberg et al. 1990; ‘implicit fitness sharing', Smith et al. 1993, 'niching', Deb & Goldberg 1989) - see Mahfoud (1995) for an overview) but the basic idea common to all methods is that the value of being a particular type is depressed as function of the number of other individuals of the same or similar type. Fitness sharing methods can in principle then permit different types of entities to coexist which is just what we need if we want to search combinations of them.

However, the problematic aspect of fitness sharing is that it requires us to define the identity of types or at least the similarity of types - and it is generally not clear what metric of similarity (genotypic, phenotypic, behavioural) is appropriate. In Cooperative Coevolution, the types are predefined and selection is separated so that one type cannot exclude another. In templating (Goldberg et al. 1989), similarity is sometimes based on genotypic Hamming distance.

In the general case, where we do not know the roles of individual *a priori*, or when we cannot assume that genotypic similarity is a good indication of performing the same functional role, we need a more general method of maintaining diversity.

3.4.6 The multi-dimensional interpretation of fitness and 'Pareto coevolution'

In previous research we have investigated issues of diversity maintenance in coevolution in simple abstract games (Watson & Pollack 2001b). These investigations have led us to identify several concepts that will be raised again later in our models:

- In coevolutionary games, it is often useful to consider a game as containing several different 'aspects of play'. For example, in chess playing there is only one objective, 'play good chess', but this may implicitly involve being good at end games, being good at controlling the centre of the board, being good at protecting the queen, etc.

- Generally, an aspect of play that is good in one context (against one opponent) is not necessarily good in another context (against another opponent).

- In some cases, the gradient of the fitness landscape leading to fitter types within an aspect of play can even be reversed depending on the selection of opponents. For example, in tennis, a fast service might be advantageous against most opponents, but against some other opponents, serving slowly might provide an advantage. This induces a condition of ‘intransitive superiority’, and is one cause of ‘cycling’ and ‘forgetting’ (revisiting the same parts of strategy space more than once) in coevolutionary dynamics.

- If there are many aspects of play, and different aspects have different importance against different players, the quality of a strategy in general should not be reduced to a single-dimensional value - a single fitness measure. For example, using the average fitness of a player over many different opponents loses significant valuable information about what that player is good at and what they are not good at.

- This leads us to consider coevolution as a multi-dimensional optimisation process. In this interpretation, each individual has a multi-dimensional fitness measure, each dimension corresponding to different aspects of play. However, we generally do not know *a priori* what those different aspects of play might be. However, loosely, we may imagine the performance of the player against different opponents to describe its quality in different aspects of play.

- In two-player competitive games, we can treat each different opponent as a representative of an aspect of play, and thus each individual in the population defines a value for one dimension in their multi-dimensional fitness (see Ficici & Pollack 2000). We use the term ‘Pareto Coevolution’ (Watson & Pollack 2000a, Ficici & Pollack 2001) to refer to the application of standard multi-objective algorithms (e.g. Fonseca & Flemming 1995) to multi-dimensional measures coming from other coevolving players (see below). The Pareto Coevolution idea has been applied and developed further in subsequent work - (Ficici & Pollack 2001, Noble & Watson 2001, De Jong & Pollack 2002, Bucci & Pollack 2002).

- In multi-player cooperative games, we may use different *groups* of coevolving individuals to provide different contexts - and in principle, each group/context represents a dimension of play. We develop this further in Section 7.3.

The basic selection model of normal coevolution is that an individual A may replace an individual B if it is better on average than B against a sample of other players. The basic selection model of Pareto coevolution is that an individual A may replace an individual B only if A gets a better or equal score than B does against each and every player in a sample of other players (and better against at least one). Thus Pareto Coevolution is basically the idea that the evolving population should approximate the Pareto front of non-dominated strategies and will thereby avoid the convergence of the population that is inherent in single-dimensional interpretations of fitness. In Chapter 7 we will describe the biological analogues of this idea coming from niching and population subdivision.

3.5 Test problems

In attempting to better understand what kind of adaptation is afforded by different kinds of mechanisms, a test problem that illustrates the distinctions and principles involved is invaluable.

The Building Block Hypothesis appeals to the notion of problem decomposition and the assembly of solutions from sub-solutions. Accordingly, there have been many varieties of GA test problems with a structure based on building blocks. Some of these problems use deceptive fitness functions to model interdependency between the bits within a block. However, very few have any model of interdependency between building blocks; those that do are not consistent in the type of interaction used intra-block and inter-block.

Before reviewing some of the well known building block problems let us define some terms. Firstly, we clarify that a *separable* problem is a problem which can be divided into sub-problems each of which has a fixed optimal solution that is independent of how other sub-problems are solved. We define a *decomposable* problem to be something more general - for now, a problem that can be decomposed into sub-problems - and, we specifically allow that these sub-problems need not be separable - i.e. the optimal solution to one sub-problem may be different according to the solution of another sub-problem. In this case we say that a sub-problem is *dependent* on another, or in the bi-directional case, the sub-problems are *interdependent*. We will define these terms more exactly later (4.3.1). Included in the term *decomposable* is the notion of identifiable component parts, i.e. *modularity*, so a system of uniformly related variables with no modular structure is excluded from the term decomposable. Next we look at the notion of hierarchy. The Building block Hypothesis describes the assembly of bits into blocks, blocks into bigger blocks, and so on to provide a complete solution - the hierarchical aspect of the process is clear in our intuitions. Our general notions of GAs, building blocks and the Building block Hypothesis all imply that the process continues over many levels *in the same fashion*. We define a *consistent* hierarchical problem as one where the nature of the problem at all levels is the same. In a hierarchically consistent problem, the difficulty of solving a sub-problem given the solutions to the sub-problems of the previous level, should be the same at all levels (see Watson & Pollack 1999a, 1999b). With these concepts in mind, we review some well-known GA test problems.

3.5.1 Building block problems

Whitley *et al*. (1995a and 1995b), provide a review of test-problems from the GA literature and summarize, “most common test problems used in the past to evaluate genetic algorithms lack” the inclusion of “strong non-linear dependencies between state variables.” That is, although, the contribution of a single variable to the overall evaluation may be non-linear, there may be no non-linear interactions *between* variables, and therefore the optimal value for each parameter can be determined independently of all other parameters.

There are several building block style functions - the Royal Road (RR) functions (Mitchell et al. 1992, Forrest & Mitchell 1993a), revised Royal Road functions (Holland 1993), (see also, Jones 1995), concatenated trap functions (Deb & Goldberg 1992a, 1992b), and others - that clearly emphasize a gross-scale building block structure. But, like the real-valued functions that Whitley investigated, these consist of concatenated blocks, each of which may be optimized independently in a cumulative fashion. The R1 version of the Royal Road problem (Forrest & Mitchell 1993a), for example, can be imagined as a staircase leading search in a stepwise fashion in the correct direction. In concatenated trap functions and the revised Royal Road functions, the fitness gradient leads search away from the solution to each block. But, whereas the bit-wise landscape is fully deceptive, the ‘block-wise landscape’ is fully non-deceptive. That is, within a block the bits are not separable but each block is still separable from every other block (i.e. the optimal setting for a bit remains the same regardless of the setting of bits in other blocks) and again the sub-solutions accumulate linearly. To continue the analogy, although each tread on the stair is inclined unfavorably the whole staircase still leads to the global optima.

The R2 function, (Forrest & Mitchell 1993a), and the revised Royal Road functions make hierarchy explicit. However the interactions between bits within the first level blocks are very different from the interactions between blocks in subsequent levels. Although the fitness contributions of the base-level blocks accumulate non-linearly because of 'bonus' contributions from higher levels, the blocks are still separable in that the optimal bit-setting for a block does not change.

3.5.2 Interdependency

Goldberg *et al*. (1993) acknowledge that they have "temporarily ignored the possibility of subfunction crosstalk". *Crosstalk*, is defined by Kargupta (1995) as "a property of both the algorithm and the landscape" and appeals to the idea that the algorithm may be misled in identifying building blocks by changes in fitness contribution. Accordingly, Goldberg et al. suggest that noise in the fitness function may be an appropriate model for crosstalk. But this interpretation of crosstalk does not provide the same kind of interdependency between blocks as the interdependency between the variables within a block. We favour a definition of building block dependency that is not reliant on the algorithm being used - i.e. the observation that the optimal bit-setting for a block is different in one context than in another is algorithm independent.

Whitley *et al*. (1995b) argue for test-problems that have non-separable components, and propose an "expansion" method for constructing scaleable non-separable functions from a non-linear base function of two variables. By selecting pairs of variables in an overlapping fashion and summing the results they create a non-separable function over an arbitrarily large set of variables. They also propose that using all possible pairs, rather than a minimal set of overlapping pairs, provides a method to vary the amount of inter-dependency. This notion has some similarities with the NK landscapes of Kauffman (1989, 1993). The NK landscapes certainly do model the interdependency of variables - the contribution of each of N variables is defined as a random look-up function of itself and k other variables. The expanded functions of Whitley are analogous to NK landscapes with k=2 (and with a repeated base-function instead of random lookup tables). This seems like a step in the right direction especially since the value of k, in some sense, defines the degree of interdependence.

But this is still not wholly satisfying for now we cannot identify meaningful building blocks. In the NK model (and the expanded function method) the dependencies form a uniform network; there is no modularity.

3.5.3 Other test problems for EAs

Holland (2000) describes ‘hyperplane defined functions’. One of the features of these is that high-order schemata that confer fitness contributions are built from pairs of lower-order schemata. This aspect of the construction technique is very similar to previous work of our own (Watson et al. 1998) that we develop here. Holland notes that the elementary schemata in the problem may overlap and may be conflicting but the importance of this point is not emphasised enough. Specifically, if the elementary schemata are not overlapping and conflicting then they are separable – the solution to a building block is independent of context. In the example that Holland gives, there are only 3 loci that have any conflicts – which means the problem is trivially solved by a simple hill climber. But more generally, by building the initial schemata out of random subsets of the variables and assigning random bit configurations we are unable to ensure that the problem is difficult, or control how difficult it is. Accordingly, Holland is unable to give a time to solution for the GA or accretive mechanisms on this class of problem. Also Holland’s notion of building blocks here assumes tight linkage (short defining length) whereas the modular interdependency concept we will develop is independent of gene ordering, and in fact not defined in terms of genetic linkage at all - rather it is defined in terms of epistatic dependency.

Other building block functions explicitly utilise the partial overlap of schemata to introduce a little building block interdependency – (e.g. Wiegand et al. 2001).

One other building block problem of interest is provided by Louis & Rawlins (1991). They describe a problem where a global optimum at “000…000111…111” can be arrived at by crossover from strings that correspond to local optima at all-0s and all-1s. Interestingly, they also discuss this problem in the context of multi-objective optimisation, which will feature heavily in later models we develop here. The interesting feature of this problem is that it rewards strongly competing schemata which are distant from the global optimum, the global optimum is an isolated needle and therefore difficult to find with an accretive mechanism, but crossover of the strings from the two local optima puts the global optimum within easy reach. The main restriction of this model is that the composition is only useful once – it is a two-module problem with no additional hierarchical levels.

3.5.4 Modularity and interdependency: non-separable modules

Some extended NK models include “NKC” models (Kauffman & Johnsen 1989) intended to model two coupled semi-independent landscapes of two semi-independent species, and “NKCS” models corresponding to several coupled landscapes/species. An NKC landscape is an NKCS with number of species, S=2. As in NK landscapes, the fitness contribution of each of N variables is dependent on its own state and the state of several other variables. But in NKCS models, the problem variables are divided into S non-overlapping (generally equal sized) groups of variables, and for each variable, K of the epistatically dependent variables are drawn from the same group and C are drawn from other groups. In this manner we may imagine that each group of variables describes an NK landscape, but each landscape is epistatically coupled to a number of other NK landscapes.

The idea here is that we may evolve one species on one landscape, and another species on another, but the movement of the first will influence the landscape of the second, and vice versa. For our purposes, we may also interpret this model as a single problem composed of interdependent modules, and the coevolving entities are coevolving modules specialising on different sub-problems. This scenario illustrates clearly the opportunity to model intra-genomic and inter-genomic dependencies in a unified model (as suggested in 2.5.5), and NKC landscapes have strong parallels with the model we develop in the next chapter.

However, there are some inadequacies in NKC models. First, all NK models use random epistatic interactions between variables - but not all types of epistatic interactions are difficult for accretive methods (this is explained in the next chapter where we will use a specific, and difficult, kind of epistatic dependency that enables us to control what the consequences of these dependencies are in terms of local optima, and the width of fitness-saddles - see 4.2.1). Second, the interdependency between modules is not a proper scaling-up of the interdependency between the original variables. Specifically, it is the state of individual variables that determines the epistatic fitness effects between variables within a module, but unfortunately, it is also the state of individual variables that determines the epistatic fitness effects between modules. In contrast, to be hierarchically consistent, the epistatic fitness effects between modules should be a function of the solution states of modules (not some subset of variables within the modules). This will become clearer as we develop our model of modular interdependency in the next chapter. Third, NKC models exhibit a single level of clustered dependencies or a two-level hierarchy. In order to understand repeated compositional operations we require a model that is scalable over many hierarchical levels.

We will discuss Simon’s (1969) notions of ‘nearly decomposable’ systems at length in later sections (Chapter 4, and 9.2.2). We also mention that the relationship of static evaluation functions (i.e. fitness functions) to dynamic models of complex systems will also be discussed later (9.2.1).

3.6 Summary

The accretive view of evolutionary change in Evolutionary Algorithms is common. But notions of compositional change based on sexual recombination or crossover are also well known though controversial. The intuition of the Building Block Hypothesis is one of problem decomposition but it has proved difficult to demonstrate a GA proceeding in this manner. However, if some form of GA could behave in this manner, in some class of problem, then the divide and conquer algorithmic advantage this provides is significant. This has led to many different varieties of GA utilising modularity in more or less explicit manners.

There are many difficulties that arise in the use of compositional mechanisms. Well-known problems for GAs using recombination include poor genetic linkage and premature convergence of the population. More generally, problematic issues include the representation, combination, evaluation, and coevolution of proposed modules.

Finally, we have reviewed some well-known test problems that have been used in the GA literature and we suggest that, in large part, the inability to clearly demonstrate a GA performing composition of schemata is due to inadequacies in our understanding of modules, and hence inadequacies in the test problems used. However, there are many valuable features of existing mechanisms and existing test problems that will be useful to us in formulating a clear model of compositional change in the following chapters.

3.7 Accretion and composition in evolutionary biology and evolutionary computation

Note that there are significant parallels between the issues and phenomena discussed in the background on evolutionary biology and those discussed here for evolutionary computation. Some concepts are the same in both disciplines: e.g. fitness landscape, ruggedness, smooth paths of adaptation, local optima, convergence and diversity. Other concepts have close analogues in each discipline. For interest, Table 3-3 outlines some of the analogies and parallels between the concepts and issues of accretion and composition in EB and EC.

Evolutionary biology concept	Evolutionary computation concept
Terminological analogues	
Accretive evolution	Greedy/Hill-climbing optimisation
Compositional evolution	Divide and conquer problem decomposition
Common compositional mechanism: meiosis and sexual recombination	Common compositional mechanism: crossover
Compositional mechanisms based on encapsulation: e.g. Endosymbiosis	Compositional mechanisms based on encapsulation: e.g. Automatic module acquisition and re-use
Symbiosis as a source of evolutionary innovation	Developing methods in coevolution and automatic module acquisition
Adaptive surface, adaptive peaks, fitness saddles	Fitness landscape, local optima, fitness saddles
Epistatic dependencies between alleles/genes	Frustration of variables
Irreducible complexity	Problems with strong and numerous interdependencies
Natural hybridisation is often non-viable	Competing conventions prevents successful crossover
Free recombination/no genetic linkage: e.g. genes on different chromosomes	Uniform crossover
Strong genetic linkage	One- or two- point crossover

Frequency dependent fitness effects	Fitness sharing
Common simplifications/assumptions	
Default population genetics model is single species in single niche	Default EA is one population and single fixed fitness function
Inter and intra genomic dependencies usually modelled separately: i.e. epistasis vs species interaction	Likewise, inter and intra genomic dependencies usually modelled separately: i.e. epistasis vs coevolution. (NKC model is an exception)
Coevolution related issues	
Organisms live in: i) a genetic, ii) biotic, and iii) physical (abiotic) environment	Schemata reside in: i) individuals in the GA, and in templates in the Messy GA, ii) and individuals reside in groups in Cooperative Coevolution, and iii) all these reside under an objective fitness function
When individuals live in a group containing fit individuals, parasitic 'free-loaders' may be promoted	When schemata are evaluated in an individual containing other fit schemata, then less useful, even detrimental, schemata may be promoted. (Credit assignment problem, Hitch-hiker problem)
If organisms reside in the same fixed niche (without frequency dependent fitness effects (e.g. resource dependence) then repeated selection inevitably results in competitive exclusion	If all individuals are evaluated against the same objective fitness function (without fitness sharing) then repeated selection inevitably results in population convergence
Different organisms are fit in different ways - not fit in an absolute sense, but they have a 'goodness of fit' to a particular balance of environmental features	In coevolution, different individuals are fit in different ways - not fit in an absolute sense, but they have a 'goodness of fit' to a particular distribution of coevolving individuals
In an ecosystem, different organisms may coexist because they have a different balance of adaptive traits from other organisms, and therefore reside in different 'niches' (often mutually defined by other entities in the environment). However, A may competitively exclude B if there is no niche in which B is fitter than A	In Pareto coevolution, different individuals coexist because they are not dominated by other individuals i.e. They have a different balance of abilities with respect to different coevolving individuals. However, A may replace B in the case that there is no opponent against which B is better than A (or no group context in which B is better than A).
Some of the most controversial issues in each discipline	
Does sexual recombination offer adaptive advantage?	What is crossover good for?

Shifting Balance Theory: a means to enable the evolution of ‘interaction systems’ and their assembly, rather than depend on the accumulation of individually useful mutations	Building Block Hypothesis: a means to enable the evolution of useful ‘building blocks’ and their recombination, rather than depend on the accumulation of individually useful mutations
SBT assumes genetic linkage is weak, and in normal circumstances, alleles are selected for only their additive effects, but Wright notes that if linkage were strong then co-adapted alleles could behave as if they were a single allele of a composite gene	Building Block Hypothesis assumes that genetic linkage is tight and therefore schemata can be transferred from one individual to another as a unit, but Holland notes that if linkage is poor, then schemata will be disrupted and promotion of schemata will fail
The possibility of more than one level of selection: nucleotide, gene, group of genes, individual, social group. c.f. Selfish Gene model, and Shifting Balance Theory.	Likewise: bits, schemata/building blocks/modules, individuals, cooperative groups. c.f. Building Block Hypothesis, and Schema Theorem.

Table 3-3: Analogies between accretion and composition in EB terms and EC terms.

Chapter 4 - Defining Modular Interdependency

The purpose of this chapter is to define a fitness landscape that is difficult for accretive mechanisms and easy for compositional mechanisms. Defining such a landscape will assist us in clarifying the different adaptive capacities of these alternate mechanisms. The landscape we define is built from a high degree of epistatic interaction between variables that produces significant ruggedness and numerous local optima—this prevents the effective operation of accretive mechanisms. However, the epistatic interactions are highly structured, creating modules and sub-modules of interdependent variables. This modular structure makes the landscape, in principle, amenable to problem decomposition and the effective operation of compositional mechanisms, as we will show in subsequent chapters.

We divide our discussion as follows. First we briefly discuss some philosophical provisos of the ‘problem/solution’ metaphor to biological evolution. In the next subsection we distinguish between types of epistatic interactions that are easy and difficult for accretive mechanisms - this analysis considers interactions between pairs of variables only. The following subsection addresses how larger networks of epistatic interactions might be arranged so that they are decomposable. This follows our intuition that compositional mechanisms may be able to manipulate systems of variables with modular dependency more effectively than accretive mechanisms and such a structure should therefore exemplify the adaptive potential of compositional mechanisms. The third subsection in this chapter uses these considerations to define a system of interdependent variables that are highly epistatic but also highly structured. Following subsections outline a method for constructing problems of this general class and a specific instance, Hierarchical-if-and-only-if that we use for experiments in following chapters. We outline some contrasts and similarities of this problem with existing test problems/landscape in the literature.

General properties of this problem class, its amenability to different fitness optimisation mechanisms, and some simple analytic results are given in the following chapter.

4.1 Philosophical provisos: The ‘problem’ for an evolving entity

We note that the notion of a ‘problem’ that an organism must solve, or a predefined landscape on which an organism may move, can be misleading premises in some circumstances. For example, in many cases it would be incorrect to assume that the niche in which an organism ultimately resides existed in a pre-defined sense before the organism concerned - (Lewontin 2000, Levins & Lewontin 1985). Similarly, in many cases, the niche to which an organism is fitted is fundamentally affected (perhaps effected) by the biotic context of other evolving organisms and it would therefore be incorrect to assume that the fitness landscape has a static form that exists independently of the organisms that reside in it (Lewontin 1982, 1983).

These provisos are well noted. If, as seems likely the case, the relevant constraints on an adaptive species are predominantly the result of biotic environment, then the fitness landscape presented to an organism at any one time is predominantly plastic and unconstrained in longer time scales, and does not pre-exist in any meaningful sense. Accordingly, if every aspect of the environment is plastic and somewhat intangible in this sense, then it is difficult to know how to accommodate the separation of organism and environment or proceed with adaptive explanations at all (Levins & Lewontin 1985).

We do not pretend to resolve such philosophical questions in this work. However, we offer at the end of this chapter, some comments on the compatibility of the adaptive landscape model that we define here with these points of view. In the meantime, we suggest that the adaptational stance provides one of those useful for understanding biological systems and phenomena, and we find the problem/solution metaphor to be a pragmatically useful one for progressing the issues we want to address.

4.2 Pairwise interactions

4.2.1 Pairwise interactions and fitness saddles

Sewell Wright (1931) stated that “the central problem of evolution... is that of a trial and error mechanism by which the locus of a population may be carried across a saddle from one peak to another and perhaps higher one.” This conception of evolutionary difficulty, and the now familiar concept of evolution as a combinatoric optimisation process on a rugged landscape (Wright 1967), are central in issues of evolvability, and the ability of adaptation to escape or otherwise avoid local optima—configurations of features where no small change in features will produce a fitter variant—is of central importance.

Ruggedness in a fitness landscape is introduced by the *frustration* of adaptive features, or *epistasis* when referring to the interdependency of genes – that is, it occurs when the ‘selective value’ of one feature is dependent on the configuration of other features. Fitness saddles are created between local optima. The simplest illustration is provided by a model of two features, **F1** and **F2**, each with two possible discrete states, A/a and B/b respectively, creating four possible configurations: **AB**, **Ab**, **aB**, **ab**. Figure 4-1 shows three kinds of interaction between these two features. In each case we are interested in the effect on fitness of different combinations of feature values.

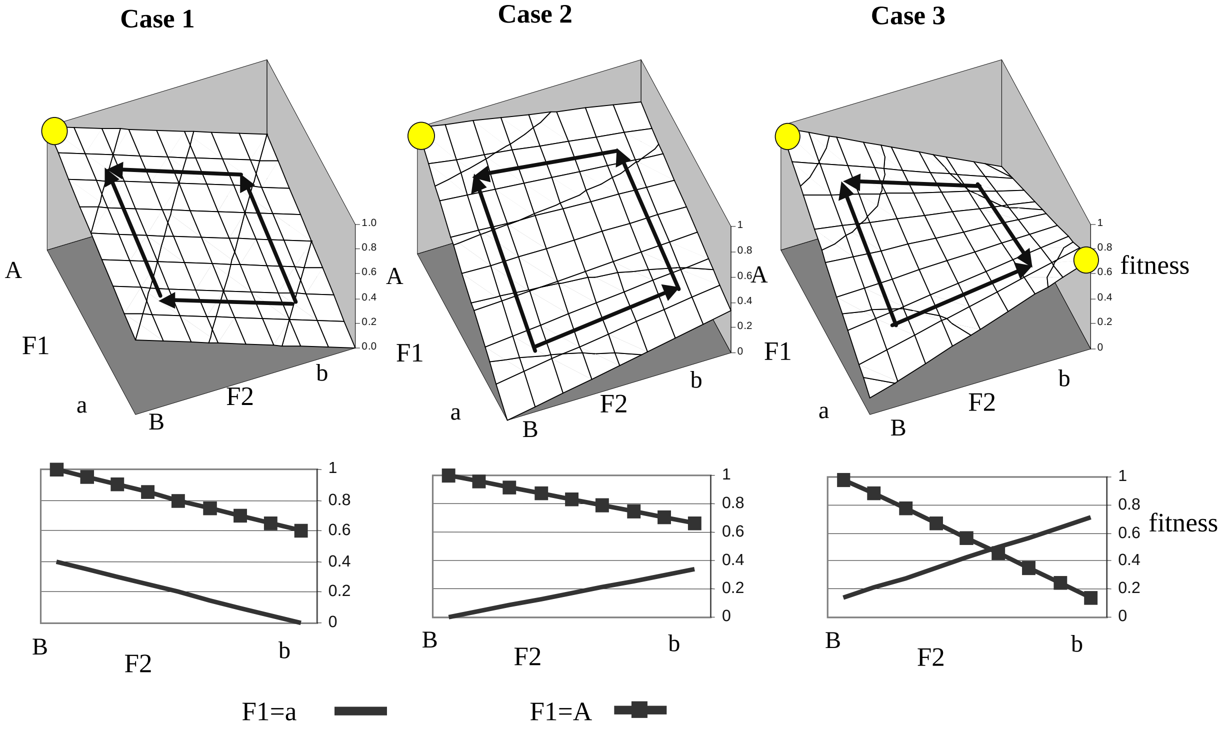


Figure 4-1: Different kinds of fitness interactions.

Case 1: no epistasis. The corresponding fitness landscape is planar. Arrows overlayed on the landscape show the paths of increasing fitness that can be followed by changing one feature at a time. Case 2: ‘easy’ epistasis; the preference of **b** over **B** is reversed depending on the value of **F1**, but **A** is always preferred over **a**. There is still only one optimum in this fitness landscape, shown by large dot. Case 3: difficult epistasis=’interdependency’. The preference relation between **b** and **B** is reversed according to **F1**, and the preference relation between **a** and **A** is also reversed according to **F2**. This creates a fitness saddle in the corresponding fitness landscape and two local optima - see also Figure 4-2.

In Case 1 the fitness effects of each variable are independent of the other. Case 2 shows some epistasis, but the resulting fitness landscape has only one local optimum and therefore no fitness saddle. Case 3 shows epistasis that creates two optima in the fitness landscape. That is, there is no path of increasing fitness from **ab** to **AB** for a mechanism that makes small changes to feature values. Escaping the local optima at **ab** requires a 'jump' switching both features at once (see Figure 4-2). Local optima only occur when the preferred value of **F1** switches according to the value of **F2**, and the preferred value of **F2** switches according to the value of **F1**. Other kinds of epistasis are solvable by incremental accumulation of single feature mutations. We define the *interdependency* of features to mean this problematic kind of epistasis, and adopt this as the basis of our models. Using this kind of epistasis, rather than random epistasis, (which may be either the easy kind or the difficult kind, in general) allows us to control what the consequences of these dependencies are in terms of local optima, and the width of fitness-saddles. Biological cases show that interdependency can be exhibited in natural systems (e.g. Lewontin 2000, p84).

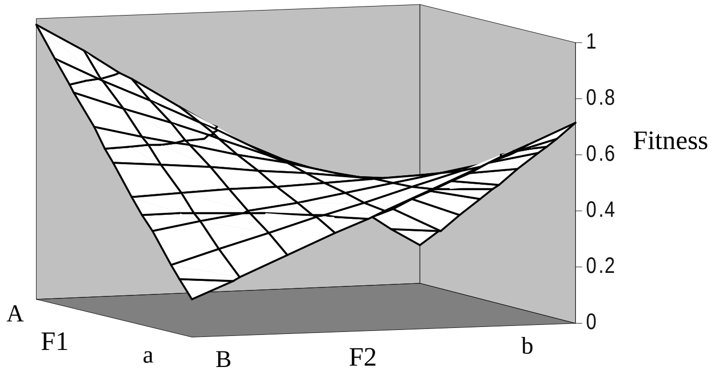


Figure 4-2: A fitness saddle created by interdependency between two variables.

This is a different view of Case 3 in Figure 4-1. This view shows the two local optima at AB and ab and the fitness saddle between them.

4.2.2 Defining ‘interdependency’

Let us define the ‘*preference for* ***a*** *over* ***A*** *in the context of* ***x***’, for a two-feature system as above, to be:

$$\mathrm{pref}(a,A,x) \equiv f(a,x)\text{-}f(A,x)$$

where f(p,q) is the fitness of the two variable system where F1 takes value p, and F2 takes value q.

If the preference for values of ***F1*** are the same for all contextual values provided by ***F2*** then there is no epistasis between these features and the fitness landscape is planar. i.e.:

$$\text{no-epistasis}(F1,F2) \equiv (\mathrm{pref}(a,A,b)=\mathrm{pref}(a,A,B)\ \textbf{and}\ \mathrm{pref}(b,B,a)=\mathrm{pref}(b,B,A)).$$

In other words, two features F1 and F2, have no epistasis if and only if the fitness difference between two values of ***F1*** is the same regardless of the value of ***F2***, and vice versa.

We now define ‘***F1*** *is dependent on* ***F2’*** as follows:

$$\mathrm{depends}(F1,F2) \Leftrightarrow \mathrm{sign}(\mathrm{pref}(a,A,b))\neq\mathrm{sign}(\mathrm{pref}(a,A,B)).$$

However, depends(F1,F2) does not imply depends(F2,F1), and, if F1 is dependent on F2, but F2 is not dependent on F1, then there are no local optima in the system.

However, ‘difficult epistasis’ between F1 and F2 is defined as follows:

$$\mathrm{interdependent}(F1,F2) \Leftrightarrow(\mathrm{depends}(F1,F2)\ \textbf{and}\ \mathrm{depends}(F1,F2)).$$

If we let the two features in the system be represented by Boolean variables then the Boolean function *equality*, or *if-and-only-if*, IFF, (and its negation, exclusive-or, XOR), define interdependency between two boolean variables. Other Boolean functions of two variables are not suitable since they either have only one solution, or solutions dependent on only one variable. We will be using the following function as the basis of discrete interdependency models in following subsections:

$$f(p,q)=(p\ \text{IFF}\ q)$$

where ***f***(<true>)=1 and ***f***(<false>)=0 as shown in the table below.

p	**q**	**f(p,q)=(p IFFq)**
0	0	1
0	1	0
1	0	0
1	1	1

Table 4-1: Truth table for if-and-only-if, IFF

An example interdependent system of two continuous-valued variables in the range [0,1] is defined by:

$$f(p,q)=pq+(1-p)(1-q).$$

This is the function illustrated in Figure 4-2.

4.3 Large-Scale interdependency structure

Having defined the basis for interdependency between a pair of features, the next step is to define a system of dependencies over more than two variables. A problem becomes more and more difficult for an accretive mechanism like a hill-climber to solve as the width of fitness saddles increases. The width of a fitness saddle is the separation of a local optima to the nearest point of equal or higher fitness. In multidimensional discrete spaces, systems of many discrete variables, this ‘width’ is measured in the number of variables whose state must change. So, if many variables need to change state all at once in order to get from a given configuration to a higher fitness configuration then this is difficult for accretive mechanisms.

We can create such a system over a large number of variables by overlapping pairwise interactions of the kind above. For example, let the fitness of three variables, p,q,r, be the sum of fitness contributions from two pairwise interactions as follows f(p,q,r)= f1(p,q)+f2(q,r) where each of f1 and f2 have the difficult dependency structure defined above. In this structure, p has a direct dependency with q and, indirectly through q, also has a dependency with r. Accordingly, the pairwise fitness contributions can be arranged such that in some cases a fitness increase in the overall system requires changing all three variables at once.

For example, if p, q, and r are binary variables we may define f1 and f2 in a table as follows.

x y	**f1(x,y)**	**f2(x,y)**
0 0	0.7	0.8
0 1	0.1	0.3
1 0	0.4	0.2
1 1	0.9	0.5

We then enumerate the configurations for the three variable system.

p q r	**f(p,q,r)**	**H**
0 0 0	0.7+0.8=1.5	none
0 0 1	0.7+0.3=1.0	1
0 1 0	0.1+0.2=0.3	1
0 1 1	0.1+0.5=0.6	1
1 0 0	0.4+0.8=1.2	1
1 0 1	0.4+0.3=0.7	1
1 1 0	0.9+0.2=1.1	1
1 1 1	0.9+0.5=1.4	3

Thus, in this example: f(010) < f(011) < f(101) < f(001) < f(110) < f(111)< f(000)

and the hamming distance to the closest configuration with higher fitness is given in the column headed “H”. The point of interest is that, in this example, the only configuration with a higher fitness than 111 is the configuration 000 and the width of this fitness saddle is three - i.e. the state of three variables must be changed at all at once to jump across this fitness saddle.

Let us extend this a little further. Equation 1 describes a system of N variables where every variable has pairwise dependencies with every other variable. In this system, for convenience, we define the interactions between all pairs to have the same pairwise fitness function i.e.

$$f(S_1, S_2, ..., S_N) = \sum_{i=1}^{N}\sum_{j=1}^{N} f(S_i, S_j)$$

Equation 1: Sum of pairwise interactions

where $S_1, S_2, ..., S_N$ are the variables of the system, and f(p,q) is the pairwise interaction of variables with states p and q. For illustration, let us define f(p,q)=(p IFF q), for binary variables p and q using our canonical base function IFF as shown in Table 4-1 previously.

With a little consideration, it is easy to see that when the number of variables with state=0 is greater than the number of variables with state=1, the change of any variable to state=0 will increase fitness, and vice versa. That is, the optimal state for any variable is to be in agreement with the majority, and any change in state that increases the size of the majority will increase the fitness of the system as a whole. Therefore the system has only two optima — i.e. all 0s and all 1s. The width of the fitness saddle between these two points is N, i.e. to jump from one to the other requires changing all variables at once. If we make the fitness of 1s, say, slightly higher than that of 0s in the f(p,q) function, then the all-0s optima is a local optima and escaping from this point to the global optima at all-1s requires changing all variables at once.

From a random starting configuration, a hill climber is almost equally likely to arrive at the local optima as the global optima, i.e. the size of their basins of attraction are in the ratio of f(0,0):f(1,1), so on average a hill-climber will become stuck at the local optima in almost half of the cases.

So, a construction of this kind, using many pairwise interactions, is sufficient to create broad fitness saddles and local optima that are arbitrarily difficult (depending on the number of variables in the system) for an accretive mechanism to escape from. But we have not yet discussed what properties of a system might make it easy for compositional mechanisms whilst still being difficult for accretive mechanisms.

4.3.1 Separability and decomposability to *modular interdependency*

The basic idea of compositional mechanisms is that they might be able to optimise and manipulate semi-independent subsystems in parallel and subsequently combine them together. Thus it is necessary for us to define exactly what form this kind of decomposable modularity might take.

The naïve approach to defining a modular structure is to make subsets of variables independent of one another (e.g. Royal Roads and Concatenated trap functions, see section 3.5). In this case, the fitness of a system of variables can be described as the sum of fitness contributions coming from two or more non-overlapping (disjoint) subsets of the variables in the system. For example, $f(s_1,s_2,\ldots,s_N)=f_1(s_1,s_2,\ldots,s_{N/2})+f_2(s_{N/2+1}, s_{N/2+2},\ldots,s_N)$.

However, using separable[21] modules that can be solved independently of one another in this manner is a mistake when trying to define scalable problems that distinguish the ability of accretive and compositional mechanisms. This is because when each module is independent of the others there is no need to manipulate modules as wholes since there are no dependencies between modules that need to be resolved. In fact, if an adaptive mechanism is capable of discovering fit configurations for modules of sufficient size, then fit configurations for all modules may be accreted serially. The inadequacy of separable modules for our purposes will become clearer when we have defined an alternative for comparison. For now, suffice to say that only when there is some form of inter-module dependency is there a need to manipulate modules as units and a corresponding advantage to compositional mechanisms. However, it is not at first obvious how a module might be identifiable and usefully manipulated as a whole if it has non-trivial dependencies with variables in other modules.

What forms might dependencies between two modules take? If the modules are separable then, regardless of the interactions within a module, the optimal configuration for one of the modules as a whole is independent of the configuration of the other module. If we do not want the modules to be separable then it should be the case that the optimal configuration for one module should depend on the configuration of the other module and vice versa. But if every configuration of one module defines a different preference for the configuration of the other module then the modules would have no reasonable meaning—there would be no progress that could be made in identifying fit configurations of a module that is independent of context. This is, we suggest, a paradox that has prevented the prior formulation of clear problem structures that are amenable to compositional mechanisms and hard for accretive mechanisms, and the lack of clarity here has caused considerable confusion in the issues around the utility of operators like crossover.

However, there is plenty of middle ground between the notion of a system that is not decomposable in any sense and one which has separable modules. In the discussion of pairwise interactions above, it is clear that a difficult interdependency is created when the preferred state for one variable is dependent on the state of the other and vice versa. When two variables are each separable from one another the preferred state for each variable is independent of the state of the other. When we are discussing only a pair of variables the

[21] We will define separability more rigorously shortly.

possible interactions between them are easily enumerated, see Figure 4-1, and the types of dependency that may be present are basically all-or-nothing; i.e. a pair of variables either exhibits the difficult kind of dependency, or they are easy. But when considering the interaction of two multi-variable modules there are more possibilities to be considered.

Consider the following. Let $C_A=\{a_1,a_2,\ldots,a_p\}$ be the set of possible configurations that module A may take and similarly, $C_B=\{b_1,b_2,\ldots,b_q\}$ be the set of possible configurations that module B may take, where $p=|C_A|$ and $q=|C_B|$. We want to know the configurations that have maximum fitness for each module. If there are no dependencies between modules then there is only one such configuration for each module regardless of the configuration of the other module. But if there are dependencies between modules then the maximum fitness configuration of A depends on the configuration of B and vice versa. So, let us define a new set, $M_{A,}$ to be the set of configurations of A that *have some configuration of B for which they are maximal,* and M_B similarly with respect to A. i.e.

$$M_A= \{m_1,\ldots,m_r\}: (\forall m(m\in M_A): m\in C_A) \wedge (\forall x(x\in C_B)\ \exists m(m\in M_A)\ \forall y((y\in C_A)\neq m): f(m,x) \geq f(y,x))$$

Equation 2: Set of maximal configurations

> Where {}:q is smallest set for which q is true, and f(s,t) is the fitness of the system when module A is in configuration s and module B is in configuration t.

Now, if $|M_A|=|C_A|$, i.e. every configuration of A is best for some configuration of B, then we have achieved nothing by proposing A as a module. But if $|M_A| < |C_A|$ then the possible configurations of A that might be maximal depending on the context provided by B is less than the total number of possible configurations for A, and we could conceivably reduce the number of possibilities we need to maintain in search independently of B. Put another way, regardless of the configuration of B, the members of C_A that are not members of M_A cannot be optimal and may be discarded from further consideration. Thus, whenever $|M_A| < |C_A|$ there is some utility in considering A as a module separate from B even though there may be dependencies between A and B.

Naturally, the smaller the size of M_A the better (for optimisation purposes), i.e. the less configurations need to be maintained and considered. Indeed the special case where $|M_A|=1$ is highly desirable in general because it means that the optimal configuration for A can be found independently of B and thus the problem can be decomposed into two separate sub-problems (see Figure 3-6). We use $|M_A|=1$ as our definition of a *separable* sub-module. Note however that this definition is more strict than the simple notion that the fitness of the system can be descried as the sum of fitness contributions from A and B. Specifically, it is possible that the *fitness contribution* of A changes with changes in the context provided by B (thus the fitness of the system cannot be described by a linear sum of that from A and B) yet the maximal fitness *configuration* of A may nonetheless be the same under all configurations of B.[22] It should be clear that it is changes in the maximally fit configuration that make optimisation difficult. Other types of non-linear contributions are relatively unimportant in general. Accordingly, note that our definition of a separable module does not require that there be no epistatic interactions between it and the rest of the system, just none of the 'difficult kind' (i.e. none like Case 3 in Figure 4-1 also Figure 4-2). Also, note that A may be separable from B and at the same time B may not be separable from A, i.e. $|M_A|=1$ does not imply that $|M_B|=1$. But, like the contrast of Case 2 in Figure 4-1 it is only when both A and B cannot be solved independently that local optima and fitness saddles are created between modules.

Decomposable but not separable = *modular interdependency*

So, the more general case, $1<|M_A|<|C_A|$, is the case of interest for our purposes. When $1<|M_A|<|C_A|$ the number of configurations of A that might be optimal given all possible configurations of B is greater than 1 but less than the total number of configurations that A might take. When this is the case, we can *reduce* the number of configurations of A that need to be considered (though we cannot reduce this number to one), and when $1<|M_A|<|C_A|$ and $1<|M_B|<|C_B|$ we will say that A and B are *decomposable* from each other, but not *separable* - or alternatively, that the system has *interdependent modules*.

Additionally, the distance between configurations in M_A from one another, controls the width of fitness saddles in the landscape.[23]

[22] This is the case in the second version of the Royal Roads problems, RR2, for example - i.e. even though there are epistatic interactions between blocks in this problem, the blocks are nonetheless separable by our definition. The utility of this characterisation is apparent in the observation that the inter-block epistasis in RR2 creates no local optima in the fitness landscape.

[23] Where *distance* and *width* are measured in the same metric - e.g. Hamming distance.

4.3.2 An example non-separable but decomposable system of four variables

To illustrate, consider the following system of four variables organised into two pairs (1,2) and (3,4):

$$g(S_1,S_2,...,S_N)=\sum_{i=1}^{N}\sum_{j=1}^{N}W_{ij}\,f(S_i,S_j)$$

Equation 3: The weighted sum of pairwise interactions over N=4 variables

where, as before, f(p,q)=(p IFF q), and W is a symmetric weight matrix indexed by i and j (1 ≤ i ≤ 4, 1 ≤ j ≤ 4) as follows:

i\j	1	2	3	4
1	0	4	1	1
2	4	0	1	1
3	1	1	0	4
4	1	1	4	0

Table 4-2: A weight matrix defining (with Equation 3) a system of two interdependent modules.

For the purposes of this illustration, we enumerate the possible configurations of the system and their fitnesses in Table 4-3.

C_{system}	f(system)	H[24]
0000	24	4
0001	12	1
0010	12	1
0011	16	2
0100	12	1
0101	4	1
0110	4	1
0111	12	1
1000	12	1
1001	4	1
1010	4	1
1011	12	1
1100	16	2
1101	12	1
1110	12	1
1111	24	4

Table 4-3: Configurations for a 4-variable system with their fitnesses.

Now let us calculate M_A.

C_A={00,01,10,11}

C_B={00,01,10,11}

When B=00, the configuration A=00 is maximal.

When B=01, the configuration A=00 or A=11 are (equally) maximal.

When B=10, the configuration A=00 or A=11 are (equally) maximal.

When B=11, the configuration A=11 is maximal.

Thus, M_A={00,11}.

Note that therefore the configurations A=01 and A=10 are never maximal, regardless of the configuration of B, and may be discarded from further consideration. Thus, since $|M_A|<|C_A|$, A is decomposable from B.

[24] We will discuss this column of the table later in the text.

But note also that A and B are not separable because different configurations of B require different configurations of A, i.e. $|M_A|>1$. In this example, the modules A and B are symmetric and therefore M_B also equals {00,11}.

Thus Equation 3 defines a system of four variables that, by our definition, is decomposable into two subsystems of two variables each, but these modules are interdependent and not separable. Inspection of the matrix W reveals why this is so. A couple of things are clear from this matrix: All variables have interactions with all variables, and; Each pair of variables has stronger interactions per variable within its own pair than it does across pairs. This follows the same structure as that proposed by Simon (p. 197-199) for what he terms a "nearly decomposable system".

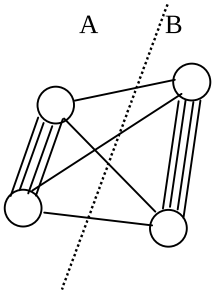


Figure 4-3: A dependency graph illustrating the dependencies of the four variables in Equation 3.

Nodes of the graph represent the four variables of the system, edges indicate an interdependency between twp variables, multiple lines indicate the strength of the interdependency, as per Table 4-2. The dotted line indicates a decomposition of the system into two interdependent modules A and B.

We will discuss briefly later whether this kind of interaction is feasible or natural in real world systems, but this is not our purpose in this chapter, nor are the claims of the thesis dependent on this. Here we are simply pursuing the definition of any system that is easy for compositional mechanisms whilst being hard for accretive mechanisms.

Our next issue then is to examine the properties of this example system with respect to the capacities of accretive mechanisms such as a mutation hill-climber. For this purpose we included the column headed "H", as previously, indicating the Hamming distance of each configuration to the nearest configuration with higher (or equal) fitness. Note that the fitness landscape has 4 optima for a single-bit mutation hill-climber: i.e. the points 0011, 1100, 0000, and 1111 have no points of higher fitness that may be reached by changing only one state variable. More precisely, 0011 and 1100 are local optima, and 0000 and 1111 are global optima. Note further that the two global optima are maximally distant from one another in Hamming space, and the two local optima are maximally distant from these and each other. In fact, this landscape has the most local optima that a four-variable system can have. See Figure 4-4.

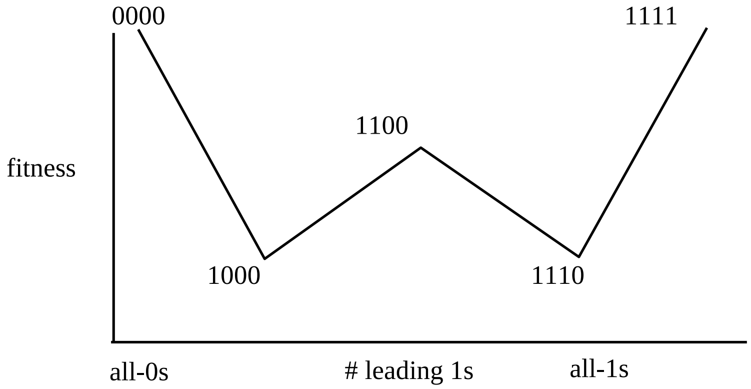


Figure 4-4: A particular cross section through the fitness landscape of Equation 3.

> This particular cross-section though the fitness landscape is useful in indicating the relationship of local optima in the system. This section is a one-point mutation walk from one global optimum to the other, moving to (one of) the highest fitness Hamming-distance-one configurations available at each step, and without revisiting any point (see Section 5.1). Specifically, this passes through the points (0000), (1000), (1100), (1110), (1111). This section is useful indicating the width of fitness saddles in the landscape - i.e. the Hamming distance from a local optima to the nearest point of higher (or equal) fitness. The global optima correspond to configurations where the two modules are in ‘agreement’ - the local optimum in the middle is created by either configuration where the two modules are internally in agreement but incompatible with each other -i.e. (0011) or (1100).

Note also that the members of M_A in this example, (00, 11), are distant from one another in Hamming space and therefore, despite the fact that this problem is decomposable (and we will show, easy for compositional mechanisms), it is difficult for accretive mechanisms to optimise either module since small changes will not allow a module to switch between the solution that is compatible with one context to the solution that is compatible with another (see Section 4.6.6). After generalising this model we will analyse its difficulty for accretive mechanisms more formally.

4.4 Generalised models

4.4.1 Aggregate effects of modules: dimensional reduction

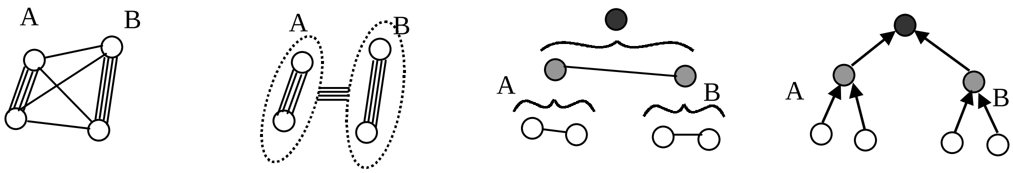
In this section we aim to better understand what it is about the system above that gives it the properties it has, better understand its structure, and to define it over a larger set of variables. We will find that there is a more compact description for systems of this type that uses an explicit abstraction of hierarchical modularity.

Simon (1969) suggests two properties for a "nearly decomposable system" that provide a useful starting point (p.198):

1. "the short-run behaviour of each of the component subsystems is approximately independent of the short run behaviour of the other components"
2. "in the long run the behaviour of any one of the components depends in only an aggregate way on the behaviour of the other components".

This is a dynamic description of properties that can also be understood by a static analysis of the dependency matrix W above (see 9.2.1). The first property is implemented in the dependency matrix by virtue of the fact that intra-module weights are greater than inter-module weights. The second is more subtle but equally important, and relates directly to our notion of decomposability that we defined above. Specifically, this property relates to the condition that $|M_A|<|C_A|$.

Conversely, when $|M_A| = |C_A|$, *every* configuration of A has sensitive dependencies with B. If this is the case then the module has no useful identity at the aggregate level. However, when $|M_A|<|C_A|$, we can describe the state of A as the state of a new 'aggregate' variable related to the cumulative state of the 'primitive' (i.e. the original, non-aggregate) variables in A. This new aggregate variable is sufficient to exactly describe the dependencies of A with the rest of the system. Let us illustrate.



i) ii) iii) iv)

Figure 4-5: Abstracting the interaction between modules.

i) The four variables of the system have intra-group (bi-directional and symmetric) interdependencies of strength 4 (shown by multiple lines) and inter-group interdependencies of strength 1 (single lines). ii) The inter-group dependencies can be abstracted into a single dependency of strength 4 acting between the ‘aggregate states’ of the two modules. iii) Alternatively, we may depict the aggregate state as a new higher-level variable, shown by grey nodes, and use the braces to depict the fact that these aggregate variables take their state from the aggregate effect of their respective subsets over the original variables. The black node indicates that, as we will see later, the state of this 4-variable system may contribute, in an aggregate way, to a larger system not shown. iv) Equivalently, we may depict the original variables as ‘feeding in’ to the state of the aggregate variables using arrows, to arrive at the style of figure used in the introductory chapter (see Figure 1-6), interpreting the grey nodes as phenotypic effects that are epistatically dependent on several genes. The black node is then a higher-level phenotypic effect dependent on the state of both subsystems.

Figure 4-5 we indicate how the inter-module interaction can be abstracted into the interaction of the ‘aggregate states’ of the respective modules. The abstraction can be implemented as follows. For every inter-module dependency, if the states of the variables involved in this interdependency agree with each other then the fitness of the system is increased, if they disagree then this dependency has no effect. Thus the inter-module interaction can be calculated from counting the number of agreements between the variables in A and the variables in B. This in turn can be calculated as (#0s in A)(#0s in B) + (#1s in A)(#1s in B). And in fact, since, with two-state variables, the number of 1s in a group is <group_size> minus #0s, and vice versa, we can calculate the interaction between groups given only the proportion of 1s in the group and a scaling factor.

Let us write the proportion of 1s in a group as *ones*A=(#1s/<group_size>), then the fitness contribution of the system coming from inter-group interaction may be written as W$\boldsymbol{f}'$(*ones*A, *ones*B) where $\boldsymbol{f}'$(p,q) = pq+(1-p)(1-q), and W is a weighting factor. Additionally, the original pairwise fitness function of two primitive variables, f(p,q)= (p IFFq) = $\boldsymbol{f}'$(p,q) when p and q are binary variables.

Then we may re-write exactly the system described in Equation 3 as follows:

$$g(S_1,S_2,S_3,S_4)=W_1 f'(S_1,S_2)+W_1 f'(S_3,S_4)+W_2 f'(ones\{S_1,S_2\},ones\{S_3,S_4\})$$

Equation 4: Re-writing Equation 3 using aggregate variables.

where, ***f'***(p,q) = pq+(1-p)(1-q), *ones*{p,q} is the proportion of 1s in the set {p,q}, and to match the weight matrix described in Table 4-2, W_1=W_2=8.

This formulation of the function makes it very obvious that the interaction between modules can be described as a simple function over aggregate properties of each module. Thus, since the aggregate properties 'lose information' about the module (i.e. the number of possible values that *ones*A can take is less than $|C_A|$) it cannot be the case that every state of B requires a different configuration of A, and thus it cannot be the case that $|M_A|=|C_A|$. This makes it clear that Simon's notion, i.e. that the long-term behaviour (or in this case long-range or inter-module dependency) of the system depends only in an aggregate way on other components, is closely analogous to our notion of decomposability based on the number of configurations that induce different maximal-fitness configurations in one another.

From these considerations, we find it useful to summarise the properties of a system with decomposable modules by noting that, although the modules are not independent, the 'interface between the modules has a low dimension'. By this we mean that the number of different interactions between modules ('different' with respect to their effect on maximum fitness configurations) is at most $|M_A|x|M_B| < |C_A|x|C_B|$. And we note that the description of the system using only the aggregate terms effects a *dimensional reduction.*[25] It is our intuition that compositional mechanisms will be able to exploit this dimensional reduction of a system, when available, by exploring combinations of modules instead of exploring combinations of the original basic variables, as we shall illustrate later.

[25] This notion of dimensional reduction, the idea that the system can be re-described more simply at a higher level of abstraction, is analogous to the methods of *renormalization groups* (Wilson 1979). Renormalization groups are integral in the analysis of complex physical systems, such as spin-glass systems, exhibiting complex dependencies and local optima in their energy functions.

4.4.2 Hierarchy and scaling-up

The above abstraction of the inter-module interaction into aggregate terms provides a convenient tool for abstracting and generalising our model system. Thus far we have developed from a system of two interdependent variables to a system of four interdependent variables (in two pairs). One way we can extend this model for increasing numbers of variables is by adding additional groups at the same hierarchical level, or increasing the number of variables within each module (in Simon's terms, we may increase the "span" at either hierarchical level of the system). Another way we may extend the system is by adding successive hierarchical levels. Systems of this type are easily extensible in either manner. We will extend the model with a focus on the latter, but the final form we arrive at is also extensible in the former manner.

To add a successive hierarchical level of modularity we define a system of 8 variables composed of two interacting subsystems of 4 variables. This can be achieved appropriately with the following weight matrix. For clarity we have also utilised the diagonal symmetry of the matrix and modified Equation 3 slightly to give Equation 5 accordingly.

$$g(S_1, S_2, ..., S_N) = \sum_{i=1}^{N} \sum_{j>i}^{N} W_{ij} f(S_i, S_j)$$

Equation 5: The symmetric weighted sum of pairwise interactions over N variables

where f(p,q)=(p IFF q) and W is the matrix below.

i \ j	2	3	4	5	6	7	8
1	16	4	4	1	1	1	1
2		4	4	1	1	1	1
3			16	1	1	1	1
4				1	1	1	1
5					16	4	4
6						4	4
7							16

Using the same method as described for the 4-variable system above, we can re-write Equation 5 as follows:

$$g(S_1,S_2,...,S_N) = \begin{array}{l} W_1(f'(S_1,S_2)+ f'(S_3,S_4)+ f'(S_5,S_6)+ f'(S_7,S_8))+ \\ W_2(f'(ones\{S_1,S_2\},ones\{S_3,S_4\})+ f'(ones\{S_5,S_6\},ones\{S_7,S_8\}))+ \\ W_3 f'(ones\{S_1,S_2,S_3,S_4\},ones\{S_5,S_6,S_7,S_8\}) \end{array}$$

Equation 6: 5 rewritten

where, for the weight values in this specific matrix W_1=W_2= W_3=16.

A note about the scaling of weight values

Note that although the strength of inter-module dependencies *per variable* is much less than that of intra-module dependencies *per variable*, the weighting of the *aggregate* inter-module dependency is the product of the per variable weight and the number of links between modules. The number of links between modules is the product of the numbers of variables in each module. The per-variable weight values in the matrix above are chosen to scale-down as the square root of the per-variable strength at the previous level as we ascend levels of the hierarchy. This is convenient as it makes the weights for inter-module dependencies turn out to be equal at all levels.

However, it should be noted that the choice of strength values is not in general arbitrary. It is important that the strength of inter-module dependencies is low enough that intra-module dependencies create local optima. That is, if aggregate inter-module dependencies are higher than a single intra-module dependency then a change in a single variable may increase the fitness of the system by placating its inter-module dependencies despite disrupting intra-module dependencies. If this is the case, then the ‘allegiance’ of the variables to the module is ‘over-ruled’ (see Section 5.1). On the other hand, if the ratio of inter-module dependency strength to intra-module dependency strength is too low then, in the limit, inter-module dependencies are simply insignificant and we may just as well treat the modules as independent. Choosing a scaling factor such that the aggregate inter-module dependency is about the same strength as each intra-module dependency at the level below, as we have, makes inter-module dependencies significant but not overpowering.

This issue is only a problem when inter-module dependencies are defined in terms of aggregate pairwise interactions, as they are here. When inter-module interactions are defined explicitly using higher-order schemata (as in Equation 11, later) inter-module dependencies may be made arbitrarily strong without ‘overpowering’ the effect of the intra-module dependencies.

Given our choice for this scaling factor for the weights in this example system, i.e. W_1=W_2= W_3, these weights may be omitted, as below. This produces only a constant scaling difference in the fitness of each configuration of the system.

4.4.3 General recursive form

Equation 6 can be further simplified into a recursive function that systematically divides the set of variables into subsets until we arrive at the atomic variables.

$$g(S_1,\ldots,S_N)=\begin{cases} f'(S_1,S_2) & ,if \quad N=2 \\ f'(ones\{S_1,\ldots,S_{N/2}\},ones\{S_{N/2+1},\ldots,S_N\})+g(S_1,\ldots,S_{N/2})+g(S_{N/2+1},\ldots,S_N) & ,otherwise \end{cases}$$

Equation 7: Recursive form of pairwise modular interdependency

where onesS is the proportion of 1s in the set *s*, and, as before, ***f'***(p,q) = pq+(1-p)(1-q). In this equation, the constant weighting for all terms has been dropped (producing only a constant scaling difference in the fitness of each configuration for the system with respect to Equation

5 / Equation 6). N=2^H where H∈Z^+ is the number of hierarchical levels in the system or subsystem.

It is also possible that the disjoint subsets be defined in a manner that is less restricted than exactly equal halves. Equation 8 shows a more general form of Equation 7 where the number of sub-modules per module is unrestricted.

$$g(S_1,\ldots,S_N)=\begin{cases} f'(S_1,\ldots,S_k) & ,if \quad N=k \\ f'(onesS^1,\ldots,onesS^k)+\sum_{i=1}^{k} g(S^i) & ,otherwise \end{cases}$$

Equation 8: Recursive form of ‘k-wise’ modular interdependency

where S_i is the ith variable of the configuration, S^i is the ith disjoint sub-partition of the variables, and *ones*S, ***f'*** are defined as per Equation 7. N=k^H where H∈Z^+ is the number of hierarchical levels in the system or subsystem, and k is the number of sub-modules per module.

4.4.4 Hierarchical consistency

We define a hierarchically consistent modular problem as one where the nature of the problem of finding modules at level h, given the solutions to modules at level h-1, is the same for all hierarchical levels h. An easy way to support hierarchical consistency is to construct a problem using general recursive functions for both fitness contributions and aggregations[26] - see ***F*** and ***T***, below. Then we may define base fitness functions and base aggregation functions, ***f*** and ***t***, that provide different variations for this general hierarchically decomposable form.

$$F(B)=\begin{cases} f(B) & if\ |B|=1, \\ f(T(B))+\alpha\sum_{i=1}^{k} F(B^i) & otherwise. \end{cases}$$

Equation 9: General Fitness function for Hierarchically Consistent Modular Interdependency

Where B is a block of variables, |B|=k^h is the size (number of variables) in B, B^i is the ith disjoint sub-partition of the variables in B, α is a scaling factor, and N=k^H where H∈Z^+ is the

[26] An aggregation function *transforms* a block of variables into a single variable.

number of hierarchical levels in the system or subsystem, and k is the number of sub-modules per module, and ***T*** is the general aggregation function defined below, and ***f*** and ***t*** are base fitness and aggregation functions.

$$T(B) = \begin{cases} t(B) & if\ |B| = k, \\ t(T(B^1),...,T(B^k)) & otherwise. \end{cases}$$

General Aggregation function

The specific function defined in Equation 8, can then be defined using Equation 9, with k=2, α=1, ***f***(p,q) = pq+(1-p)(1-q), and ***t***(p,q)=(p+q)/2. Note that even if the primitive variables are binary, this aggregation function, ***t***, will approximately double the number of possible values that aggregate effects may take at each hierarchical level.

This general recursive form of ***F*** and ***T*** is useful in several respects: a) it makes it very clear that the resultant problem definition is hierarchically consistent; b) it conceptually separates the fitness contribution of a block from the aggregation function; c) it allows us to describe a broader class of systems with hierarchical modular interdependency by substituting alternate base functions ***f*** and ***t***.

4.4.5 Alternate base functions

It is important that ***f*** and ***t*** have (at least) the following properties in order to define appropriate interdependent modular structure:

a) ***f*** must incorporate interdependency between at least some of its variables (see 4.2.1) if the overall problem is to have any local optima/ fitness saddles, and be difficult for accretive mechanisms.

b) It is also important that ***t*** be a function that reduces the dimensionality of the arguments it acts on (see 4.4.1) in order that inter-module interaction can be described via aggregate effects. That is, the number of possible values that ***t***S may take must be less than the number of configurations that S may take if configurations are discrete. In the above example, the number of values that ***t***S can take is k+1 where k is the number of variables in S, but the number of configurations that S can take is 2^k. Alternatively, if the variables are continuous, it may be sufficient to reduce more than one variable into one variable: i.e. ***t***: $R^k \rightarrow R$, or ***t***: $Z^k \rightarrow Z$.[27] The last stipulation for the properties of ***t*** is that, it should (approximately) re-scale its outputs to be the same as each of its arguments (if ***f*** is sensitive to scaling), or else α may be adjusted to compensate. This will ensure that aggregate inter-module dependencies will be significant but not over-powering (see note in section 4.4.2).

c) There are additional constraints between ***f*** and ***t*** that must be satisfied in order that the resultant landscape is amenable to compositional mechanisms. For example, it must be the case that the high-fitness configurations within a module correspond to aggregate values that are involved in high-fitness values of inter-module interaction. As a counter-example, if we defined ***t***(p,q)=(p XOR q) and ***f*** (p,q)=(p IFF q) then these functions would satisfy the properties listed above, and $|M_s| < |C_s|$, but the high-fitness single-module configurations would lead search to favour configurations that aggregate into low-fitness multi-module configurations.[28]

The above method of constructing a hierarchically decomposable can be further relaxed to encompass a still broader range of systems with modular interdependency. For example, it is not necessary that the number of sub-modules per module be regular throughout the system. Similarly, it is not necessary that ***f*** and ***t*** be defined uniformly over all modules at all levels as they are in Equation 9, but it not so easy to determine what the overall effect will be when different functions are combined. Pelikan and Goldberg (2000) give a problem definition including these relaxations.

The recursive construction of Equation 9, and the above guidelines for relaxing ***f*** and ***t***, indicate a broad class of hierarchically decomposable systems that, as we will clarify, are difficult for accretive mechanisms

[27] A simple way to ensure appropriate reduction is to use a function ***t*** that is insensitive to the ordering of its arguments (like counting the number of 1s, or summing)—this ensures that $|M_s| < |C_s|$, since some configurations for *s* will be indistinguishable at the aggregate level.

[28] One means to select appropriate functions is to use functions where low-fitness configurations within a module are those where the variables in the module ‘cancel out’ each other’s inter-module effect. In other words they correspond to aggregate values that have no inter-module effect on fitness. More formally, we can state that F becomes separable, i.e. F(T(B1),T(B2)) approaches f_1(B1)+f_2(B2), as either F(B1) or F(B2) approaches a low fitness value. For example, in ***f***(p,q) = pq+(1-p)(1-q), as we use above, a low fitness configuration for (p,q), such as (0,1), corresponds to an aggregate value of 0.5. The value 0.5, when passed up to inter-module interactions, results in ***f***(x,0.5) = 0.5, thus 0.5 can be seen as a kind of ‘null’ aggregate value that is insensitive to inter-module configurations. This ensures that our aggregation function behaves suitably with our fitness function, i.e. eliminating low-fitness modules from consideration will not eliminate high-fitness multi-module configurations.

and easy for compositional mechanisms. The particular instances of ***f***(p,q)=pq+(1-p)(1-q) and ***t*(**p,q)=(p+q)/2 provide a useful example system. With some care, less restricted systems may be also be defined in the style of Equation 3. The verification that a proposed system is in the class of systems exhibiting modular interdependency is given in 4.3.1.

4.5 Hierarchical-Equality and Hierarchical-if-and-only-if (HIFF)

In this section we define a simple instance from the general class of functions described above that acts on discrete variables and will be easy for us to analyse. This function utilises the hierarchical construction of Equation 9 and is based on f and t below:

f(p,q)=(p IFF q).

t(p,q)=0 if p=q=0, and 1 if p=q=1, and 'null' otherwise.

More generally, we can define f and t over k variables using configurations where all variables have equal values, as follows:

f(p_1,…p_k)=1 if ($\forall_i$: p_i=***s***), and 0 otherwise, where ***s*** is any value from a discrete set of valid values (e.g. 0 or 1 for binary variables).

t(p_1,…p_k)=s if ($\forall_i$: p_i=***s***), and 'null' otherwise, where ***s*** is any value from a discrete set of valid values (e.g. 0 or 1 for binary variables), and 'null' is a special invalid symbol.

Notice that, following the guideline (c) above, configurations of a block that are not high-fitness aggregate into 'null' values that therefore cannot be involved in high-fitness meta-blocks. This ensures that finding good solutions to a block is useful in finding fit solutions to a meta-block. Configurations involving any null value confer no fitness contribution under ***f*** above. One of the main differences between this function and the ones defined previously is that the aggregation function returns one of {0,1,'null'} for all modules at all levels - i.e. the number of values that an aggregate variable may take is the same at all hierarchical levels. Moreover, the only way to attain *any* inter-module fitness contribution in this function is to have all variables within each module agree in their states. In contrast the previous definitions of ***f*** and ***t*** responded to a degree of agreement. (This contrast is indicated in Figure 5-4). This has the effect of introducing explicit high-order interdependencies into the function, as we will briefly discuss shortly.

This particular instantiation of ***f*** and ***t*** allows us, with a little manipulation, to embed ***t*** within ***f*** and dispense with any explicit description of **t** altogether, as shown in Equation 10 below.

$$g(S_1,\ldots,S_N)=\begin{cases} 1 & ,if\ \ N=1 \\ f(S_1,\ldots,S_k)+\sum_{i=1}^{k} g(S^i) & ,otherwise \end{cases}$$

Equation 10: Hierarchical-Equality no-scaling

> where S_i is the i^{th} variable of the configuration, S^i is the i^{th} disjoint sub-partition of the variables, *f* is the base fitness function, defined below. $N=k^H$ where $H\in Z^+$ is the number of hierarchical levels in the system or subsystem, and k is the number of sub-modules per module.
>
> ***f*** $(p_1,\ldots p_k)=1$ if $(\exists s \forall i\colon p_i=s)$, and 0 otherwise, where s is a member of a discrete set of allowable values for the problem.

In this simplified function, ***f*** is intrinsically discontinuous. Specifically, a module does not confer any fitness contribution unless *all* variables are in agreement. This means that the resultant function cannot be described in terms of the sum of pairwise interactions, as in Equation 3/Equation 5, and fitness contributions of higher-order schemata are necessary to define the function accurately. This also has the consequence that we can, if we wish, change the relative importance of intra- and inter-module interdependencies without the risk of intra-module dependencies being ‘overpowered’. These differences do not affect the number of local optima, the relative position of local optima or therefore the width of fitness saddles, or the decomposability of the system. But these differences do make it easier to analyse the resultant fitness landscape (because the relative fitness of different configurations is more regular (see Section 5.1).

In our own previous work (Watson et al. 1998, Watson 2001, Watson & Pollack 1999a, 1999c, 2000b, 2001a, 2002), and consequently, in the previous work of others (Knowles et al. 2001, Oates & Corne 2001, Van Hoyweghen et al. 2001a, Van Hoyweghen & Naudts, 2001, Jansen & Wegener 2001, Wiles et al.

2001), a slightly different variant of this function was used that has a different scaling for the fitness contribution of modules. Specifically, in this variant, the fitness contribution of a module is equal to the number of variables in the module. This function is known as “Hierarchical-if-and-only-if”, HIFF. We will continue to use H-IFF in order to align with the existing literature.

H-IFF, and also a more general form, “Hierarchical-Equality”, (H-Equal), and both are defined by Equation 11. H-IFF is the particular case for binary variables, i.e. $s \in \{0,1\}$ and where k=2; H-Equal is the general case allowing any number of sub-modules per module and any base for the constituent variables. The only difference in Equation 11 when compared to Equation 10 is that the fitness contribution for a block is multiplied by the size of the block, i.e. N. HIFF and H-Equal also assume, by default, that the sub-partitions of the variables corresponding the sub-modules are equal-sized consecutive blocks, as below.

$$g(s_1,...,s_N) = \begin{cases} 1 & ,if \quad N=1 \\ Nf(S_1,...,S_k) + \sum_{i=1}^{k} g(S^i) & ,otherwise \end{cases}$$

Equation 11: Hierarchical-if-and-only-if (HIFF) and Hierarchical-Equality (H-Equal)

> where S_i is the i^{th} variable of the configuration, S^i is the i^{th} disjoint sub-partition of the variables, i.e. for ‘unshuffled’[29] HIFF with equal sized sub-modules, $S^i=(s_{1+k(i-1)},\ldots,s_{ki})$, ***f*** is the fitness contribution function, defined below. $N=k^H$ where $H \in Z^+$ is the number of hierarchical levels in the system or subsystem, and k is the number of sub-modules per module.
>
> ***f***$(p_1,\ldots p_k)=1$ if $(\exists s \forall i: p_i=s)$, and 0 otherwise, where s is a member of a discrete set of allowable values for the problem, by default $s \in \{0,1\}$.

HIFF can also be defined using HIFF(B)=N.g(B), using g(B) from Equation 9 with ***f*** and ***t*** as defined at the beginning of this subsection, k=2, s={0,1}, α=1/k, and equal-sized sub-partitions of the variables at each level.

[29] We will also consider random linkage or ‘shuffled’ versions of HIFF in later experiments.

4.6 Discussion

4.6.1 Symmetries and variations

The HIFF function will be used in the Chapters 5, 6 and 7 in our simulation experiments. HIFF has a very regular and symmetric structure but in some respects this regularity is immaterial to the experiments that follow. For example, none of the algorithms we illustrate depend on trans-locational symmetries - that is, it is never assumed that the configuration that optimises one module will be related to the configuration that optimises another, and that therefore, a schema discovered at one partition can be relocated to another partition. Accordingly, we may put candidate configurations through a random map, e.g. an XOR with a pre-determined random string, effectively randomising the absolute locations of the optima, without affecting the results. For some results, it is also not important that the global optima, (or the sub-strings they contain), are complementary strings (see Watson 2001).

Features such as whether the sub-modules are equal-sized (same ‘span’ in each module), and whether the hierarchical interdependency structure is ‘balanced’ (same hierarchical depth of sub-modules in each module), are also not the features of interest. What should be retained is the property of *modular interdependency* as defined in 4.3.1.

4.6.2 The HIFF landscape and natural hierarchy

HIFF is used in our experiments to exemplify the class of adaptive landscape in which the evolvability of composition can be contrasted with the evolvability of accretive evolution. We do not claim that HIFF is representative of the structure of adaptive landscapes in general. However, the problem of defining appropriate models for adaptive landscapes is an open one and, in passing, we note that HIFF exhibits some interesting landscape characteristics with respect to hierarchy in natural systems (Simon 1969), (see 9.2.2). In particular, dynamical systems exhibiting an interdependency structure that is similar at many scales might be a natural product of self-organized dynamical systems—as evidenced by ‘power law’ signatures in their dynamics (e.g. Bak 1996). Then, to the extent that natural adaptive landscapes are the result of such systems, scale-invariant fitness landscapes, such as that which HIFF defines, might not be entirely hypothetical.

4.6.3 Other hierarchical modular interdependency functions

Pelikan & Goldberg (2000) defined a function based on a hierarchically nested trap function with k=6.

Specifically, $\boldsymbol{f}(s_1,s_2,\ldots,s_6)=$

$$
\begin{array}{ll}
1.0 & \text{if } \boldsymbol{ones}(s_1,s_2,\ldots,s_6)=0/6,\\
0.0 & \text{if } \boldsymbol{ones}(s_1,s_2,\ldots,s_6)=1/6,\\
0.8 & \text{if } \boldsymbol{ones}(s_1,s_2,\ldots,s_6)=2/6,\\
0.9 & \text{if } \boldsymbol{ones}(s_1,s_2,\ldots,s_6)=3/6,\\
0.8 & \text{if } \boldsymbol{ones}(s_1,s_2,\ldots,s_6)=4/6,\\
0.0 & \text{if } \boldsymbol{ones}(s_1,s_2,\ldots,s_6)=5/6,\\
1.0 & \text{if } \boldsymbol{ones}(s_1,s_2,\ldots,s_6)=6/6.
\end{array}
$$

Like early work of our own (Watson et al. 1998), they also defined aggregation (“interpretation”) recursively, i.e. $\mathbf{t}(s_1,s_2,\ldots,s_N)=\mathbf{t}(\mathbf{t}(S^1),\mathbf{t}(S^2),\ldots,\mathbf{t}(S^k))$.

Unfortunately, in this paper the authors mistakenly[30] define the aggregation function as the majority of its arguments - i.e. $\mathbf{t}(s_1,s_2,\ldots,s_6)=0$ if $\boldsymbol{ones}(s_1,s_2,\ldots,s_6)\leq 0.5$, 1 otherwise.

In this form, any configuration of variables in a sub-module of any size will be either 0 or 1 (there are no “nulls”). This means that even random search can find either the 0 or 1 solution for a module of any size in a couple of guesses, and the ‘deceptive’ aspects of the trap function would actually lead a hill-climber to a point where it would be able to switch between the 0 and 1 solution by changing only one bit. This property makes the problems hierarchical structure degenerate in the sense that it is not necessary to search combinations of small modules to find large modules. However, an alternative definition, e.g. $\mathbf{t}(s_1,s_2,\ldots,s_k)=p$ if $(\exists p \forall i\colon s_i=p)$, and “null” otherwise, (see Watson et al. 1998, as used above) appropriately maps the arguments of **t** onto an aggregate value. In this revised form, high-fitness (non-null) configurations are exponentially rare in the size of the module. When defined this way (see Pelikan & Goldberg 2001) the problem is an interesting difficult variety of hierarchical modular interdependency.

4.6.4 Comparison with Royal Roads, concatenated trap functions, and NKC landscapes

Both versions of the Royal Road functions have separable building blocks. Similarly, the blocks in concatenated trap functions are also separable. In each case, the difficulty of solving a block is significant,

[30] Goldberg, personal communication, July 2000.

but since there is only one best-solution for each block (regardless of the context of other blocks, i.e. |M|=1, see 4.3.1), there is no need to search combinations of blocks to find the solution to the whole problem. In contrast, the blocks in HIFF have significant inter-module dependencies creating a problem that is hierarchically consistent (Watson & Pollack 1999a). Thus, having found the solution to the first level of blocks, an algorithm must subsequently search combinations of blocks at the next level, and so on. In HIFF, an algorithm that can manipulate and recombine building blocks is valuable, but in the previous building block problems this capacity is not required (hence Royal Roads and concatenated trap functions can be solved by a ‘macro-mutation hill climber’, Jones 1995).

As indicated earlier, the main differences between NKC landscapes are threefold:

1. All NK models use random epistatic interactions between variables - but not all types of epistatic interactions are difficult for accretive methods.

   HIFF uses a specific, and difficult, kind of epistatic dependency that enables us to control what the consequences of these dependencies are in terms of local optima, and the width of fitness-saddles (see 4.2.1).

2. The interdependency between modules in NKC is not a proper scaling-up of the interdependency between the primitive variables. Both are implemented in terms of pairwise interactions between the primitive variables, and since the epistatic dependencies are random, there is no means to regulate whether the aggregate effect of many inter-module dependencies act in unison to create large fitness saddles, or ‘average-out’ and degenerate into a random landscape.

   In contrast, HIFF is hierarchically consistent; the epistatic fitness effects between modules are a function of the solution states of modules.

3. NKC models exhibit a single-level of clustered dependencies, or equivalently, a two-level hierarchy.

   In HIFF, we can test the effect of repeated compositional operations since HIFF is scalable over many hierarchical levels.

We outline these differences, not to criticise NKC models, but to enable us to understand what the critical differences are between HIFF and existing models. Moreover, a particular interpretation of NKC models is very useful in interpreting the nature of the HIFF landscape, and is useful for addressing some aspects of the philosophical provisos we mentioned at the beginning of this chapter, specifically, the notion of coupled fitness landscapes - discussed shortly.

4.6.5 Modular interdependency as a hierarchical cooperation problem

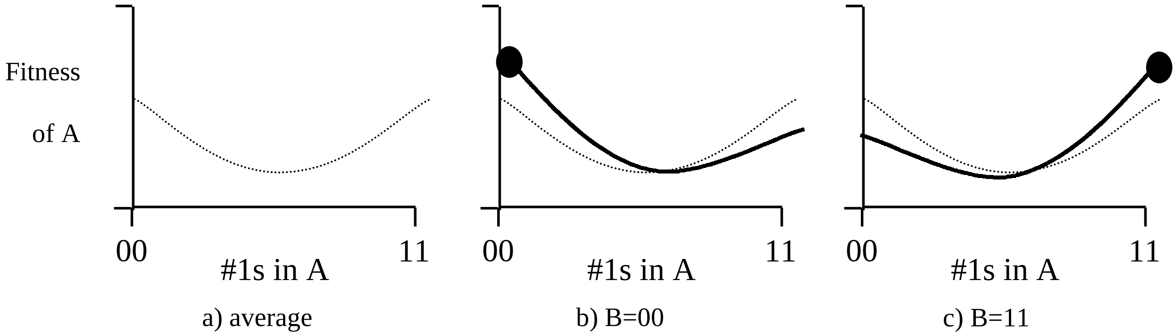
An alternative interpretation of the two-feature epistasis model in Section 4.2.2 is obtained by viewing the two different features as two different players in a symmetric two-player game, and the feature values as their possible strategies. In this view, the fitness contributions become the values of a pay-off matrix and the salient characteristic of Case 3, interdependency, is that the optimal strategy for player one is dependent on the behaviour of player two, and vice versa. The epistasis model we arrive at using (p IFF q) is analogous to the ‘mutual benefit’ matrix from (Maynard Smith and Szathmary 1995, p.262), but here there is not yet any distinction between the two attractors of the system i.e. which of (11) or (00) is the ‘defect’ and which is the ‘cooperate’ strategy, because we assign them equal value.

As we recursively re-apply the two-feature model we apply the two-player matrix in a recursive fashion to define a four-player game. Note that now, in the context of (00), (11) is a ‘defect/defect’ result for the other module, because it is in their selfish interest for each player not to change from this strategy, but if they both changed to (00), this would provide a higher payoff. Conversely, in the context of (11), (00) is ‘defect/defect’ and (11) is ‘cooperate/cooperate’. In other words, whether a strategy provides mutual benefit or not depends on the context in which the game is played.

Thus HIFF describes a hierarchical cooperate/defect game. The nature of the pay-off values is such that maximising the payoff for all (e.g. 128) players is achieved when two-subgroups (of 64) players are compatible. Other attractors in the evolutionary game occur when particular subsets of players are compatible intra-group but not inter-group. Accordingly, optimising HIFF requires the induction of hierarchical cooperation. The pay-off values at every level of resolution help to identify good combinations of strategies—but, which of the two optima at every level is best does not become clear until the context of other players is stabilised. HIFF deliberately dissolves the distinction between epistasis (the interdependency of genes within an individual) and multi-player evolutionary games (the interdependency of features of one entity with those of another) as is required from a model incorporating composition and the resultant changes in the unit of selection.

4.6.6 How changes in Module A affect the fitness landscape for Module B

Let us consider the adaptation of the system with two modules, A and B, in Section 4.3.2, as though each module were being evolved semi-independently on separate but coupled fitness landscapes. The fitness of the system for different configurations of A, in the context of different configurations of B is shown in Figure 4-6



a) average

b) B=00

c) B=11

Figure 4-6: The fitness landscape for A, for different configurations of B.

Each figure shows the fitness for module A when in different configurations, specifically (00), (01)/(10), (11). a) The fitness of A averaged over all configurations of B. b) The fitness of A when B=(00) is shown in the solid curve, and the optimal configuration of A=(00) is indicated by the dot. c) The fitness of A when B=(11) is shown in the solid curve - note that the optimal configuration of A has moved to (11).

This figure shows that the fitness landscape of A changes radically in some respects as B changes configuration - specifically, the optimal configuration of A when B=00 is the complete opposite of the optimal configuration of A when B=11. So, an entity evolving the values of A would find itself in a landscape that is radically dependent on the evolution of B. However, note that in other respects, A’s fitness landscape is insensitive to the configuration of B - specifically, in respect of the configurations of A that are *not* optimal, i.e. 01 and 10, are the same regardless of B. This is the property of modular interdependency that makes it amenable to dimensional reduction and thereby easy for a mechanism that can manipulation modules yet difficult for accretive mechanisms.[31]

4.6.7 Relation to philosophical provisos

Notice that from the above observations, the fitness landscape resulting from modular interdependency is not wholly incompatible with the notion that an organism's fitness landscape is radically dependent on the state of its environment. In Section 3.4.3 we introduced the perspective of compositional mechanisms as the coevolution of modules, and this is a view that we will be developing further in the compositional models that follow. In this perspective the particular adaptive landscape presented to an organism at any one point in time is some *subspace* of the landscape, and although the landscape *as a whole* is fixed, the subspace occupied by a particular entity is radically dependent on the state of other evolving entities at that time. Moreover, we will not predefine what roles or niches an entity is allocated - i.e. we do not predetermine what subspace of the landscape an entity will reside in. Thus within the fixed constraints that the fitness function defines, niches are created and destroyed dynamically, and the shape and therefore properties of the niche for a given organism are not predefined.

4.7 Summary

Our motivation in this chapter has been to identify the properties of a system that make it hard for accretive mechanisms and yet amenable to compositional mechanisms. To this end we identified a class of system that has strong and difficult epistasis between variables, but also a highly modular structure that makes it amenable to decomposition.

First we examined characteristics that are difficult for accretion in pairwise interactions (systems of two variables). We defined the difficult type of epistasis as *interdependency*.

[31] The notion of removing inferior configurations from further consideration is allied to the notion of focussing on 'non-inferior' configurations (Van Hoyweghen et al. 2001a). But actually, as we see here, it is not the fact that (01) and (10) are not one of the equally good configurations that make them undesirable, it is the fact that *there is no context* in which either (01) or (10) are the best configuration that makes them undesirable. Indeed, it is not necessary that (00) and (11) be assigned equal fitness contributions - see 'Biased HIFF' (Watson & Pollack 1999a).

We distinguished the term *separable* from *decomposable*, and importantly, showed that the latter does not imply the former. Accordingly we are able to give an example system of 4 variables that is decomposable in to two modules that are strongly interdependent and therefore not separable.

We then scale up this structure, in a hierarchically consistent fashion, to arbitrarily large systems of variables that form hierarchical modular interdependency.

We defined one form of such a system several different ways so that we could see what was important about it from an adaptational point of view, what features created difficulty for accretion, and its relationship to concepts like nearly-decomposable systems of Simon, like aggregate effects. We discussed the features that can be relaxed/generalised.

Finally, we defined a simple example of the class, Hierarchical-if-and-only-if, HIFF, that will suffice for illustrating the class in the coming simulations. Moreover, this form is hierarchically consistent, the nature of the problem of finding solutions to modules at a given level, given solutions to modules at the previous level, is the same at all levels. This property of scale-invariance will be useful in simplifying formal analyses.

Chapter 5 - Mutation on Modular Interdependency

In this chapter we will briefly overview some properties of systems with hierarchical modular interdependency, in particular, systems of the specific form we will use for the main experiments of subsequent chapters, namely, HIFF. In particular we will look at characteristics of fitness landscapes that are generally deemed problematic for (accretive) evolution: the number of local optima, the separation of local optima or ‘width’ of fitness saddles, and the irreducibility of high-fitness configurations. We then show results for simulations of mutation on HIFF.

5.1 Cross-sections through the fitness landscape

In all of the following base fitness functions (as used above) the maximum fitness configurations for the system, and for every module it contains, are those where all states are equal:

$\boldsymbol{f1}(p,q) =$ (p IFF q) - as used in HIFF

$\boldsymbol{f2}(p,q) = pq+(1-p)(1-q)$ - continuous aggregation version of HIFF

$\boldsymbol{f3}(p_1,\ldots p_k) =$ 1 if ($\exists s \forall i$: $p_i=s$), and 0 otherwise - as used in H-Equal

i.e. for binary variables, the configurations with all 0s and all 1s give maximal fitness. The next best fitness configurations are those where all the biggest modules of the system are ‘correct’ (i.e. either all 0s and all 1s) but where one of them does not agree with the others. For systems where the span, the number of sub-modules per module, is 2 these second-best configurations are half 0s followed by half 1s, or conversely, half 1s followed by half 0s. Similarly, lower-fitness configurations occur when sub-sub-modules are correct but one of the sub-modules has sub-sub-modules of the incompatible type, and so on.

A system having a large number of variables creates a fitness landscape that is multidimensional—thus it is difficult to represent it pictorially. However, the systems that we have defined above are highly regular and symmetric so a single cross-section through the landscape, interpreted with some care, can actually give us a reasonable intuition about its properties in some respects.

The particular section that we find most informative is one that focuses on the high-fitness configurations. So, given our observations above, an appropriate section runs through the following points:

(000…000)
(100…000)
(110…000)
(111…000)
.
.
.
(111…100)
(111…110)
(111…111)

That is, the k^{th} point in the cross section, for $0 \leq k \leq N$, has k leading 1s and the remainder of states are 0s.

i.e. the kth point in the cross section, for $0 \leq k \leq N$, is the configuration

$(s_1,s_2,\ldots,s_N)$ where $s_{(i<=k)}=1$ and $s_{(i>k)}=0$.

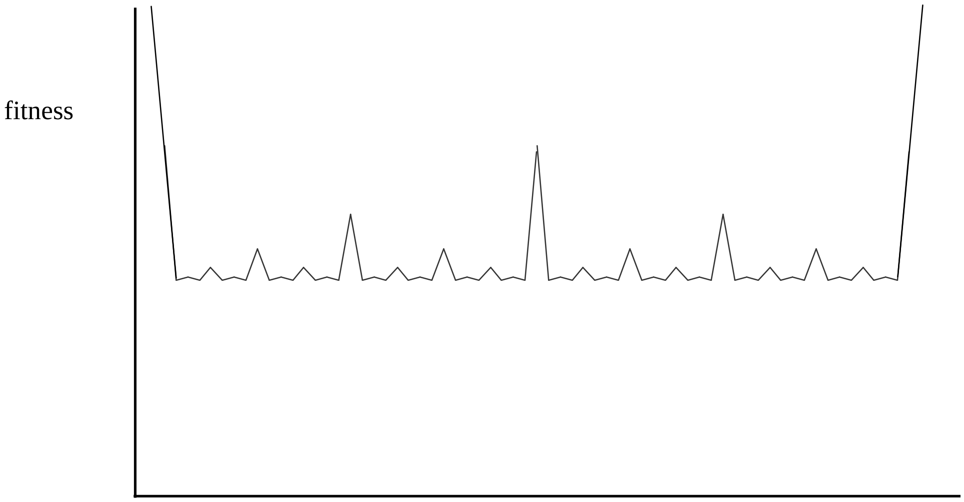
Consecutive points in this series are separated by Hamming distance 1, i.e. one point-mutations, and thus may be imagined as a particular one-point-mutation 'walk' across the fitness landscape from one global optima to the other. This particular series satisfies a mutation walk of the following form:

1. Start at one global optimum,
2. Of all one-point mutations that do not 'undo' a previous mutation of the walk, move to one (of those) that has highest fitness (even if its lower in fitness than the current point).
3. Go to 2.

Figure 5-1: 'Ridge mutation walker'

Thus there are exactly N+1 points in this walk. This walk crosses N+1 best-fitness points, although for many of the points there are many others with equal fitness: for example, f(000… 111)=f(111…000).

A section through a 64-variable system defined with HIFF (Equation 11) is given in Figure 5-2.



Consecutive points of ridge mutation walker

Figure 5-2: A particular cross-section through the HIFF fitness landscape.

The fractal nature of the fitness landscape, resulting from the recursive definition of modules and sub-modules, is clearly visible in this section.

One very useful characteristic of this cross-section is that it appropriately portrays the Hamming distance from each point to the nearest points with equal or higher fitness. That is the distance of one peak to the nearest equal or higher peak corresponds accurately with the values we calculated in the “H” column of Table 4-3, for example. Thus this section gives us a clear intuition about the width of fitness saddles in the landscape and some ideas about its ruggedness and the number of local optima. In particular, as is clear from the construction of the functions, at each hierarchical level of modules the hamming distance to the next best configuration doubles. i.e. the number of state variables that must flip-state in one go is equal to the size of the module which doubles at each hierarchical level. This is exactly the intent of the functions—to describe large systems of variables that are difficult for accretive mechanisms accumulating small changes.

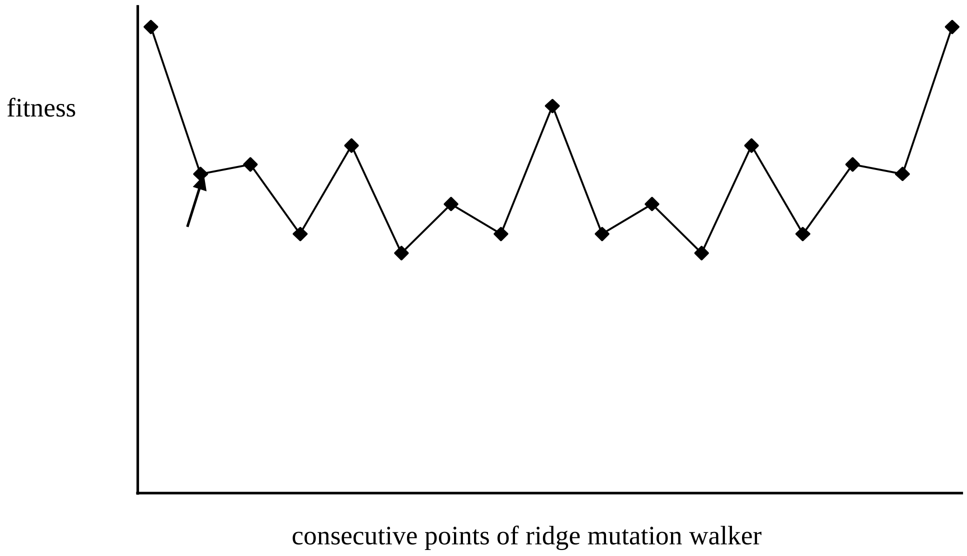


Figure 5-3: A section through the hierarchical modular interdependency system built on f2.

This section passes through the same point as those shown in Figure 5-2 (but with one less hierarchical level). See Equation 7. Note that the points like the one indicated with the arrow above are local optima with the weight values we defined in Equation 7, but with different weight scaling, if intra-module dependencies are not significantly stronger than inter-module dependencies, then this point will not be a local optima and the basin of attraction for the global optimum will grow larger. In the discrete HIFF this is not an issue, since it is built explicitly from high-order fitness contributions rather than a matrix of pairwise fitness interactions of different weights. Presumably, in a system of random weighted interdependencies like NK-landscapes, sometimes the cumulative effect of a number of interactions produces a local optima and sometimes it does not. In HIFF we are able to control these interactions explicitly and focus our model on the difficult interdependencies of epistatic effects.

For interest, Figure 5-3 shows a section through the continuous version of HIFF as defined in Equation 7. This landscape, has the same number of local optima in HIFF at the same configurations as those in HIFF, and therefore the separation of local optima is also the same as HIFF. (However, the fitness values of different configurations are different). An intuition for this landscape, and HIFF, is provided by considering the superposition of their base fitness functions at different scales. That is, the section through HIFF can be built from the sum of many appropriately scaled copies of the curve in Figure 5-4 (left), and the continuous version can be built from the sum of many appropriately scaled copies of the curve in Figure 5-4 (right).

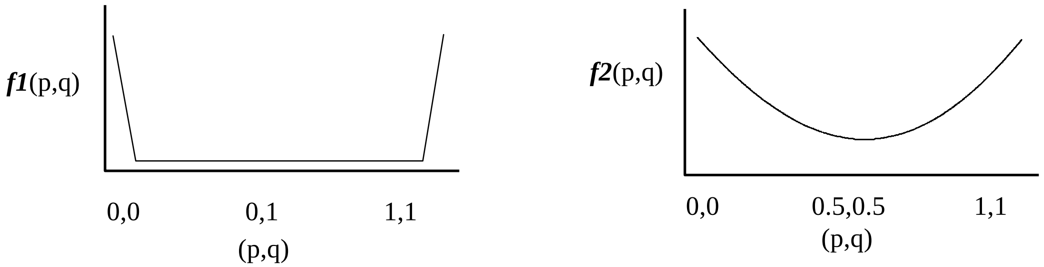


Figure 5-4: Base fitness functions

Cross sections through the base fitness functions of HIFF (left) and continuous HIFF (right). HIFF is based on ***f1***(p,q) = (p IFF q), and the continuous version is based on ***f2***(p,q) = pq+(1-p)(1-q), (see also Figure 4-2) Following the convention we have been using in figures such as Figure 5-2 and Figure 5-3, these sections show the two global optima at 00 and 11 at opposite extremes of the curve, and the intermediate points reachable by a walk of small changes in between.

The cross-sections in Figure 5-2 and Figure 5-3 are useful in gaining an intuition for why these landscapes are difficult for accretive mechanisms.

5.2 Difficulty of modular interdependency for accretive mechanisms

The characteristics of difficulty, common in our intuitions of evolutionary difficulty (2.1.2), that we will discuss are:

Ruggedness—the number of local optima.

Width of fitness saddles—the Hamming distance between a configuration and the nearest configuration with equal or higher fitness.

Irreducibility—whether the system ‘ceases to function’ under any small change.

These characteristics together tell us about whether there exists a path of monotonically increasing fitness from all points to the global optima, and more generally, the likelihood of finding such a path between a random configuration and a high-fitness configuration.

5.2.1 Ruggedness

The number of local optima in a landscape is intimately related to the number of variables that an algorithm can change ‘in one go’, as it were. To a single-bit mutation hill-climber for example, HIFF will have more optima than for a two-bit mutation hill-climber. In HIFF, a single-bit mutation hill-climber (or any algorithm that can change at most one variable at a time) has $2^{N/2}$ local optima (where *N* is the size of the problem in bits), only two of which are globally optimal.

More generally, any string made by concatenating *h*-sized correct blocks requires a change of at least h bits to find a fitter configuration. Such a string therefore cannot be improved by a *k*-bit mutation algorithm where *h*>*k*. There are $2^{(N/h)}$ possible strings of such concatenations. Thus a *k*-bit mutation hill-climber will be faced with $2^{(N/h)}$ local optima, where *h* is the smallest integer power of 2 which is greater than *k*, i.e. $h=2^p>k$.

In short, HIFF has a number of local optima that is exponential in the size of the problem for any given mutation radius.

5.2.2 Width of fitness saddles

Since HIFF has many local optima it also has many fitness saddles (by definition), but to properly understand the difficulty of the problem we also need to know the width of these fitness saddles. From our examination of local optima above we see that HIFF has $2^{N/2}$ saddles of width-2, and $2^{N/k}$ fitness saddles of width k. That is, there are $2^{N/k}$ configurations of the system for which the nearest configuration with equal or higher fitness differs in the setting of k variables. Put another way, there are $2^{N/k}$ configurations of the system for which all other configurations with equal or higher fitness differs in the setting of at least k variables.

Another characteristic we may be interested in is the size of the ‘basin’ for an optima. For example, for a k-bit mutation hill-climber, how many different (starting) configurations are there from which this hill-climber will be able to reach one of the global optima? For a 1-bit hill-climber, the proportion of the space from which one of the global optima can be reached is $P_1=2N/2^N$ (there are N configurations 1-bit away from each global optimum, 2^N points in total in the space, and 2 global optima). In general, for increasing mutation rates, as the width of saddle that can be jumped increases the size of the basin of the optima increases exponentially. However, note that for mutation, the expected time to make a specific jump of a given size also increases exponentially with the size of the jump.

These restrictions apply not just to mutation but to any method of undirected, random change (see 2.5.1). For example, genetic drift (Wright 1977), or neutral drift introduced by the addition of arbitrary neutral networks in the landscape. See (Shackleton et al. 2000), (Shipman et al. 2000), and (Ebner et al. 2001), for experiments on the addition of neutral mappings to functions including HIFF, and see (Knowles & Watson 2002), for experiments showing that algorithmically this performance is no better than increased mutation. See also (Knowles et al. 2001) for experiments using simulated annealing on HIFF. In previous work (Watson et al. 1998), we also showed that HIFF is not amenable to macro-mutation and passes the 'Headless chicken test' (Jones 1995) i.e. using crossover but crossing one parent with a random string does not solve large HIFF problems efficiently.

Of course, a method that has some *a priori* bias, for example, one that tends to explore the complement of fit strings, may perform very well on HIFF (in fact, this particular bias is useful in later discussion 6.6.3). But, any method of increasing the exploration ability of an algorithm *arbitrarily* does not help in solving HIFF - i.e. any method using single inheritance plus random variation is incapable of exploiting the decomposability of the landscape.

5.2.3 Irreducibility - HIFF *appears* to be irreducibly complex (but is not)

According to Behe, a system is irreducibly complex if any small change in the configuration of the system causes it to 'cease functioning'. In our model problem, each module is 'correct' only when all of the variables it contains are in mutual agreement (all 0s or all 1s). If we take a configuration at one of the global optima in HIFF and change one the state of one variable, then we 'break' the fitness contribution conferred by the agreement of the two largest modules, and the fitness contribution conferred by the agreement of the two largest modules within the half we changed, and within the quarter, and so on. In other words, we lose the fitness contribution from one module of each size at each level in the hierarchy. Arguably, this reasonably constitutes a 'ceasing to function' of the system as a whole. Certainly, the system does not 'degrade gracefully' under small perturbations.

Quantifiably, this should be related to a decrease in fitness. The exact decrease in fitness will depend on the scaling of the weighted contributions from one level of module to another. In HIFF, each correct module has a fitness contribution equal to its size (the number of variables it contains) so a one-bit mutation from a global optimum results in a decrease in fitness of N+N/2+N/4+N/8+…+4+2=2N-2. Let us look at this decrease in fitness in comparison to the range of fitnesses in HIFF, and then compare the change in fitness to the size of the genetic change that produced it.

The minimum fitness value in HIFF is: N - e.g. coming from the string 010101…

The maximum fitness value in HIFF is: $N(\log_2 N+1)$ - e.g. the fitness of the string 00000… which has all modules at all levels correct is $N+2(N/2)+4(N/4)+\ldots+N(N/N)= N(\log_2 N+1)$.

Thus the range of fitness values in HIFF is: $N(\log_2 N+1)-N= N\log_2 N$.

The ratio of the change in fitness from any (or the best) single-point mutation from a global optimum in HIFF is therefore: $(2N-2)/ N\log_2 N \approx 2/\log_2 N$.[32] We will call this the ‘relative change in fitness’.

The ratio of the change in genotype from a single-point mutation to the size of the genotype is: 1/N. We will call this the ‘relative change in genotype’.

So, the relative change in fitness caused by a one-point mutation from a global optimum (or any fully-correct module), $2/\log_2 N$, is very much larger than the relative change in genotype, 1/N, for large N.

So, a small change from a high fitness point in HIFF ‘breaks’ at least one module of all sizes, and by the above criteria, is significantly deleterious. Thus we might conclude that by Behe’s criteria, a high-fitness point in HIFF is irreducibly complex. However, as we indicated in 2.4.1, a system that ceases to function under any small change is not necessarily unevolvable. And under a more general notion of reducibility, HIFF is reducible - i.e. although it is not into small parts, it is reducible into large parts. That is, there is a large change we can make to the system that does not break a module at every level and causes only half the fitness decrease. Specifically, substituting one N/2 sized block for another might break the top-level fitness contributions of the system, but it is still composed of two independently viable halves. More to the

[32] This is on the order of 1/H, where H is the number of hierarchical levels in the system.

point, since a correct and compatible N/2 sized module can be evolved in some other individual and swapped-in by compositional mechanisms, HIFF as a whole will be easily evolvable for compositional mechanisms.

5.2.4 HIFF *appears* to have low epistasis (but does not)

Although epistatic dependencies are observed in some biological systems, it is also observed that most of the fitness effects from allelic changes in biological systems can be accounted for by their independent, non-epistatic, components. Thus, it might seem that the emphasis on strong epistatic dependencies shown by the HIFF landscape is biologically unrealistic but, in fact, HIFF is compatible with these observations. When a set of genes is well optimised in HIFF, any single allelic change is catastrophically deleterious, as we analysed above. Moreover, no pair-wise interactions, no second single-point mutation, can recover any significant positive benefit relative to the deleterious effect of the first mutation. So, it might appear from these observations that HIFF has weak epistatic interactions because most of the change in fitness can be accounted for by the independent fitness effects of alleles.

However, this is not the case. In fact, the independent effects of genes account for none of the fitness changes—when averaged over *all* possible genetic contexts, every allele (at every locus) is equally valuable in HIFF (0s confer the same fitness contributions as 1s). The apparent weakness of epistatic dependencies results from measuring only pair-wise (or small group) interactions on a configuration of genes that have been optimised to a high level. At this level, pair-wise interactions are not sufficient to exhibit large fitness effects—but such epistatic effects are available through the interaction of large modules.

So, a well-optimised configuration of features in HIFF appears to be *irreducibly complex* because any small genetic change is catastrophic, causing the system to ‘fall-off’ its needle in the fitness landscape. However, although the adaptive problem cannot be reduced into the additive effects of individual genes, it can be decomposed into the large modular sub-systems of which it is composed.

5.3 Expected time to solution for accretive mechanisms

The previous section indicates that HIFF has all the properties we listed that are usually associated with evolutionary difficulty under accretive assumptions. In this section, we address the expected time to solution for accretive mechanisms.

Using the notion of a path to solution and the expected time for steps on this path, we may reasonably say that an algorithm is *not* reliably successful on a problem if any of the following conditions are true:

- there is no guaranteed path to solution,
- if the path may be exponentially long, or
- if any step on the path takes exponential time.

For a single-bit mutation hill-climber there is no guaranteed path on H-IFF. H-IFF has $2^{N/2}$ local optima under single-bit mutation (where *N* is the size of the problem in bits), only two of which are globally optimal. More generally, a *k*-bit mutation hill-climber will be faced with $2^{(N/h)}$ local optima, where *h* is the smallest integer power of 2 which is greater than *k*, i.e. $h=2^p>k$. A random mutation hill climber (RMHC) (Forrest & Mitchell 1993a) mutates every bit with a given probability *Pmut*. In principle no problem can have local optima under this operator since the probability of moving from any point to any other point is non-zero. However, consider the case where the current string is *N*/2 zeros followed by *N*/2 ones. The next best string is the global optima at all ones or all zeros. To achieve this jump, mutation must flip *N*/2 bits whilst keeping *N*/2 undisrupted. The best mutation rate to achieve this is *Pmut*=0.5 and this gives an expected time for the step which is 2^N – so search with RMHC is no better than random guessing at this mutation rate. A macro-mutation hill-climber, MMHC (Jones 1995), has the best chance of success. MMHC chooses two delimiting loci and randomises the loci between them, thus concentrating mutations on a particular sub-string whilst leaving the remainder untouched.[33] But still, to escape from the next-best optima to either global optimum it must choose the right delimiting loci which has probability 1/*N(N-1)* (this allows macro-mutations both 'inside' and 'outside' the chosen points), and assign all ones (or all zeros)

[33] This algorithm has an advantage over other mechanisms because it explicitly uses the assumption of tight genetic linkage (3.4.2).

to $N/2$ bits – this occurs in expected time $\boldsymbol{O}(N^2 2^{(N/2)})$. Thus we see that by these criteria, all these mutation-based hill-climbers either have no guaranteed path to the optimum, or cannot be guaranteed to take a step on the path in time less than exponential in N.

5.3.1 Preliminary Comparison with compositional mechanisms

We will not attempt a detailed analysis of the expected time to solution for a compositional mechanism until after we have detailed a specific compositional mechanism. But for now, it is worth emphasising that although HIFF is very difficult for accretive mechanisms, it is decomposable into small number of more manageable problems and is thereby easily solvable *if* this decomposition is known or can be discovered, and *if* the search mechanism can manipulate modules as wholes.

Specifically, in HIFF, for any module A, M_A (the maximal configurations of A - see 4.3.1) will only contain configurations that are some combination of a configuration from M_B and a configuration from $M_{C,}$ the maximal configurations for its sub-modules, B and C. And thus finding M_A, only requires searching at most $|M_B|$ x $|M_C|$ configurations if M_B and M_C are known and we have a mechanism that can 'swap in and out' different members of M_B and M_C as whole units. In HIFF, $|M|=2$ for any module at any level in the hierarchy, so finding M_A given M_B and M_C only requires testing 4 configurations regardless of the size of the module. At the h^{th} level in the hierarchy (primitive variables level is indexed 0), there are $2^{(H-h)}$ modules where $H=\log_2 N$ and $0<h<H$. The total number of modules to be found over all levels is thus 2(N-1), and each of these can be found in 4 combinations of the modules from the previous level. The total number of combinations of sub modules that need to be searched to find the solutions to all modules in the problem is thus 8(N-1), and is therefore the maximum number of configurations that need to tested to find a global optima.

In summary, if the decomposition of the problem is known and we have a mechanism that can identify the set M for a module and manipulate modules as units, then the number of configurations that needs to be searched is less than 8N. In contrast, as we have seen above, accretive mechanisms require the testing of a number of configurations exponential in N.[34]

This illustrates the broad margin of difference that might be possible between accretive and compositional mechanisms. But in order to vindicate the adaptive capacity of compositional mechanisms we will need to show that a compositional mechanism can discover the problem structure and manipulate modules effectively.

5.4 Simulation results for mutation

Random Mutation Hill-Climbing, (RMHC), repeatedly applies mutation to the features of a single binary string (a fully specified feature set) and accepts a variant if it is fitter (Forrest & Mitchell 1993a). We conducted experiments with various mutation rates (probability of assigning a new random state {0,1} to each feature)—specifically, ***mut*** = 1/128, 2/128, 4/128, 6/128, 8/128, 12/128, 16/128, 24/128, 32/128 and 40/128. In the summary figure later we show the performance of RMHC with ***mut***=16/128=0.125 which gives the best maximum average maximum fitness over all these values. (See Oates & Corne 2001, for an investigation of the mutation landscape for HIFF, and the behaviour of the GA with small populations).

[34] Even if an accretive mechanism 'knew' the decomposition of the system and could 'focus' mutations on modules that were sub-optimal, an accretive mechanism would still require $2^{N/2}$ guesses to find the solution to the largest sub-modules, since progress on sub-sub-modules cannot be used to inform mutational changes.

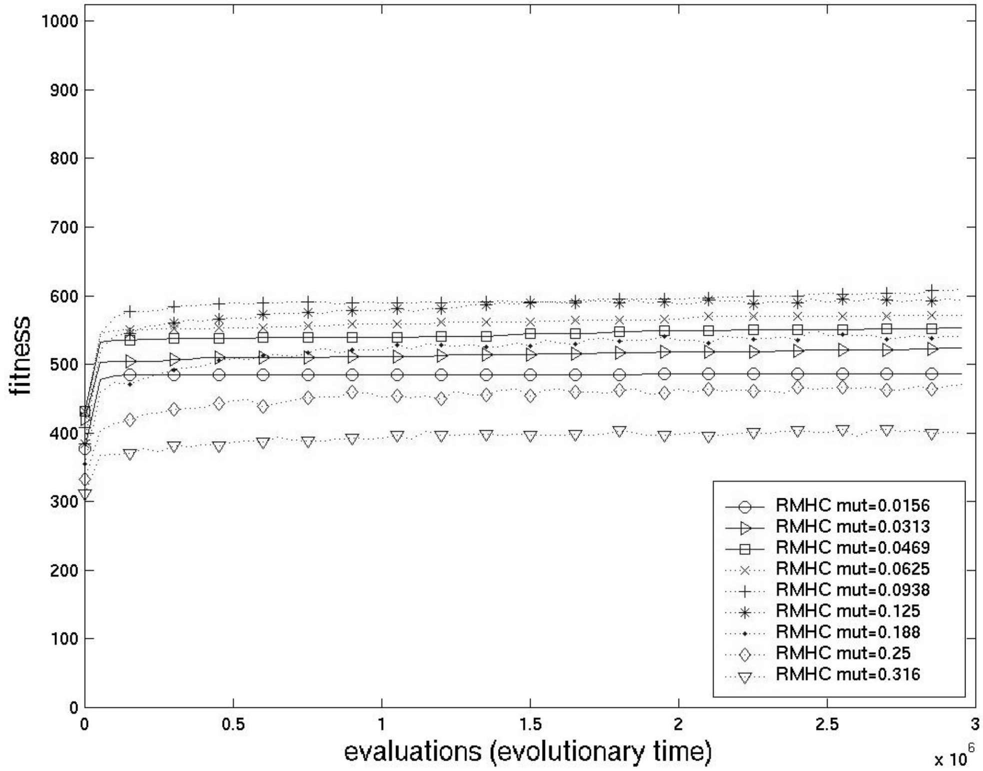


Figure 5-5: Results of RMHC on a 128-variable decomposable system.

HIFF (Equation 11) defines the fitness landscape. Curves show the average fitness of all individuals measured in the last 2000 evaluations (of newly created individuals). Several curves are shown for different mutation rates. The maximum fitness value possible in this fitness landscape is 1024. The maximum number of evaluations permitted (duration) is $3 \cdot 10^6$.

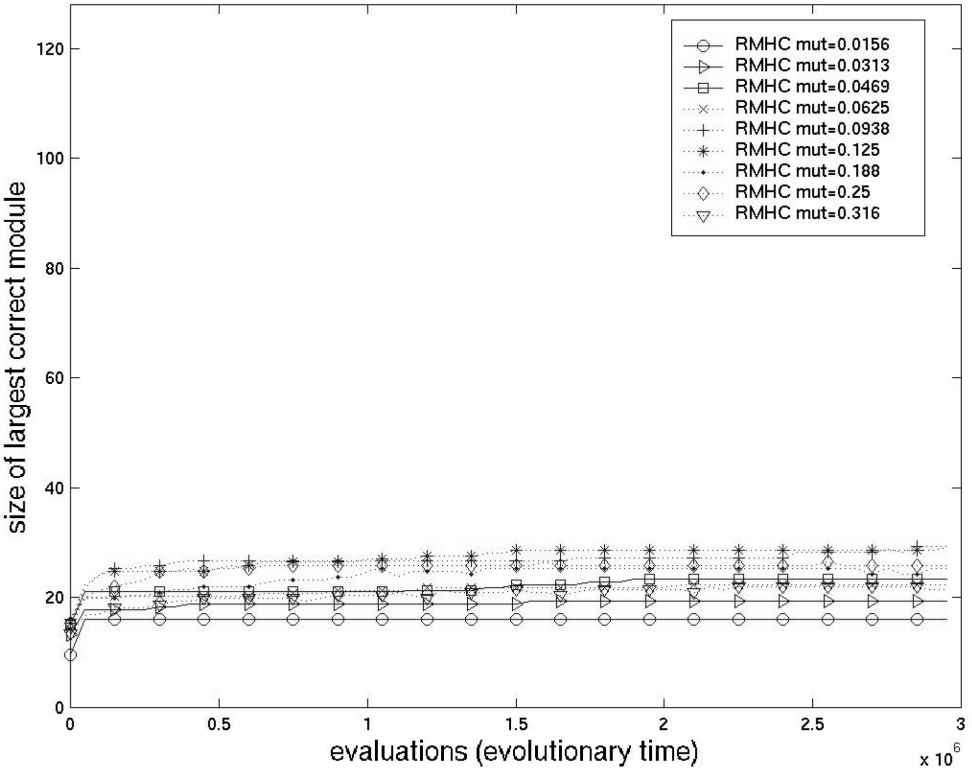


Figure 5-6: Results of RMHC on a 128-variable decomposable system.

Details as per Figure 5-5 but curves show size of largest module fully-optimised in the past 2000 evaluations.

5.5 Summary

In this chapter we assessed the difficulty of HIFF for accretive mechanisms such as mutation. We examined the properties of the landscape that are normally considered difficult for (accretive) evolution. We find that it has the following properties:

- The number of local optima (for mutation based algorithms) is exponential in N (N = # variables in system).

- The width of fitness saddles is a function of N, i.e. fitness saddles of all sizes up to N/2 exist in the landscape. Therefore, no constant mutation radius is sufficient to solve HIFF in general.

- High-fitness configurations of the system appear to be irreducibly complex - i.e. any small change is catastrophically deleterious.

- There is no path of small changes conferring monotonically increasing fitness that approaches the optima in the landscape.

Accordingly, HIFF satisfies all these criteria of evolutionary difficulty.

A simple analytic examination shows that a mutation based algorithm cannot be guaranteed to succeed in time less than exponential in N. This also holds for any algorithm using undirected exploration such as genetic drift (from stochastic sampling error), or neutral walks in redundant encodings. In other words, algorithms of single inheritance and arbitrary exploration cannot, by reasonable definitions, solve HIFF in less than exponential time.

Simulations of a simple mutational algorithm, RMHC, illustrate the performance of mutational mechanisms on HIFF. In the next 2 chapters we will investigate the operation of sexual recombination and symbiotic encapsulation on HIFF.

Chapter 6 - Sexual Recombination on Modular Interdependency

In this chapter we examine the adaptive capabilities of an evolving population with sexual recombination on the class of modular interdependency systems defined previously. In particular, we will use the Hierarchical-if-and-only-if function, Equation 11, to define the epistatic dependencies of genes and the corresponding adaptive landscape. From the prior analysis we know that this problem is difficult for mutation but we also know that it is, in principle, amenable to hierarchical decomposition. The question of this chapter then, is whether there are circumstances where an evolving population using variation by sexual recombination is able to exploit this decomposable structure by discovering and manipulating modules and sub-modules appropriately to assemble together a high-fitness configuration for the system.

We will use a simple computational simulation, the Genetic Algorithm, GA, to investigate this question. The GA models a population of individuals. Each individual is represented by a single haploid chromosome[35] and this chromosome is simply a binary string of ‘genes’ - i.e. there are two alleles for each locus. Each gene represents the state of one variable in the HIFF problem. Thus each individual represents a complete state specification for the whole interdependent system of variables, or in engineering terms a ‘candidate solution’ for the ‘problem’. The fitness of each individual is given by the HIFF function.

Our intuition, following that of notable GA pioneers and practitioners (see Building Block Hypothesis, 3.3.2) is that different members of the population will discover different modules, and that recombination between individuals will be able to search combinations of modules to find larger modules, and so on. We

[35] A haploid representation is chosen for simplicity. Equivalently, we may imagine a diploid chromosome where fitness is, for some reason, a function of only one haploid part, e.g. we may imagine that selection is applied only on the haploid part of the lifecycle. It is also perhaps equivalent to suppose that the same model for multi-loci epistatic interactions also models single-locus dominance interactions. That is, every single locus in the diploid representation corresponds to a pair of variables in the first level of the dependency hierarchy of the system. Under HIFF, the heterozygote loci would then have no fitness dependencies at the next level in the hierarchy and only the homozygote loci (00,11) would be important - effectively returning us to a haploid representation.

will find that with the use of HIFF we can clearly illustrate when this process is possible and when it will fail. In particular, we will find that appropriate diversity and appropriate gene ordering and strong genetic linkage are required for effective recombination.

We will explore some different varieties of the GA that vary in the respects of the selection scheme and the variation operators. These different selection schemes will affect the diversity of the population: the basic form models a single panmictic (freely mixed) population; the more sophisticated form models a particular kind of population sub-division based on removing competition between dissimilar types. The variation mechanism of interest in this chapter is sexual recombination. Different assumptions about the recombination mechanism have different effects on genetic linkage. In particular, if we assume that linkage is weak (crossover points are numerous) then subsets of genes cannot be exchanged between individuals as wholes, i.e. no particular subsets of genes from a parent are more likely to appear in the offspring than any other subset. And even if linkage is strong, the appropriateness of exchanging gene subsets between individual depends on the correspondence between gene ordering (proximity on the chromosome) and epistatic dependency. In other words, weak linkage and/or poorly organised linkage are problematic for crossover.

We find that a population with appropriate mechanisms to support diversity, and appropriate assumptions about genetic linkage, is able to solve HIFF easily. This demonstrates that under these circumstances, an evolving population with sexual recombination can compose together modules to find successively larger subsystems of genes, and eventually find the globally optimal configurations for the genes, in this case. It is thus our first example of a compositional mechanism. We provide some analysis, based on simplified conditions, that shows an expected time to solution that is polynomial in the size of the system.

However, these experiments equally show that there are conditions under which sexual recombination performs no better than mutation on this problem class. We can see that a single freely mixed population, or a recombination mechanism with weak linkage, or a linkage arrangement that does not correspond with epistatic dependencies, will each cause the compositional process to fail. In these cases the adaptation of individuals in the population is accretive—at best, only able to discover more fit configurations when they are available via small incremental changes. Accordingly, we see that the effect of sexual recombination can be either accretive or compositional depending on circumstances and assumptions.

6.1 Overview of models

In the following subsections we will briefly describe the simulation models and the corresponding experimental results for different assumptions.

- **Basic model:** single freely-mixed population, one-point crossover: GA with fitness proportionate selection.
- **Subdivided population model**: GA with one-point crossover and deterministic crowding.
- **Genetic linkage models:**
    - No genetic linkage: uniform crossover
    - Unfavourable genetic linkage: HIFF with randomly reordered gene positions - ‘Shuffled HIFF’.

This chapter uses the GA and variants of the GA throughout because the issues of module discovery and manipulation under crossover are easily demonstrated with a simple linear bit-string representation and bit-string recombination. However, the issues of module discovery and manipulation under compositional mechanisms are relevant in other types of EAs also - the GA merely provides a simple illustration. The discussion on genetic linkage however is specific to the linear representation of chromosomes used in GAs - in Chapter 7 we move outside the normal GA framework into a novel form of EA that is not dependent on the linear order of genes, and could be applied in other substrates.

6.2 Basic Model: Single panmictic population—Simple GA

In this section we investigate the adaptive capabilities of a population with sexual recombination using a simple model. Our basic model will be the Simple Genetic Algorithm (3.1.2). The SGA assumes a single freely-mixed or unstructured (panmictic) population of individuals. In the following experiments we will use a ‘generational’ algorithm meaning that the reproductive cycles of the individuals in the population are synchronised (Syswerda 1990). The algorithm is described in Figure 6-1. This model uses standard fitness proportionate selection.

```
1) Initialise population with random individuals, i.e. random bit strings.
2) Repeat until stopping criterion (duration complete[†]).
    i) Evaluate the fitness of all individuals in population (using HIFF).
    ii) Repeat <popsize> times
        (1) With selection probability of each individual proportionate to their fitness, select
            (with replacement) two individuals from population to be ‘parents’.
        (2) Generate an ‘offspring’ individual by:
            (a) Create a new string by one-point crossover of parent strings (see 3.3.1)
            (b) Apply point-mutation with low per-locus probability.
        (3) Collect offspring in new_population.
    iii) Replace population with new_population
```

Figure 6-1: A procedural outline for a simple ‘panmictic’ GA.

This GA has no diversity maintenance and the population is unstructured or ‘panmictic’ - as is normal for a simple generational GA with fitness proportionate selection. [†]The duration of the experiment will be measured in evaluations of individuals (corresponding to lifetimes).[36]

6.2.1 Simulation results

Figure 6-2 shows the simulation results for this model.

[36] This method, rather than counting generations, helps to avoid unfair comparisons of adaptive operators arising from different population sizes. For example, in principle, any variation operator (or no variation operator) can find the globally optimal configuration for the fitness landscape in one generation if the initial population is large enough.

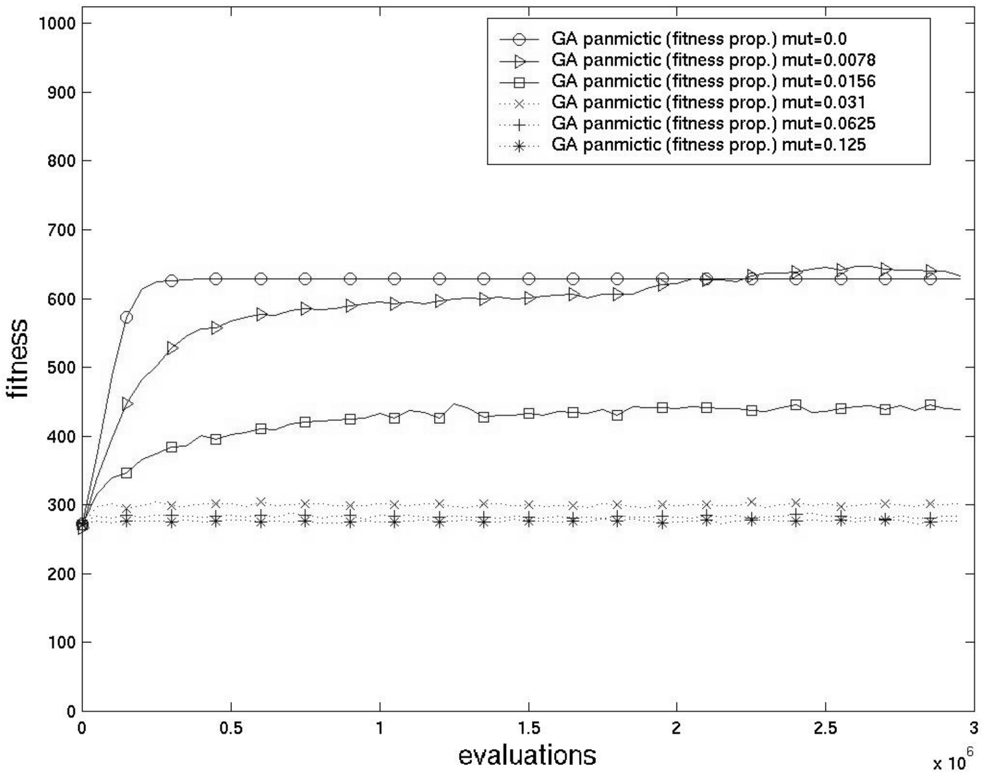


Figure 6-2: Results of simple panmictic GA with one-point crossover on a 128-variable HIFF.

HIFF (Equation 11) defines the fitness landscape. The population size is 2000 individuals.[37] Results measure the average fitness of all individuals measured in the last 2000 evaluations (new individuals). Several curves are shown for different mutation rates. Each curve is the average of 30 independent runs. The maximum fitness value possible in this fitness landscape is 1024. The maximum number of evaluations permitted (duration) is $3 \cdot 10^6$.

We see that the population is not able to reach very high fitness values in this landscape Figure 6-2. Figure 6-3 shows that the maximum-size sub-module discovered in these runs is around 32 bits on average, only a little better than the results shown for a mutation hill-climber.

[37] Preliminary investigations indicated that the reliability of the results in Section 6.3 were improved with a large population size such as that used here. We use the same population size in all other experiments in this chapter to be sure that the result in Section 6.3 is not merely the effect of a large population size. Some indication of the effect of different population sizes is given in Section 6.7.2.

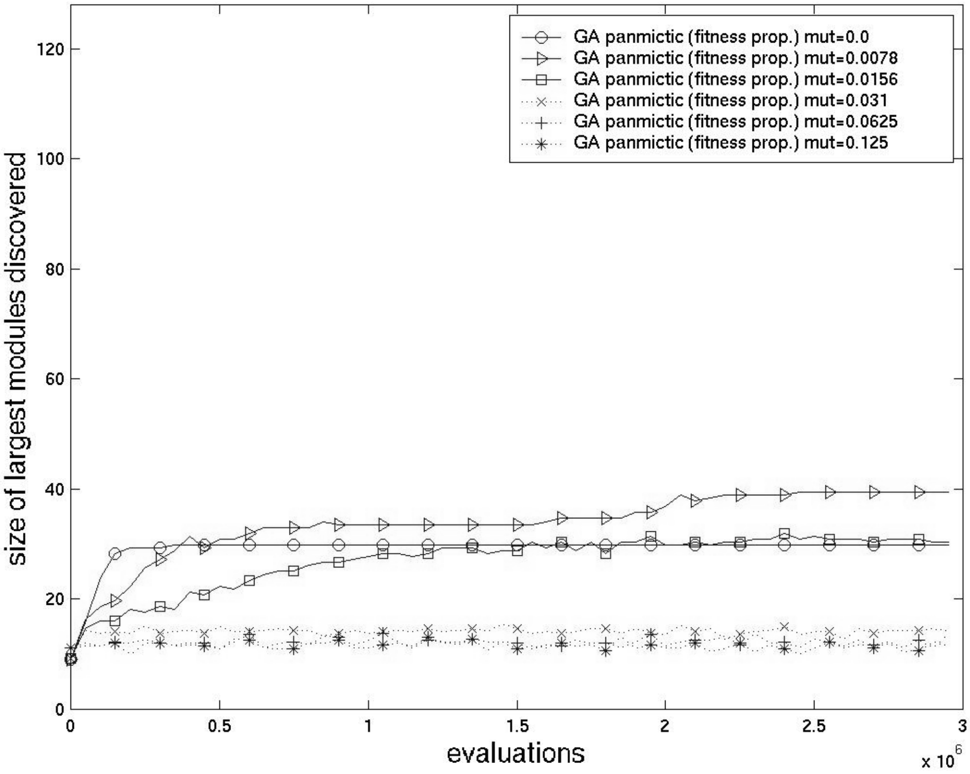


Figure 6-3: Results of simple GA with one-point crossover on a 128-variable HIFF.

Details as per Figure 6-2 but showing size of largest all-ones module in population (rather than average fitness of population).

6.2.2 Discussion

The Simple GA with crossover did not do much better than the mutation hill climber of the previous chapter - crossover appears to make negligible difference in this problem. However, if we look at the evolved populations in detail we find that in this model the population very quickly becomes genetically converged (not shown). This is to be expected—when there is no pressure to do otherwise, the population will quickly converge to the best-fitness individual thus far discovered. If this happens before modules can be recombined between individuals then the string the population converges to will not be very fit (see “mixing” and trade-off with selection in Thierens & Goldberg 1993). To the extent that the population has converged, crossover between individuals results in little change between parents and offspring. Thus the only new variation in the population is supplied by mutation, and accordingly the results are not much better than previous mutation-only results.

6.3 Diversity: Subdivided/niched population—GA with crowding

Following from the previous experiment we modify the selection scheme to model a process that may better preserve the initial diversity of the population. Specifically, we will use a GA with a diversity maintenance technique known as ‘Deterministic Crowding’ (Mahfoud 1995). Crowding methods in general can be understood as a mechanism to focus competition in such a way that similar things compete but dissimilar things do not. In deterministic crowding, DC, this is achieved in two ways: First, individuals only compete with their own immediate ancestors (their parents); Second, individuals of a ‘brood’ (i.e. with same parents) compete with the parent to whom they are most closely related. Pseudocode for a simple GA based on Deterministic Crowding is given in Figure 6-4.

```
1) Initialise population with random individuals, i.e. random bit strings.
2) Evaluate the fitness of all individuals in population (using HIFF).
3) Repeat until stopping criterion.
    i) Select two individuals from population at random to be ‘parents’, p1 & p2.
        (a) With probability 0.7, generate a complementary pair[†] of ‘offspring’ individuals, c1 & c2 by one-point crossover of parent strings (see 3.3.1). Otherwise offspring equal parents.
        (b) Apply point-mutation with low per-locus probability to offspring.
        (c) Evaluate new individuals (using HIFF).
    ii) Pair-up each parent with one offspring according to the pairing rule below.
    iii) For each parent/offspring pair, if the offspring is fitter than parent then replace the parent with the offspring (otherwise discard offspring).
```

Pairing rule: if **H**(*p*1,*c*1)+**H**(*p*2,*c*2) <**H**(*p*1,*c*2)+**H**(*p*2,*c*1) then pair *p*1 with *c*1, and *p*2 with *c*2, else pair *p*1 with *c*2, and *p*2 with *c*1, where **H** gives the genotypic Hamming distance between two individuals.

Figure 6-4: A procedural outline for a simple steady-state GA with Deterministic Crowding.

[†]A complementary pair of offspring,[38] c1, c2, is created from two parents, p1, p2, by using the same crossover point such that c1=p1Xp2 and c2=p2Xp1 where aXb is the result of crossover where genes are taken from parent-a on the left of the crossover point and from parent-b on the right of the crossover point (or vice versa). In other words, each gene of each parent will occur in either one offspring or the other.[39]

The important characteristic of this diversity maintenance method is that competition is restricted to similar individuals but breeding is not. This model maintains diversity in the following manner. First note that individuals are selected to be parents *at random* not according to their fitness. If fitness effects are

[38] The algorithm also performs well without the stipulation that both offspring are created with the same crossover point, but this feature makes analysis easier (see 6.6).

[39] This algorithm is wasteful of evaluations when the mutation probability is zero: specifically, the offspring created without crossover remain identical to parents, and cannot produce a change in the state of the population. However, previous work suggested that 70% crossover performs better than 100% for some mutation rates - so we used 70% crossover throughout.

introduced to select the parents then this applies a global (population-wide) pressure to promote the better individuals found so far regardless of their similarity/diversity. This would cause the population to converge on the best adaptive peak found thus far. Instead, selection is applied at replacement (3-iii) where it is restricted to parent/offspring pairs. This isolated competition allows different subdivisions of the population to coexist temporally on different adaptive peaks in the landscape even if they are quite different in fitness. This effects an efficient form of niching where sub-populations climb local peaks but individuals on different peaks do not compete. However, if crossover should result in an individual that is fit in both subdomains it can take over both peaks. For example, if the genes of p2 are carried by c1, and c1 is fitter than p1 but c2 is not fitter than p2, then the genes of p2 remain in p2 *and* replace those of p1. Thus if a collection of genes are fit in more than one context they can replicate and take over more than one peak. This allows an appropriate balance of diversity maintenance and competitive exclusion.

Deterministic Crowding was devised for pragmatic optimisation purposes, and at first glance seems quite ‘unnatural’. However, there are several ways to interpret this model. For example:

- **Deterministic crowding and population subdivision.** In sub-divided populations individuals compete strongly with other individuals in the same deme but not with individuals in different demes. Moreover, when migration between demes is low, individuals within a deme are generally genetically related. Accordingly, individuals compete most strongly with their own ancestors. When migration does occur, the resultant offspring also need to compete with the residents of the new niche.[40]

- **Deterministic crowding and niching.** The general notion of allowing similar things to compete and dissimilar things to coexist is quite a normal interpretation of a multi-species ecosystem. The DC method can be interpreted as a specific interpretation of this general model. In particular, if a given individual is assumed to compete most strongly with the other individual in the whole population to which they are most similar then this is likely to be one of their parents (see ‘preselection’ Cavicchio

[40] In this interpretation, instead of interpreting each DC-individual as a literal individual, we may alternatively interpret each DC-individual as a highly converged sub-population. In this case, the particular configuration of features in the DC-individual can represent the sub-population ‘consensus’ string. Then the reproduction of one DC-individual with another is a model of inter-deme migration and crossover.

> 1970, Mahfoud 1995 p. 129). In this respect, DC is merely an implementational approximation to a model where individuals compete only with that other individual with whom they are most similar.

So, with some open-mindedness, there are perhaps biologically plausible interpretations of this particular model. And more general biological models of population subdivision may have an equally advantageous affect on population diversity. However, the biological plausibility of deterministic crowding and the particulars of the deterministic crowding algorithm are not important for our purposes. We merely want to show that if appropriate diversity is maintained in the population, by whatever means, then crossover is able to assemble together modules effectively.

A completely different diversity maintenance method, based on competition over shared resources, is also able to maintain diversity appropriately (Watson et al. 1998) - but the resource model uses more domain knowledge than the DC method used here.[41] Previous work in other domains indicated that, under certain conditions, competitive resource consumption in a continuous space of resources can result in diversification and cooperative displacement into complementary roles (Ebner et al. 2000).[42] [43]

The results of the deterministic crowding model (Figure 6-4 above), are shown below in Figure 6-5/Figure 6-6.

[41] The resource-based fitness-sharing model uses a resource for each module in the problem and depresses the value of solving a module in proportion to the number of individuals that already solve that module. See (Horn et al. 1994) for an example of resource-based fitness-sharing where each resource corresponds to a dimension in a multi-dimensional optimisation problem.

[42] This model turns out to be very similar to the idea of "resource utilisation functions" in a continuous space of resources (MacArthur 1968). Whereas, our previous work on fitness sharing in HIFF is more closely allied to a 'niche as hypervolume' model (Hutchinson 1965).

[43] I am particularly grateful to Martin Oates for investigating many different 'restricted mating' types of diversity maintenance for GAs on HIFF, and to Christopher Ronnewinkel who brought Deterministic Crowding to our attention in this effort.

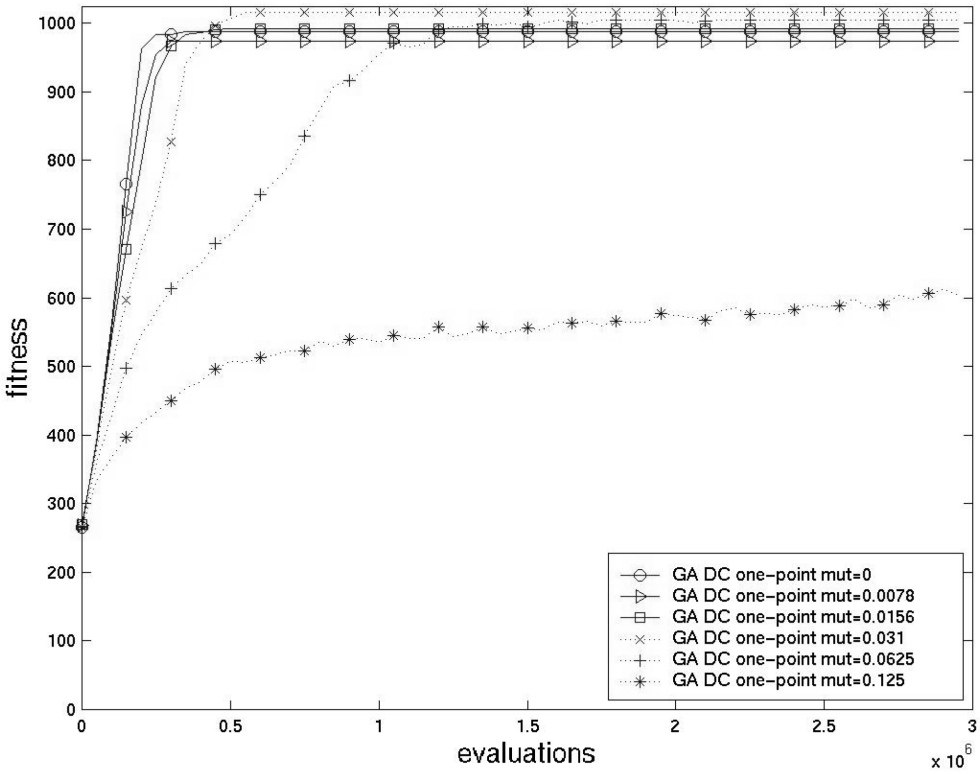


Figure 6-5: Results of deterministic crowding GA with one-point crossover on 128-variable HIFF.

HIFF (Equation 11) defines the fitness landscape. The population size is 2000 individuals. Results measure the average fitness of all individuals measured in the last 2000 evaluations (new individuals). Several curves are shown for different mutation rates. The maximum fitness value possible in this fitness landscape is 1024. The maximum number of evaluations permitted (duration) is $3 \cdot 10^6$.

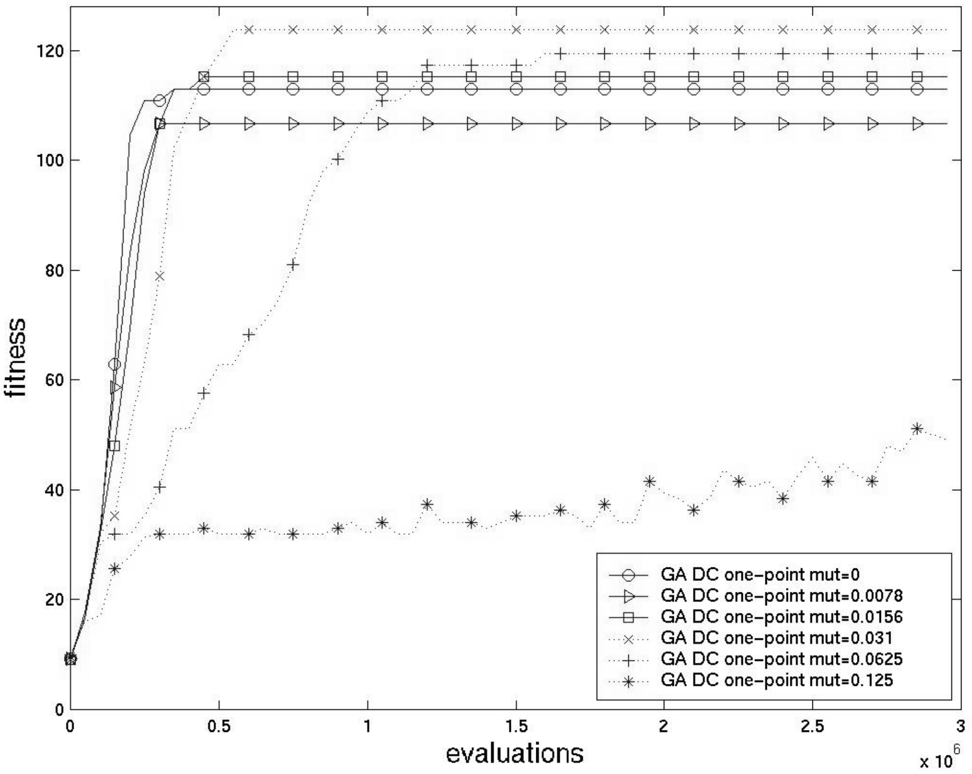


Figure 6-6: Results of deterministic crowding GA with one-point crossover on 128-variable HIFF.

Details as per Figure 6-5 but showing size of largest all-ones module in population (rather than average fitness of population).

We see that with the aid of the crowding mechanism the population is able to reach very high fitness values in this landscape, in all cases except when using very high mutation rates. In fact, with a mutation rate of 0.0310 (2/N) it finds *both* globally optimal configurations for the system, modules of 128-variables, in 28/30 runs, in about 500,000 evaluations.

The DC model is an ‘off-the-shelf’ diversity maintenance technique for GAs - so for the experiments on genetic linkage in the remainder of this chapter we use the model as is.[44] However, the exact form of the DC model can be relaxed in different ways. For example, the parent/offspring pairing in the pairing rule is

[44] In the following chapter on symbiotic encapsulation we do not use the DC model.

not that important - the simulation results look similar though a little inferior without it (Figure 6-7, left). However, restricting the competition to similar individuals is required - if we pair-up the offspring with two random individuals instead of the two parents then diversity is quickly lost and the algorithm fails (Figure 6-7, right).

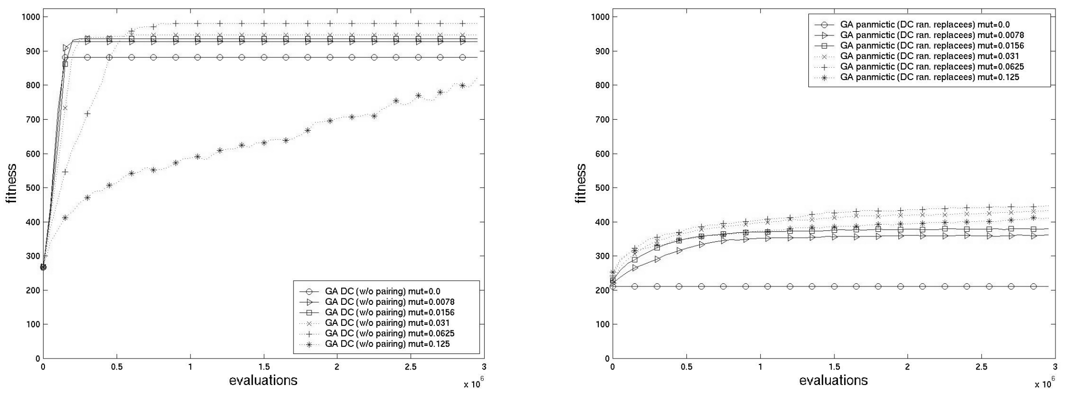


Figure 6-7: Variations on crowding algorithm

Left) Deterministic crowding works fairly well on this problem without using the pairing rule that makes offspring compete with the most similar parent. Right) But it does not work well when offspring compete with two randomly selected individuals (instead of the parents).

6.3.1 Mutation is not essential

We also see that with no mutation the GA is still successful (one of two global optima is found in 23 of 30 runs). This means that random accretive variation is actually not necessary under these conditions. This is because the population of diverse individuals, optimising different modules in different individuals, is able to supply all the required variation to discover larger modules. Moreover, when individuals are already well-adapted, random mutational variation, is so much more likely to disrupt a large module than it is to discover a new one that it is better not to use it in this situation. (Alternatively, we may imagine crossover events between different individuals that introduce small numbers of genes as being effectively equivalent to small mutations—thus additional mutation is not required.)

It is noted that the random initialisation of the population is required to cover all of the alleles at all of the loci. This is not so difficult when the number of alleles is small compared to the population size. But in other domains, for example where the variables of the system are continuous or take a very large number of possible values, it may not be plausible that the initial population covers the required alleles. In this case, a low mutation rate will presumably be valuable. Thus we do not suppose that this observation about mutation is general, however it does illustrate that there are conditions under which sexual recombination can provide all of the necessary variation, in principle.

6.3.2 Gradualism is neither sufficient nor required

It should be noted then, that in the experiments of the previous chapter we showed that, under certain conditions on this class of problem, accretive change is not sufficient to find fit configurations of the system. The simulations above now also show that mutation is not required to find fit configurations of the system. Thus under these conditions on this class of problem, gradualism is neither sufficient nor required.

We are quick to clarify however, that we do not intend to imply that gradualist models of change are not an accurate model of some, almost certainly most, evolutionary change. We make these observations just to show that there are alternate scenarios for evolutionary change that are fundamentally different from, and independent of, gradualist models. (These claims are validated more strongly in the next chapter where we gain 100% successful runs with no explicit mutation.)

6.4 Genetic linkage

In this model, it might look like sexual recombination has unequivocally provided robust adaptation in this class of adaptive landscapes. However, we have not yet discussed our assumptions about genetic linkage and these are, in fact, critical.

The experiments in the previous section used two important assumptions about genetic linkage coming from the tendency of gene subsets to occur together during crossover events:

1) That genetic linkage is significant in strength: i.e. some subsets of genes are more likely to travel together during crossover events than others by virtue of their proximity on the chromosome.

2) That genetic linkage is arranged or ordered favourably: For our purposes this means that epistatically dependent genes are close to each other on the chromosome so that those subsets that are likely to

travel together during crossover correspond well with those subsets that form modules in the interdependency network of the genes.

These are strong assumptions so it is important that we examine what happens when they are relaxed.

6.4.1 No genetic linkage: ‘Free recombination’ or ‘uniform crossover’

In the previous experiments we used one-point crossover. Clearly, whether two genes travel together from parent to offspring in a crossover event depends on whether they are on the same side of the crossover point. One point crossover can be understood as an approximation to a crossover probability of 1/N per-locus, where N is the number of genes. In multi-point crossover, the probability of two genes travelling together is inversely related to the probability that an even number of crossover points (possibly none) falls between their loci. This approaches 1 as the distance between genes approaches 0. But in general, as crossover probabilities increase, the distance between a pair of genes on the chromosome has progressively less affect in determining whether they appear together in the offspring with more than random chance (3.4.2).

In the following experiment we use uniform crossover (3.3.1). This can be understood as an approximation to very high crossover probabilities where the proximity of genes to each other on the chromosome has no effect on their probability of co-occurring in the offspring. That is, it models sexual recombination with *no* genetic linkage. In natural systems, situations with no linkage, or very weak linkage, include:

- All genes corresponding to variables in the system of interest reside in different chromosomes. In this case there is free-recombination between the genes when an offspring is produced.
- Many crossover points per recombination (A recombination in Drosophila, for example, may have many chiasma (Figure 2-6).)
- If an evolutionary system has one-point crossover, but selection pressure is low or applied only after several successive recombinations, then offspring equivalent to offspring created from many crossover points may be produced.

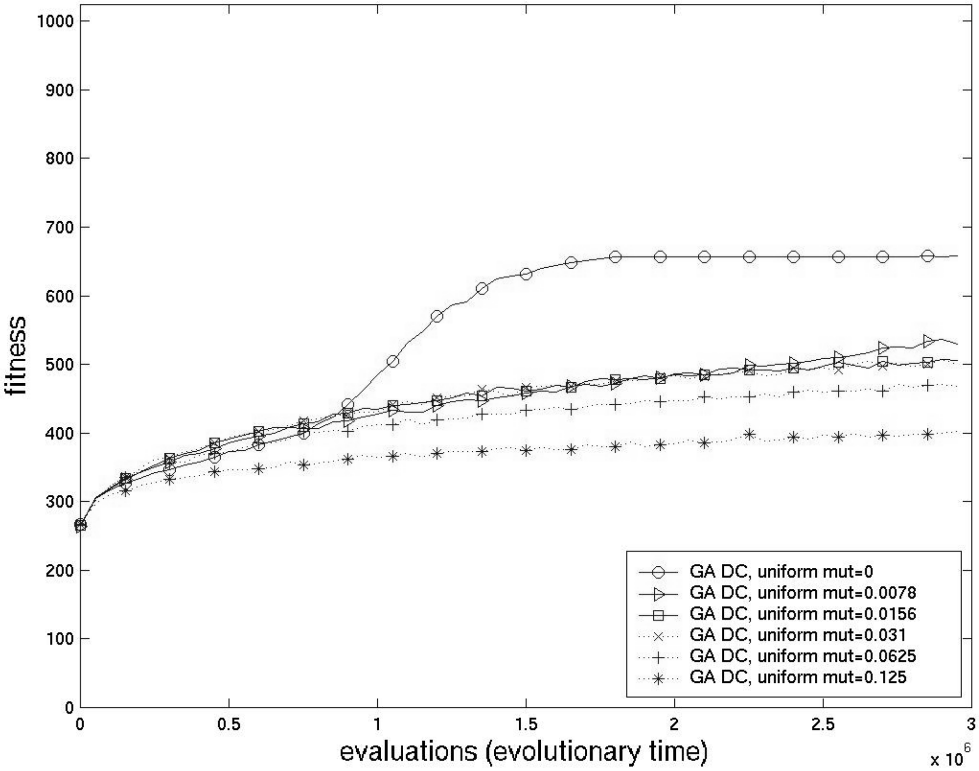


Figure 6-8: Results of DC GA with uniform crossover on a 128-variable HIFF.

HIFF (Equation 11) defines the fitness landscape. The population size is 2000 individuals. Results measure the average fitness of all individuals measured in the last 2000 evaluations (births). Several curves are shown for different mutation rates. The maximum fitness value possible in this fitness landscape is 1024. The maximum number of evaluations permitted (duration) is $3{\cdot}10^6$.

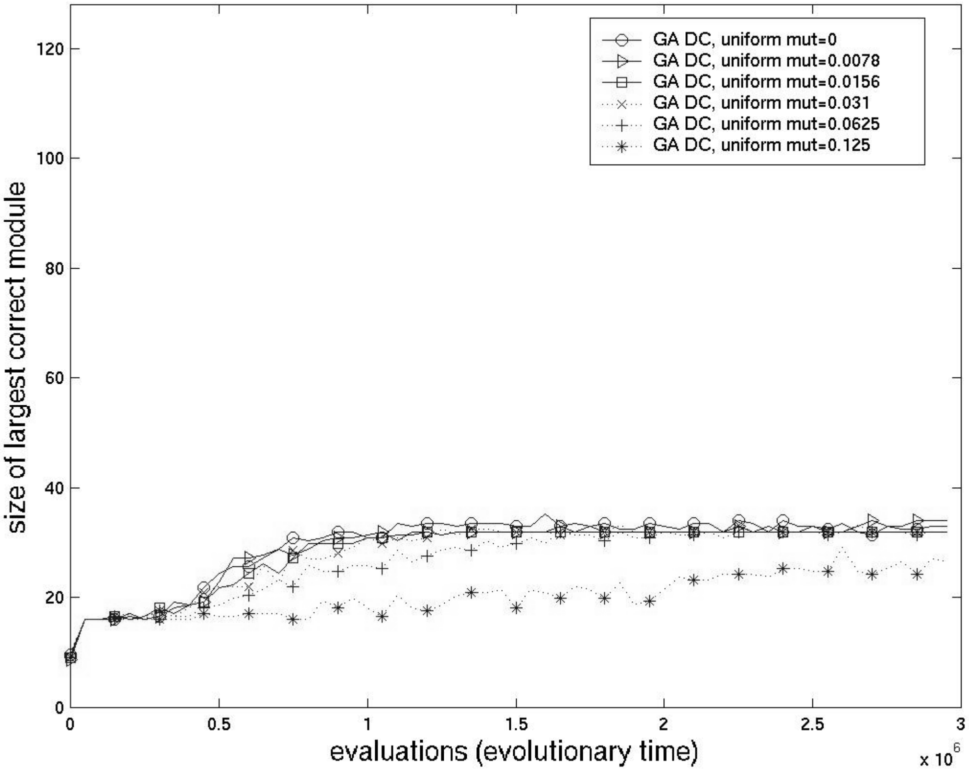


Figure 6-9: Results of DC GA with uniform crossover on a 128-variable HIFF

Details as per Figure 6-9 but showing size of largest all-ones module in population (rather than average fitness of population). (No mutation is best - see Section 6.4.3).

It is clear from these results that the absence of genetic linkage between the variables of the system in these circumstances prevents the population from finding fit adaptations. Most runs do not find modules greater than 32 bits, no runs find modules of 128 bits. This failure can be understood by considering Figure 6-10 below.

	one-point crossover	**uniform crossover**
parent 1	**1010100100000000**	**1010100100000000**
parent 2	0000000010011110	0000000010011110
offspring 1	**10101001**10011110	**1**0**1**00**0**0**1**10**000**110
offspring 2	00000000**00000000**	0**0**0**01**0**0**0**00**011**000**

Figure 6-10: Crossover operations on bit-string individuals.

> Here the genes for complementary all-0s modules are shown underlined in each parent. A particular selection of possible crossover points are indicated by the short vertical tick marks: only one for one-point crossover, N/2, on average, for uniform crossover. We show that there is a choice of single crossover point that will get both (underlined) modules into an offspring together. However, when there are many crossover points, the desirable modules will probably be broken-up in crossover.

In Figure 6-10 we illustrate crossover between two individuals that happen to be well adapted to one module each but to different modules (left and right halves of the chromosome, shown underlined). Under one-point crossover there is a 1/N chance that the single crossover point may land in the correct position to enable these two modules to come together in one of the offspring. But, under uniform crossover this is much less likely. In fact, to get the two all-0s modules into an offspring from the more extreme example strings, 00001111 and 11110000, has probability exactly equal to randomly guessing the string 00000000 ‘from scratch’ i.e. 2^{-N}. [45]

In general, the probability of getting two modules into an offspring together is related to the number of per-locus ‘disagreements’ that the parents exhibit over the loci of the modules. For example in Figure 6-10 above, parent 1 disagrees with parent 2 on the allele values for 4 loci on the left-half module, and 5 loci on the right-half module (in this example, the two modules cover the entire chromosome, but they need not in general). All loci where the parents agree, or loci that are outside the modules, have no effect on this

[45] This also applies to ‘parameterised uniform crossover’ (Spear & De Jong 1991) where the probability of taking a gene from parent-1 may be different from the probability of taking a gene from parent-2. In the example above, any advantage an imbalanced transmission probability might have through increasing the likelihood of transferring one of the modules to the offspring is lost through the decreased likelihood of transferring the other module to the offspring.

probability. But for all loci where the parents disagree, the allele that appears in the offspring must come from the correct parent. This probability is independently 0.5 for each locus under uniform crossover. Therefore the probability of both modules appearing together correctly in the offspring is $1/2^d$ where d is the number of disagreements, in this case $1/2^9$.

So, it should be clear from these considerations that regardless of the arrangement of genes, sexual recombination without genetic linkage (e.g. uniform crossover) cannot recombine conflicting modules with probability better than random mutation. From this we suggest that the compositional advantage of sexual recombination is dependent at least on there being significant genetic linkage. However, the advantage of sexual recombination with strong genetic linkage (e.g. one-point crossover) is further dependent on the arrangement of genes.

6.4.2 Re-ordered linkage: Shuffled gene positions

In the example of Figure 6-10, the successful transference of both modules to one of the offspring under one-point crossover is simply not possible unless the crossover point can be positioned such that all of the loci for one module are on one side of the crossover point, and all of the loci for the other module are on the other side of the crossover point. If the positions of genes within one module extended over the range of genes containing the other module then this would not be possible.

Figure 6-11 below illustrates a reordering of the genes for the individuals in Figure 6-10.

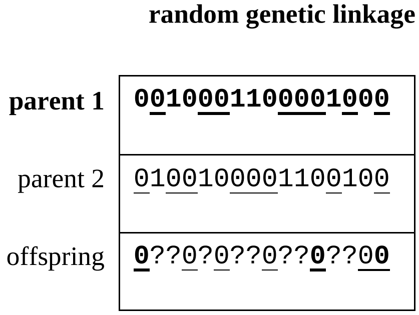


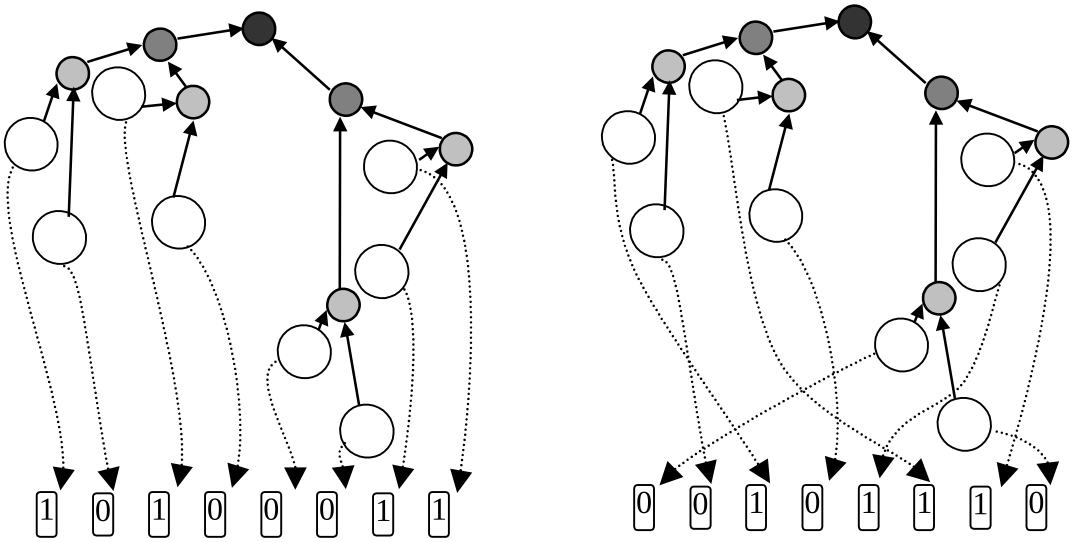
Figure 6-11: Crossover operations with random genetic linkage - i.e. random gene ordering.

Here the genes for complementary all-0s modules are shown underlined in each parent. The question is, how to chose crossover points so as to get both these modules into an offspring. (See also Figure 6-12)

In Figure 6-11 we illustrate crossover between two individuals that each have one well-adapted module but the genes from these modules are distributed along the chromosome making their combination difficult. Conceivably, uniform crossover, or in this case 8-point crossover, could place crossover points in the exact positions that allow both modules to be transferred to one offspring. But in general, we need to allow an arbitrary number of crossover points and we already know (from Figure 6-10) that recombination without genetic linkage cannot combine two modules together with probability higher than randomly guessing the resolution of all disagreeing alleles.

Accordingly, effective recombination of modules even in the presence of strong linkage also requires that genetic linkage correspond well with epistatic dependencies - i.e. epistatically dependent genes must be close together on the chromosome. We can test this reasoning by randomly re-ordering or shuffling the position of genes on the chromosome and repeating the previous experiments. There is no need to repeat the uniform crossover experiments since it is in all respects insensitive to the ordering of genes—so we just repeat the one-point crossover experiments.

For the following experiments we use “Shuffled HIFF”. That is, for each run, a different random ordering of the genes is chosen. The modules in the problem therefore do not, in general, have any structural correlation with the position of genes on the chromosome (see Figure 6-12). Figure 6-13 and Figure 6-14 show results for a GA with deterministic crowding, (as per Figure 6-4), but applied to “Shuffled HIFF” rather than HIFF.



‘Tight’ genetic linkage

Random genetic linkage

Figure 6-12: Alternate genetic linkage possibilities.

A number of genes, large circles, each contribute interdependently (solid arrows) to a number of aggregate effects, grey circles, and these contribute to higher-level features, dark circles, and so on. This hierarchy pictorially represents the modular interdependency structure of the variables in HIFF. This system of genes must be mapped to positions on a chromosome. Left) Tight genetic linkage: Ideally, genes which are epistatically related, i.e. which contribute to the same features (solid arrows), will be next to each other on the chromosome. In this case, as in regular HIFF, sexual recombination with low crossover rates (strong genetic linkage) will be able to recombine subsystems effectively. Right) Random genetic linkage: But without such a favourable situation, epistatically dependent subsets of genes may be arbitrarily positioned on the chromosome. In this case, as in “Shuffled HIFF”, sexual recombination is unable to recombine subsystems effectively (regardless of the crossover rate).

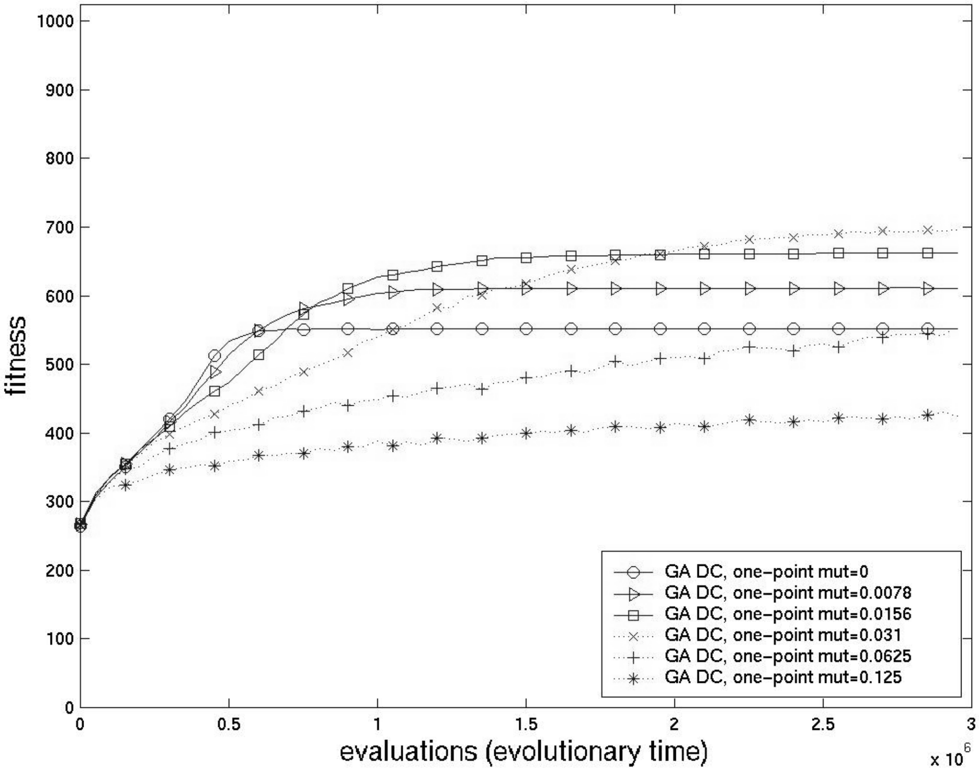


Figure 6-13: Results of DC GA with one-point crossover on a 128-variable Shuffled HIFF.

HIFF (Equation 11) with a random re-ordering of the genes defines the fitness landscape. The population size is 2000 individuals. Results measure the average fitness of all individuals measured in the last 2000 evaluations of new individuals. Several curves are shown for different mutation rates. The maximum fitness value possible in this fitness landscape is 1024. The maximum number of evaluations permitted is $3 \cdot 10^6$.

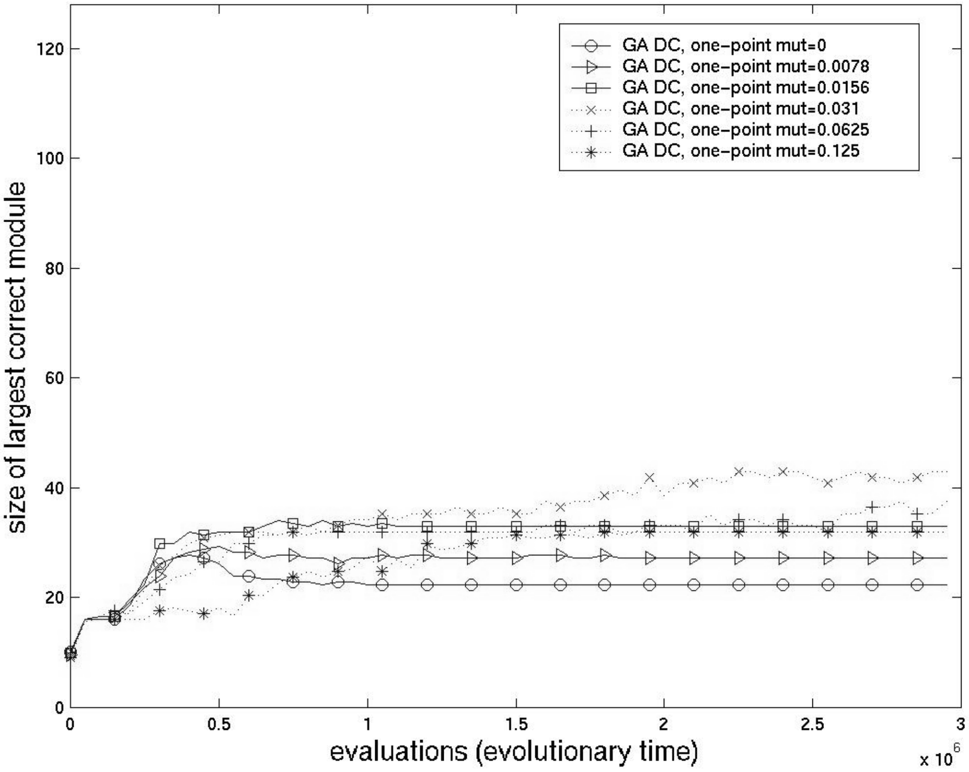


Figure 6-14: Results of DC GA with one-point crossover on a 128-variable Shuffled HIFF.

Details as per Figure 6-9 but showing size of largest all-ones module in population (rather than average fitness of population).

These results, in comparison to Figure 6-5, and Figure 6-6, support our reasoning that successful composition of modules via sexual recombination requires that epistatically dependent genes be genetically linked through their proximity on the chromosome.

6.4.3 Convergence controlled variation

Interestingly, uniform crossover does appear to do slightly better than mutation on HIFF (shuffled or otherwise). This can be understood because although the chances of transferring two different modules from different parents into an offspring are minimal, the chances of not disrupting other modules that appear in both parents is better for uniform crossover (or one-point crossover) than it is for mutation. For example, consider two parent strings A and B that ‘agree’ with one another on, say, 2 out of 4 modules, but do not agree on the other 2 - e.g. 00001100 and 00000011. Then the probability of getting from this pair of strings to either global optima by uniform crossover is $1/2^d$, where d=4 in this case is the # disagreements between parents. However, the mutation probability is much lower because it requires that the good modules of all-0s are not changed whilst the incorrect modules are mutated as is necessary. Specifically, the best mutation rate to change half the bits and not change the other half is 100% mutation (assignment of a new random allele per locus). Thus, we see that the variation applied by uniform crossover then is much like that of 100% mutation only for the modules that disagree, but no mutation at all on modules where the parents already agree. In other words, variation from uniform crossover is focussed on modules where the population has not yet converged (making an interesting form of 'convergence controlled variation' - Eshelman et al. 1996). Since, in some circumstances, this is a reasonable heuristic for focussing mutation (Chen 1999) it may perform better than ordinary 'un-focussed' mutation.

However, previous work has shown that the fact that all types of crossover preserve those parts of the chromosome where the parents agree is not sufficient to explain the success of one-point crossover on HIFF. It is in fact possible to use a crossover operator that deliberately disrupts all loci where the parents agree and applies one-point crossover on the parts that disagree. Since this still succeeds (Watson & Pollack 2000b), preserving similarity is not a required feature of a crossover operator in these circumstances. This also serves to illustrate that our selection schemes make traditional disruption analysis quite ineffective at predicting GA success under these conditions.

Interestingly, this operator, does not require the use of deterministic crowding or other diversity maintenance techniques since the operator itself prevents the population from converging. It should be noted however, that this result, though it proves the desired point that preserving the agreeing loci of the parents is not required, is dependent on the properties of HIFF (specifically the fact that competing solutions to modules are the exact complement of one another). It therefore does not offer a general problem solving mechanism, and neither does it have a biological analogue.

It should also be noted that although uniform crossover does better than regular mutation, the chances of innovation (i.e. the acquisition of a new module) by uniform crossover decrease exponentially as fitness saddles become wider, and jumps require changing more loci at once. In other words, uniform crossover is not able to exploit the decomposition available in the problem and compose together sub-modules to find new modules.

Also, one-point crossover performs better than uniform crossover - perhaps because for small schemata that happen to have short defining length, there is less disruption than there is with uniform crossover.

6.4.4 Linkage learning

In biological systems, the position of genes might be adaptively re-ordered by variation mechanisms, such as inversion and translocation, that re-order the position of genes on the chromosome. In engineering domains, favourable linkage cannot generally be assumed. Accordingly, many “moving locus” (e.g. Messy GA, Goldberg et al. 1989) and “linkage learning” schemes have been proposed (see 3.4.2). The Shuffled HIFF problem provides a good test problem for these methods and for biologically plausible models of gene re-ordering.

However, gene re-ordering mechanisms ultimately rely on a linear ordering representation of the dependencies of genes. This works well in HIFF, where there is a natural linearisation of the tree-like dependency structure (like the flattening of a hierarchical document structure into the linear order of pages in a book). But this is an approximation to the ideal linkage structure. For example, under HIFF, it should *not* be the case that the 4th and 5th genes in this string “00001111” are any more likely to travel together than the 1st and last, but of course they are when their linkage is dependent purely on linear separation. Similarly, if the span of the hierarchy is wider (the number of sub-modules per module is greater) this problem is more acute, since for example we cannot arrange three modules on a string so that all modules are equally close to each other. Moreover, it is not necessarily the case that all dependency structures can be appropriately linearised even approximately. This may mean that composition via sexual recombination has limitations even when combined with the action of re-ordering mechanisms.

Other “linkage learning” methods do not rely on gene re-ordering[46] e.g. Distribution estimation algorithms, like the Bayesian Optimisation Algorithm, BOA, (Pelikan et al. 1999). dBOA, a variety of BOA, actually

[46] In the EC literature, the generic term ‘linkage’ is used to refer to both genetic linkage (coming from the proximity of genes on the chromosome), and ‘epistatic linkage’ meaning what we have called epistatic …

works extremely well on Shuffled HIFF[47] (Pelikan & Goldberg 2000). Interestingly, the operation of the BOA algorithm can be seen to be analogous to an incremental assembly of small dependency structures into larger dependency structures (as conditional probabilities are incrementally added to an initially empty Bayesian network) and is, in this respect, a statistical method for implementing a form of compositional variation. This makes it quite different in operation from regular genetic algorithms (with or without crossover) and a very interesting direction for future modelling.

6.5 Summary of simulation results

Figure 6-15 and Figure 6-16, showing results on HIFF, clearly show that, as expected, the GA with appropriate diversity maintenance and tight genetic linkage succeeds easily (finds a maximally fit configuration in 28/30 runs), and all other algorithms examined so far fail (do not find either global optimum in any runs), on this problem under these conditions. Figure 6-17 and Figure 6-18, showing results on Shuffled HIFF, clearly show that, all the methods we have seen so far are unable to discover and manipulate modules effectively on this problem under these conditions.

dependency. These two concepts are often conflated in the EC literature since it is assumed that 'good linkage' means a tight correspondence between gene positions and epistatic dependency. In fact, the structure of epistatic dependency is quite a different issue from genetic linkage—as is clear from the fact that in Shuffled HIFF we retain the same hierarchically modular structure of epistatic dependencies whilst randomly re-ordering the positions of genes. (see 3.4.2 and Figure 6-12).

[47] dBOA performs very well in terms of fitness evaluations but it does introduce a large computational overhead in systematically analysing the fitness correlations of co-occurring gene subsets over all the strings in the population. Additionally, dBOA also requires a diversity maintenance method based on Hamming distance which happens to be very appropriate in HIFF - in the next chapter we will show a method that solves Shuffled HIFF without using the assumption that genetic dissimilarity (Hamming distance) is a good indicator of functional dissimilarity.

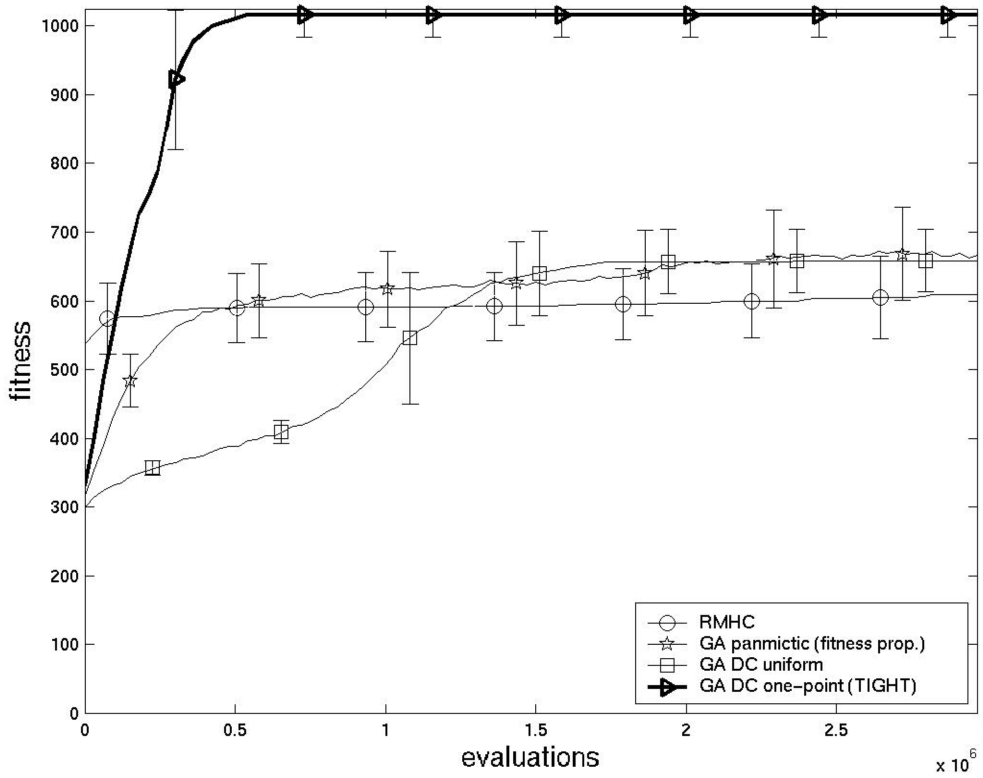


Figure 6-15: Summary of simulation experiments 1a - RMHC and GA methods on HIFF.

Simulation results for various algorithms on 128-variable HIFF. Best mutation rates (of those sampled earlier) are used for each method. RMHC - Random Mutation Hill-Climber (mut=0.0938), ‘GA panmictic’ - Genetic Algorithm without any diversity maintenance, using fitness proportionate selection, and one-point crossover (mut=0.0078). ‘GA DC uniform’ - GA with Deterministic Crowding diversity maintenance method, and no genetic linkage (i.e. uniform crossover) (mut=0). ‘GA DC one-point (TIGHT)’ - GA with DC, and strong and favourably ordered genetic linkage (i.e. one-point crossover with tight linkage) (mut=0.031). Error bars show plus and minus one standard deviation.

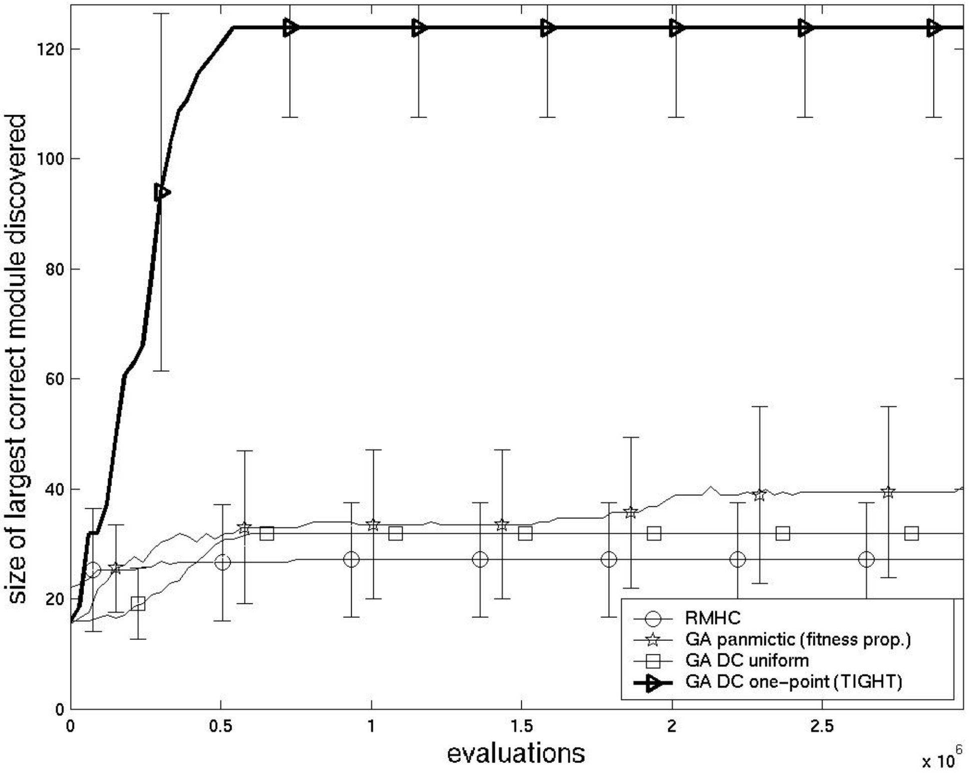


Figure 6-16: Summary of simulation experiments 1a - size of largest module discovered.

Experiments as per Figure 6-15 but showing the size of the largest module discovered.

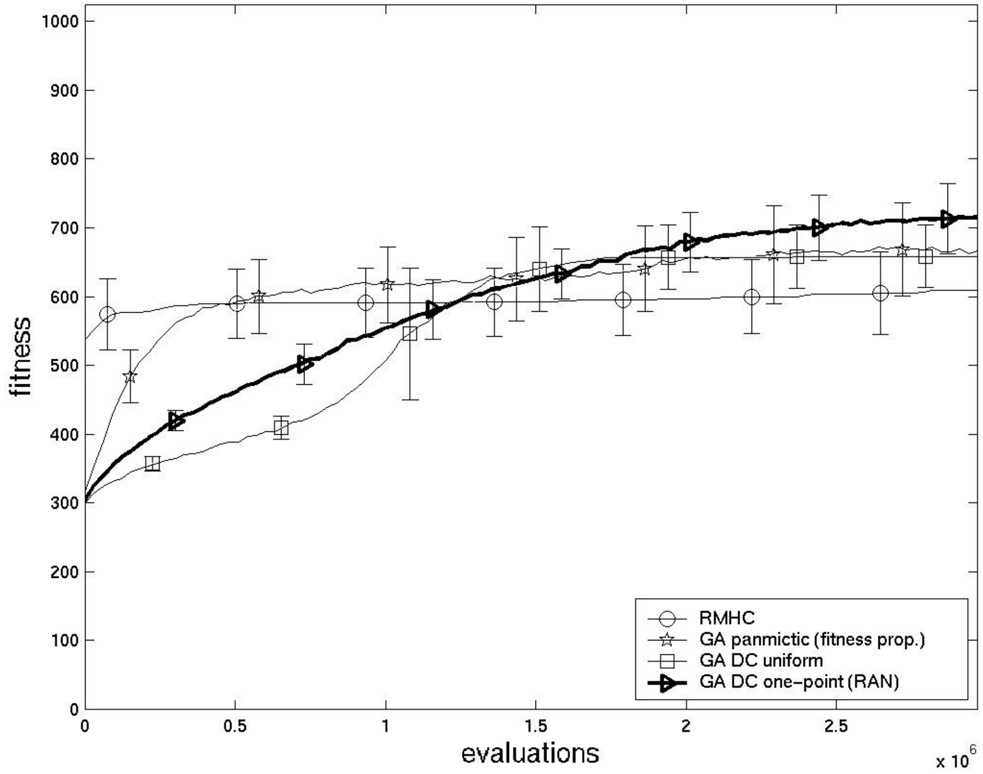


Figure 6-17: Summary of simulation experiments 1b - RMHC and GA methods on Shuffled HIFF.

Simulation results for various algorithms on 128-variable HIFF. Best mutation rates (of those sampled earlier) are used for each method. RMHC - Random Mutation Hill-Climber (mut=0.0938), 'GA panmictic' - Genetic Algorithm without any diversity maintenance, using fitness proportionate selection, and one-point crossover (mut=0.0078). 'GA DC uniform' - GA with Deterministic Crowding diversity maintenance method, and no genetic linkage (i.e. uniform crossover) (mut=0). 'GA DC one-point (RAN)' - GA with DC, and one-point crossover, and random genetic linkage (mut=0.031). Error bars show plus and minus one standard deviation.

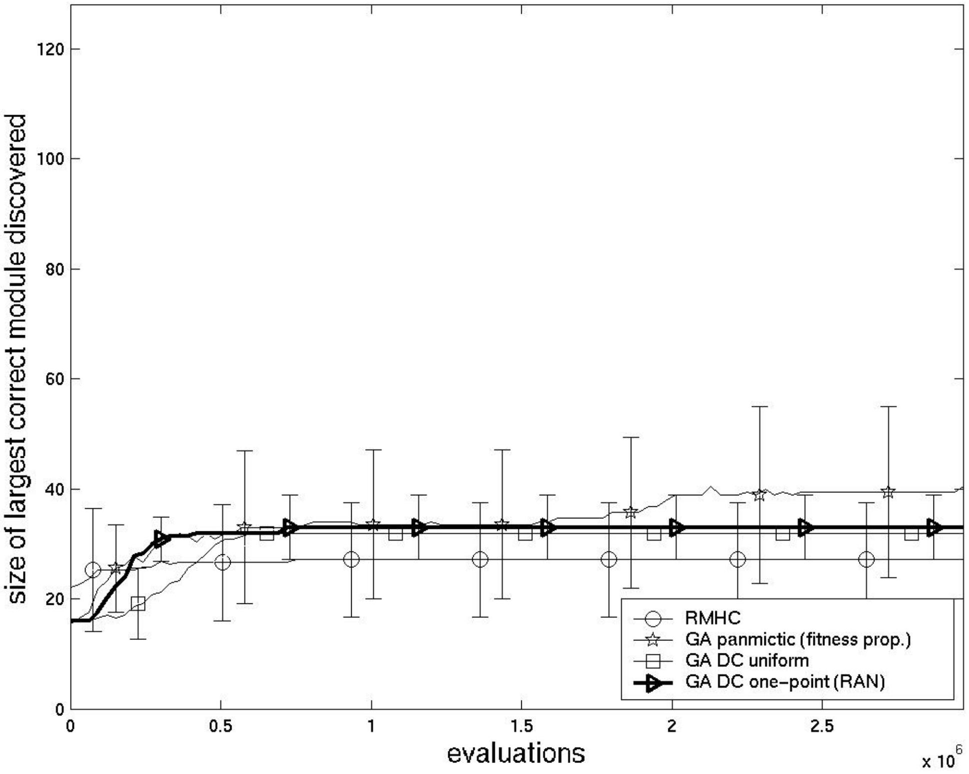


Figure 6-18: Summary of simulation experiments 1b - size of largest module discovered.

Experiments as per Figure 6-17 but showing the size of the largest module discovered.

6.6 Analysis of sexual recombination on HIFF

In this section we provide some analytic results for various recombinative algorithms applied to HIFF. For these purposes we will not attend to biological plausibility of each model because our intent is only to understand the combinatorics involved in principle.

A non-separable building block problem such as H-IFF is not amenable to some analytic approaches usually adopted in the literature. Firstly, it is not possible to apply any analysis that assumes that the population as a whole converges incrementally on particular hyperplanes. In H-IFF the operation of recombination may be defeated if the population is allowed to converge at even one locus. Secondly, most analyses assume separable problems, and, surprisingly often, focus on the extreme case where every bit is separable – the max-ones problem. However, Wright and Zhao (1999) provide an approach to analysis that, although directed at separable building block problems, can be adapted for our purposes. Their approach is to prove that there is always a way to improve fitness, and then to give a solution time based on the product of the length of the path to the solution, and the time for each step on the path. Here we extend this work to use the same approach for a non-separable problem.

It can be quite straightforward to calculate an expected time to solution when there are no local optima in a problem – that is, when there is a path of monotonically increasing fitness from any point in the search space to the solution. In this case, the expected time is simply the product of the length of this path and the expected time for each step on the path. This is the approach that we will take in the following analyses. The interesting part will be to prove the existence of such a path under recombination despite the fact that there is no such path for a mutation based algorithm. Ordinarily, such a straightforward approach would be defeated because both the existence of a path, and the time for a step, are dependent on the state of the population when using recombination. However, to calculate an upper bound on the expected time it is sufficient to know certain properties, or invariants, of the population rather than its exact state. For example, Wright and Zhao (1999) provide an analysis of a recombinative algorithm on a separable building block problem by using the property of the algorithm that prevents alleles from being lost. Under these conditions (that we will detail shortly) there is always some recombination operation that will improve the fitness of the best individual in the population.

Here we extend this idea to a non-separable building block problem i.e. HIFF. First, we analyse a ‘recombinative hill-climber’ that applies crossover repeatedly to just two strings. This simplification provides appropriate invariants that enable us to prove that there is always some choice of crossover points that will improve fitness, and to give an expected time to find such an improvement. Accordingly, we are able to give an analytical time to solution on this problem. These analyses are possible because of particular regularities in the standard form of the problem; when these regularities are removed the recombinative hill-climber fails. Nevertheless, the principle of an algorithm that follows the recombination landscape is useful to us in less restricted cases. We show that a variant of the problem that is not solvable by the recombinative hill-climber is solvable by a true GA – that is, a GA using a population. Though our analysis does not transfer directly to the true GA, a solution time based on the assumption that the GA is exploiting the recombination landscape agrees with empirical results.

6.6.1 Analysis on separable problems

The analysis provided by Wright & Zhao (1999), and the analyses that follow, feature the Gene Invariant GA (GIGA) (Culberson 1992). The variant of GIGA that best suits our purposes is described in Figure 6-19.

- Choose an initial population (see text)
- Repeat until satisfied:
    - Pick two parents at random from the population.
    - Produce a pair of offspring from these parents using crossover only.
    - If the fittest offspring is fitter than the fittest parent then
      replace the two parents with the pair of offspring.

Figure 6-19: A simple form of the Gene Invariant GA.

There are several features to note about this algorithm. Competition is restricted to parents versus their offspring. The two offspring are created from the parents using the same crossover point(s) and so each gene donated by the parents will be acquired by exactly one of the offspring. Given this and the fact that either both offspring are retained or both parents are retained (and there is no mutation) it follows that alleles are never lost from the population – hence, 'Gene Invariant' GA. Finally, note that the algorithm is elitist – the fittest member of the population cannot be replaced by an inferior individual.

Wright & Zhao add several other assumptions to this model: the problem is separable; a set of crossover masks is used that restrict crossover to operations that move exactly one block from one parent to the other; finally, the population is systematically initialised such that every possible bit combination within each block is present. This initialisation gives a population size of c^k, where c is the size of the alphabet and k is the number of bits in a block.

These simplifying assumptions enable a useful property to follow: given the gene invariance property of the algorithm and that crossover points are not permitted to move partial blocks, it is guaranteed that building blocks are never lost from the population; since all possible candidates for a block are present in the initial population and they can never be lost, there will always be some individual in the population that has any block that should be required. Thus, if the best individual in the population is not yet optimal then it must have some sub-optimal block, and this block can be obtained from some member of the population using some crossover mask. This is the invariant property of the population that is required for the analysis. Although the exact structure of the population is not known, this property ensures that there is always some way to improve fitness using recombination.

To calculate an upper bound on the expected time, *T*, required for the population to find an individual that has reached the global optimum[48] Wright & Zhao focus attention on the fittest individual in the population. Their version of the algorithm asserts that one of the parents is the fittest individual (and the other parent is selected at random). The possible states that the algorithm may pass through in the course of search, i.e. the possible populations, are then categorized into equivalence sets based on the fitness of the fittest individual. *T* is then given by the maximum possible number of fitness increases of this fittest individual, and the time expected for each increase. This yields $T=Br(D/d)$, where *B* is the number of blocks in the problem (equals the number of crossover masks), *r* is the size of the population, *D* is the difference in fitness between the global optimum and the worst possible string, and *d* is the difference in fitness between the global optimum and the next best string (i.e. the minimum fitness increase for finding a correct block).

In Theorem 1 below we modify the proof provided by Wright & Zhao and use the same assumptions about the problem, crossover masks, and initialisation. However, Wright & Zhao categorize the state of the algorithm using the fitness of the fittest individual in the population and always use this individual as one of the parents. But, we will categorize the state of the algorithm using the number of fully optimised blocks in a particular individual, which we will always use as one of the parents. But, this individual may not be the fittest individual in the population. Indeed, although the fittest individual possible (the global optimum) must have all blocks fully optimised, the fittest individual in the population at a given time does not necessarily have any blocks fully optimised.

[48] In Wright & Zhao (1999), time, *T*, is measured in steps of the algorithm in Figure 1, but each step requires two evaluations.

In order to transform the theorem to work on the number of optimised blocks we need to be sure that we never reduce the number of optimised blocks when we accept an offspring. We cannot directly measure the number of fully optimised blocks in an individual; neither can we infer the number of fully optimised blocks from a fitness measure. However, we can use the following observation: a fitness increase created by changing one block in an individual cannot reduce the number of fully optimised blocks in that individual. Figure 2 describes a modified algorithm that ensures this condition is applied.

- Choose an initial population (as before)
- Pick one parent at random from the population, *p*1.
- Repeat until satisfied:
    - Pick one parent at random from the population, *p*2.
    - Produce a pair of offspring from *p*1 & *p*2 using crossover only.

      Let *c*1 be the offspring that results from *p*1 plus one block from *p*2; let *c*2 result from *p*2 plus one block from *p*1.
    - If *c*1 is fitter than *p*1 then replace *p*1 with *c*1, and *p*2 with *c*2.

Figure 6-20: A modified Gene Invariant GA.

This algorithm is almost the same as the algorithm in Figure 6-19, but the replacement is more specific about which offspring is compared to which parent. Note that the algorithm only selects one new parent in each iteration, and only evaluates one of the new offspring. This means that maybe the second offspring, *c*2, was fitter than the one evaluated – and maybe we missed an opportunity to increase fitness. But for our purposes, it is more important that we know that *p*1 does not decrease in the number of optimised blocks. Although, the algorithm in Figure 2 explicitly makes reference to the fact that the partitions of blocks are known, and that the crossover masks only swap one block at a time, we are not adding any new assumptions to those used by Wright and Zhao. Importantly, note that the modified algorithm maintains the gene invariant property, and the property that building blocks are never lost from the population.

> **Theorem 1:** An upper bound on the expected time for some individual in the population to find the globally optimal string, using the algorithm in Figure 2, is given by $T \leq B^2 r$, where B is the

number of blocks in the separable problem and, r is the size of the population ($r=c^k$, where c is the alphabet size and k is the number of symbols in a block).

Proof:[49] Partition the set of possible algorithm states into categories based on the number of optimal blocks in $p1$. Category, j, for $j=0,1,\ldots,B$, is the set of states where $p1$ has exactly j blocks fully optimised. Let Sj be the random variable which denotes the number of evaluations that the algorithm spends in category j. If the GA is in category j with $j<B$, then $p1$ has some block which is not at optimum, and there is a crossover operation that will swap in the optimal configuration of bits for that block from some other member of the population. This operation moves the algorithm state to a new category and has a probability of at least $1/(Br)$ since there are B crossover masks and r possible choices for the second parent. An upper bound for the expected time to leave any state is the inverse of a lower bound on the probability of leaving that state. Thus, the expected value of Sj, at $j<B$, is at most Br. The expected time to reach a population that contains the globally optimum string is at most the time it takes for the algorithm to reach category B, i.e.

$$T \leq \sum_{j=1}^{B} S_j \leq B^2 r \qquad \text{[end].}$$

The intuition behind this analysis is now very simple. T is given by the product of the number of blocks to be found, and the time to find a block in another individual and move it in by crossover. The number of blocks is B, and getting a block into the best individual requires finding the right crossover mask (of which there are B), and picking the right donor individual (of which there are r) – hence B^2r, or $B^2(c^k)$.

This expected time has the desirable property that it is not dependent on any measurement of fitness values in the problem. And the resultant upper bound for T is lower than that which Wright & Zhao provide, at least whenever $B<(D/d)$. Both analytic times assume that each crossover mask corresponds to exactly one block. However, Wright and Zhao's analysis could be modified to a more general set of masks, giving T ≤ $Mr(D/d)$, where M is the number of crossover masks used. This set of masks may include masks that match with more than one block so long as they only match with whole blocks, and it must include the set of

[49] This proof is a direct adaptation of that provided by Wright & Zhao (1999).

masks that correspond to the single fitness blocks, as before. In any case, an approach based on counting optimised blocks, rather than fitness improvements, is better suited to our needs in the following sections.

So far we have an upper bound on the expected time to solution for a recombinative algorithm on a separable building block problem. However, we have to ask whether the simplified problems and algorithms involved in these analyses have captured the interesting properties of a GA. We have permitted modifications to the algorithm that do not violate the assumptions made about the problem – but we notice that we are getting gradually closer to something much simpler than a GA. In fact, notice that using only the same assumptions about a problem as those used above (i.e. that the partitions of building blocks are known), a simple systematic hill-climbing technique would suffice. That is, in a separable problem with known partitions, each subset of interdependent parameters may be searched exhaustively – i.e. each block may be processed systematically by testing every possible bit combination within that block and selecting the highest fitness configuration. Whilst one block is being processed, we hold the setting of bits in other blocks constant to some arbitrary configuration. The particular configuration of bits used in other blocks does not matter since we know that the blocks are separable. Since systematic search within a block requires testing c^k bit configurations, and there are B blocks to be processed, this simple systematic search takes only $T \leq Bc^k$.

Further, for their analysis using GIGA, Wright & Zhao suggest that if the partitioning of blocks is not known then their analytic time can be multiplied by the number of possible partitionings. Since this can be applied to our simple systematic hill-climber too, by this analysis, the simple method is still superior even if the partitioning is not known. We make this observation not to criticize the approach of Wright and Zhao, rather we present it as an illustration that separable problems simply do not justify the use of a population-based recombinative technique. Nonetheless, the above version of Wright & Zhao’s analysis provides a good starting point for an analysis of GIGA on H-IFF which is not separable and, we shall argue, does require a recombinative algorithm.

6.6.2 Analyses on HIFF

The approach of Wright and Zhao provides an appropriate tool for analysing the action of sexual recombination on HIFF. Their approach is quite intuitive: their algorithm (GIGA detailed shortly) and assumptions guarantee that there is always some choice of parents for which some crossover operation can produce an offspring that is the next step in a progression of individuals which increase in fitness. In other words, their approach is to prove that separable building block problems have no local optima using their algorithm. We follow the same approach but for non-separable building block problems, and whereas the proof of Wright and Zhao is based on the fitness of an individual, our approach, as used in Theorem 1, is based on the number of blocks that are fully optimised in an individual.

> **Condition 1:** There is always some choice of parents for which some crossover operation can produce an offspring that is the next step in a progression of individuals which increase in the number of fully-optimised building blocks. (Except when the progression has already arrived at a global optimum.)

This condition is central to our approach in this section. It refers to the ability of an algorithm to utilize the building block structure of a problem. Or alternatively, to an invariant property of a population that enables an algorithm to utilize the building block structure. Any algorithm that satisfies Condition 1 and follows such a progression will take time $T \leq BS$, where B is the number of blocks to be found in a globally optimal string (equals the maximum number of steps in the progression), and S is the maximum expected time for each step in the progression. For a recombinative algorithm where Condition 1 holds, $S \leq PM$, where P is the number of different ways of choosing parents, and M is the number of crossover masks, thus

$$T \leq BPM.$$

Equation 12: Expected time to solution in terms of steps on path to optimum

Equation 12 will hold for any problem that can be described in terms of blocks in some way such that Condition 1 holds. Proving that Condition 1 holds (and that an algorithm follows this progression) will be more or less difficult depending on the assumptions made about the algorithm and the problem. In the separable problem addressed in Theorem 1, Condition 1 holds because all possible configurations for each block are given by initialisation and cannot be lost by the variation operators that the algorithm uses. In Theorem 1, $M=B$, and $P=r$, the size of the population. We cannot expect to prove Condition 1 for a standard recombinative algorithm on a general non-separable problem. However, H-IFF is specifically designed to be easy for a recombinative algorithm and, GIGA is ideally suited for adaptation to our purposes.

6.6.3 Analyses for a recombinative hill-climber

For Theorems 2 and 3 we will reduce GIGA to a form of hill-climber – by which we mean that we will not use a population to speak of – just two strings. However, this will be a recombinative hill-climber not a mutation-based hill-climber, Figure 6-21.

- Initialise a population of two random, but complementary, strings.
- Repeat until satisfied:
  - Using the two strings as parents create a pair of offspring by crossover only.
  - If the fittest offspring is fitter than the fittest parent then replace the parents with the pair of offspring.

Figure 6-21: Recombinative hill-climber based on GIGA.

Hoehn and Reeves (1996) call recombination like that above "complementary crossover". Alternatively, one might call it a macro-mutation hill-climber (MMHC) from (Jones 1995) that uses bit-flip mutation (instead of assignment of a new random value). Arguably, this is a degenerate form of recombination. Hoehn and Reeves point out that a second parent is redundant. And we concede that the choice of complementary parents is ideally suited to the complementary schemata rewarded in H-IFF. Nonetheless, we maintain that it is more informative to regard the above algorithm as recombining two strings (that happen to be complementary) rather than bit-flipping. Hoehn and Reeves agree that there is a close relation between this operator and crossover, and suggest that the fitness landscape under this operator "reasonably approximates the crossover landscape". The purpose of using this algorithm here is to illustrate the concept of the crossover landscape as it is found in H-IFF, and to provide a basis for understanding how a true GA might operate on this class of problems by avoiding the local optima inherent in the mutation landscape.

The following theorems do *not* assume that the algorithm is provided with a set of crossover masks corresponding to blocks in the problem. Ordinary crossover may be used with no restriction on where crossover points may fall.[50] Theorem 2 is based on two-point crossover, Theorem 3 is based on one-point crossover - in the latter case it is a little more difficult to show that Condition 1 holds. Note that these analyses are not restricted to just the one problem instance defined in Equation 11. H-IFF has regular block sizes within a level, regular fitness contributions for blocks within a level, exactly two sub-blocks per block, and global optima that happen to be all-ones and all-zeros. The analyses hold for the class of problems with any of these particulars relaxed, but for these analyses, the global optima must be complementary, and the fitness contribution of competing schemata in the same partition must be the same. We will discuss in Section 0 an extension to the class where building blocks are not necessarily complementary.

Theorem 2: $T \leq ½N^3$, where T is an upper bound on expected time to find a global optimum in H-IFF using the algorithm in Figure 6-21 with 2-point crossover, and N is the problem size in bits.

Proof: From Equation 12, $T \leq BPM$ if Condition 1 holds. For H-IFF, B, the maximum number of blocks that need to be discovered in a globally optimal string is

$$\sum_{p=0}^{(lg\,N)-1} 2^p = \text{N-1},$$

thus $B<N$ (Here, and henceforth, we use “lg” to mean logarithm base 2). For two-point crossover, M, the number of possible pairs of crossover points is ½N(N-1)<½N^2. And for the algorithm given in Figure 6-21, the number of different ways to choose parents, P, is just 1. Thus, $T \leq ½N^3$ if Condition 1 holds. Now prove that Condition 1 holds: The two strings are complementary at initialisation and since the gene invariance property of the algorithm holds the strings must always be complementary. This means that if a block is discovered in one individual the complementary block must necessarily be discovered in the other individual. Therefore any block not present in one individual can be swapped in from the other individual given the right crossover mask. Two-point crossover is a superset of the crossover masks required for this crossover to occur. The

[50] Tough in this section we are assuming tight linkage.

algorithm will detect such a successful crossover and follow this progression since it will always result in a fitter individual. So Condition 1 holds. [end]

Theorem 3: $T \leq N^2$, where T is an upper bound on expected time to find a global optimum in H-IFF using the algorithm in Figure 6-21with 1-point crossover, and N is the problem size.

Proof: Following the proof of Theorem 2, the number of crossover masks for one-point crossover $<N$, and thus, $T \leq N^2$ if Condition 1 holds under one-point crossover. Now prove that Condition 1 holds: If an individual is non-optimal then it contains some block that either has both of its sub-blocks correct (but not matching) or has some sub-block incorrect. This applies recursively. Consider the smallest incorrect block that has both of its sub-blocks correct (possibly just a pair of bits) and consider the crossover point between them. Since no fitness contribution can occur for a block that spans the crossover point (because it would include mismatched sub-blocks), the fitness of this individual is the sum of fitnesses of the sub-strings to the left and right of this point. The other individual used in the algorithm has the same fitness as the first, and its fitness is the same sum of fitnesses from the left and right sub-strings defined by the crossover point. Thus, if this crossover point is used, the fitness of the resulting offspring will have at least the same contributions from the two sub-strings and in addition, since the two individuals are complementary, the crossover will necessarily introduce matching sub-blocks that create a new correct block spanning the crossover point. Thus there is always some one-point crossover operation that will increase the number of correct blocks in the best individual, i.e. Condition 1 holds. [end]

Figure 6-22 illustrates the reasoning for Condition 1 under one point crossover.

11010001 Consider an arbitrary string as an example. This size-8 string is sub-optimal. It is composed of two incorrect size-4 blocks. So we recurse on either of the two incorrect size-4 blocks, for example, the left block.

1101---- This size 4 block is composed of a correct size-2 block (left), and an incorrect size-2 block (right). So we recurse on the incorrect block i.e. right.

--01---- This size 2 block is sub-optimal. But, it is composed of two correct size-1 blocks, that are incompatible.

110|10001 So, we place the crossover point between the two size-1 blocks (bits) at this point. The fitness of the entire string can be expressed as the sum of F_left=f(110------) on the left hand side of the crossover point, and F_right=f(---10001) on the right of the point.

001|01110 The other string used in the algorithm is the exact compliment. And the left and right sides of the crossover point have the same fitnesses, i.e. F_left and F_right.

110|01110 The string created by combining the left of the first string and the right of the second string, still has at least the fitness of F_left + F_right.

--0|0---- But in addition, it now has an additional correct block here, created across the crossover point.

Figure 6-22: Illustrating that Condition 1 holds.

Illustrating that Condition 1 holds using recombinative hill-climber on H-IFF with one-point crossover. Since there is always some crossover point that comes between correct but incompatible blocks, it is always possible to use this point to create a string that has higher fitness by using one-point crossover.

Thus far we have proved that Condition 1 holds on H-IFF for the algorithm in Figure 6-21 using both 1-point and 2-point crossover. This shows that although a mutation hill-climber finds a number of local optima that is exponential in *N*, H-IFF has no local optima at all under recombination – there is always some way to increase the number of correct blocks, as Condition 1 states. In this sense, H-IFF exemplifies the transformation of a fitness landscape under different operators from a problem that is very hard under mutation into a problem that is as easy as it could be under crossover. Having shown that there is always a path to an optimum, we now focus on improving our estimate of the time to traverse this path.

We may improve on the upper bounds given in theorems 2 and 3 by returning to our original reasoning, $T \leq BS$, where B is the number of blocks to be found (i.e. steps on the path to the optimum) and S is the expected time to find a block (i.e. time to take a step). In the above theorems we prove Condition 1 to assert that there is at least one crossover operation that will find a block given some choice of parents. So we have used $S \leq PM$, and accordingly, $T \leq BPM$. But it should be clear that for this algorithm on H-IFF there will be many opportunities to increase the number of correct blocks. For example, at the beginning of search there are many size-two blocks to be found and many possible crossovers that may find one.

So, we can use an estimate of time to progress along the path to the optimum that takes account of the fact that the expected time for a step changes with each step, as the number of possible ways to find a block changes. The time to find one of q available blocks is PM/q, thus;

$$T \leq \sum_{b=1}^{B} \frac{PM}{q_b}$$

Equation 13: 12 more accurately

> where B is the maximum number of steps in the path to the optimum, P is the number of choices of parents, M is the number of possible crossovers, and q_b is the number of ways that an additional block can be swapped-in at the b^{th} step.

Thus, we may write:

$$T \leq PMu,$$

Equation 14: 13 simplified

where u is the sum of $1/q$ for all steps in the path to the optimum. Theorem 4 uses an upper bound on u to give an improved time to solution on H-IFF.

> **Theorem 4:** $T \leq ½M\lg^2 N$, where T is an upper bound on the expected time to find a global optimum in H-IFF using the algorithm in Figure 4, M is the number of crossover masks (i.e. $M=N-1<N$ for 1-point crossover, $M=½N(N-1)$ for two-point), and N is the problem size in bits (for $N>20$).
>
> **Proof:** To use $T \leq PMu$ we must find u which is the sum of $1/q$ over all steps in the path, where q is the number of blocks that may be discovered at that step. At the first hierarchical level in H-IFF there are $N/2$ size-2 blocks to be discovered (in the worst case, we may assume that the initial

strings have none of their size-2 blocks correct). By the reasoning of Theorems 2 or 3, any one of these blocks may be discovered by one or two-point crossover. Thus q_1=(N/2). There are now N/2-1 blocks remaining to be found at the first level, so q_2=(N/2-1). For the entire first level we require the sum of 1/q_1 through 1/$q_{(N/2)}$=1+1/2+1/3+…+1/(N/2) ≤ ln(N/2)+1. The sum of 1/q for the pth level is ≤ ln(N/2^p)+1. The overall sum of 1/q for all levels, u, is

$$u = \sum_{p=1}^{lg\ N} \sum_{k=1}^{N/2^p} \left(\frac{1}{k}\right)$$

$$\le \sum_{p=1}^{lg\ N} \left(ln\left(\frac{N}{2^p}\right) + 1\right)$$

$$= \sum_{p=1}^{lgN} (lnN - p\,ln2 + 1)$$

$$= lgN\left(\frac{lgN}{lge} + 1\right) - \frac{ln2}{2}\left(lgN(lgN + 1)\right)$$

$$= \frac{ln2}{2} lg^2 N + \left(1 - \frac{ln2}{2}\right) lgN.$$

For the algorithm in Figure 4, P=1. So,

$$T \le Mu = M\left(\frac{\ln 2}{2} \lg^2 N + \left(1 - \frac{\ln 2}{2}\right) \lg N\right)$$

Thus T is $\boldsymbol{O}(M\lg^2 N)$, and numerically, $T \le \frac{1}{2} M \lg^2 N$, for N>20. [end]

Summary of analysis for recombinative hill climber

	one-point crossover	two-point crossover
theorem 2	N^2	
theorem 3		½ N^3
theorem 4	½ $N\lg^2 N$	½ $N^2\lg^2 N$

In summary, Theorem 4 reduces the upper bound on the expected time for the recombinative hill-climber by at least a factor of N to $\lg^2 N$ in each case.

6.6.4 From a recombinative hill-climber to a recombinative population

Theorems 2 through 4 use the recombinative hill-climber of Figure 6-21, and in fact, demonstrate that a population greater than this degenerate case of two is not strictly necessary to solve the canonical form of H-IFF. However, this is only the case because of biases in the algorithm that match particular properties of H-IFF. Specifically, GIGA is biased toward finding complementary individuals (a bias which becomes 'the rule' when the population is of size 2), and H-IFF has competing schemata that are exactly complementary. It is quite easy to break this non-population based version of GIGA by making the global optima in H-IFF non-complementary. We can do this by choosing two random strings as the global optima and rewarding left and right sub-strings recursively. Alternatively, and without loss of generality, we can keep one of the optima at all ones and randomise the other. Either way, the bits in the two global optima will agree in some loci and be complementary in others. This prevents the algorithm given in Figure 6-21 from succeeding since the complement of a good block is no longer (necessarily) a good block. Equation 4 defines a variant of H-IFF based on this idea, that we will refer to as H-IFF2. $\boldsymbol{F}(A)$ runs through the string twice using $\boldsymbol{f}$ and sums results; before the first pass the string is XORed with one global optimum, and before the second it is XORed with the second global optimum. $\boldsymbol{f}(A)$ now simply checks for blocks of all zeros. The fitness of a string, A, under H-IFF2 is given by:

$$\boldsymbol{F}(A)=\boldsymbol{g}(A\otimes \mathrm{g1})+\boldsymbol{g}(A\otimes \mathrm{g2}),$$

$$g(s_1,...,s_N)=\begin{cases} 1 & ,if \quad N=1 \\ Nf'(\mathrm{S}_1,...,\mathrm{S}_\mathrm{k})+g(<\mathrm{S}_1>)+...+g(<\mathrm{S}_\mathrm{k}>) & ,otherwise \end{cases}$$

Equation 15: HIFF2 = HIFF with arbitrary global optima.

where $p\otimes q$ is the bit-wise exclusive OR of *p* with *q*, *g1* and *g2* are the two global optima, and the other variables are as per Equation 11: i.e. S_i is the i^{th} variable of the configuration, $<S_i>$ is the i^{th} disjoint sub-partition of the variables, i.e. for 'unshuffled'[51] HIFF, $<S_i>=(s_{1+k(i-1)},\ldots,s_{ki})$, *f'* is the base fitness function, defined below. $N=k^H$ where $H\in Z^+$ is the number of hierarchical levels in the system or subsystem, and k is the number of sub-modules per module.

$f'(p_1,\ldots p_k)=1$ if $(\exists s\forall i\text{: } p_i=0)$, and 0 otherwise.

Note that f' is modified to accept only blocks of zeros as high fitness configurations. Competing schemata are introduced by the fact that the two global optima are not identical. Note also that HIFF and H-Equal are the special case where the two global optima are the string of all zeros and the string of all ones.

In the experiments that follow we will use the all-ones string as *g1*, and *g2*= "010101…" (i.e. loci with an odd index take the value 0, and the even loci take the value 1). Thus the two optima have exactly 50% of bits in agreement and 50% complementary. The proofs of Theorems 2 through 4 are not valid for the function in Equation 4 – it should be clear that the recombinative hill-climber cannot succeed on this problem. However, if we re-introduce an adequate population we can recover a successful algorithm. Unfortunately, we are not able to prove a time to solution on this problem using a population. However, we will see that an estimate based on *assumed* invariants of the population gives reasonable times. Specifically, we use the rather heavy-handed assumption below. This assumption states explicitly that diversity is being maintained appropriately for recombination to work effectively. It supposes a level-wise discovery of blocks and that all members of the population progress together.

> **Assumption 1:** When looking for a block at level *p*, all individuals in the population consist of complete blocks from level *p*-1, and these blocks will be a sub-string of either global optimum, *g1* or *g2*, with equal probability.

[51] We will also consider random linkage or 'shuffled' versions of HIFF in later experiments.

Theorem 5: $T \leq ½N^2 \mathrm{lg}^2 N$, where T is an upper bound on expected time to find a global optimum in H-IFF or H-IFF2 using a GA with two-point crossover, where Assumption 1 holds, and $N>20$ is the problem size.

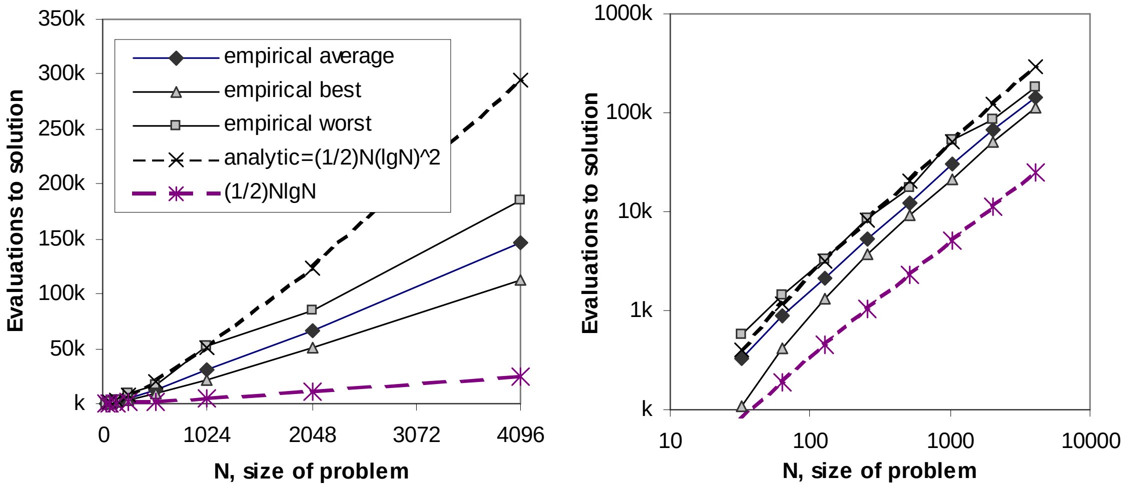
Proof: Using $T \leq Pmu$, (Equation 14). Without the condition that every pair of parents are the exact compliment of one another we cannot apply the reasoning of one-point crossover from Theorem 3. But the reasoning for two-point crossover from Theorem 2 applies with a slight modification. Since we do not assume that the parents are necessarily complementary but rather they consist of correct sub-blocks of either type with equal probability, half of the blocks swapped-in will make no improvement. The sum of $1/q$, where q is the number of ways to make an improvement at each step, is therefore twice the value of u calculated in Theorem 4. Given that Assumption 1 refers to all individuals in the population, the same opportunity for improvement is available for any choice of parents, so $P=1$. The expected time to solution is therefore twice that of Theorem 4, with $M=½\mathrm{N}(\mathrm{N}-1)<½N^2$ for two-point crossover. [end]

This estimate of the expected time for a GA on H-IFF takes no account of the population size, selection pressure, or any other aspect of the algorithm – all of these are embedded in Assumption 1. Nonetheless, our empirical results given in the next section give times to solution that are under this upper bound on expected time and, as we will discuss (6.7.3) support the validity of the assumption for our experimental set-up.

6.7 Empirical illustration of analytic results

6.7.1 Simulation for Theorem 4: Recombinative Hill-climber with one-point crossover on H-IFF.

To validate Theorem 4 we implemented the algorithm of Figure 6-21 with one-point crossover. Figure 6-23 shows the results of 30 runs on each problem size $N=32$ doubling to $N=4096$. All 30 runs were successful at every size. We show the fastest, the slowest and the mean time to find the solution from each N. The analytical time, $T \leq ½N\mathrm{lg}^2 N$, provides a good upper bound. It appears to be too high by a factor less than lg$N$ (see ½Nlg$N$ curve for comparison). Figure 6b) shows the same data on a log log scale.



a) linear scale

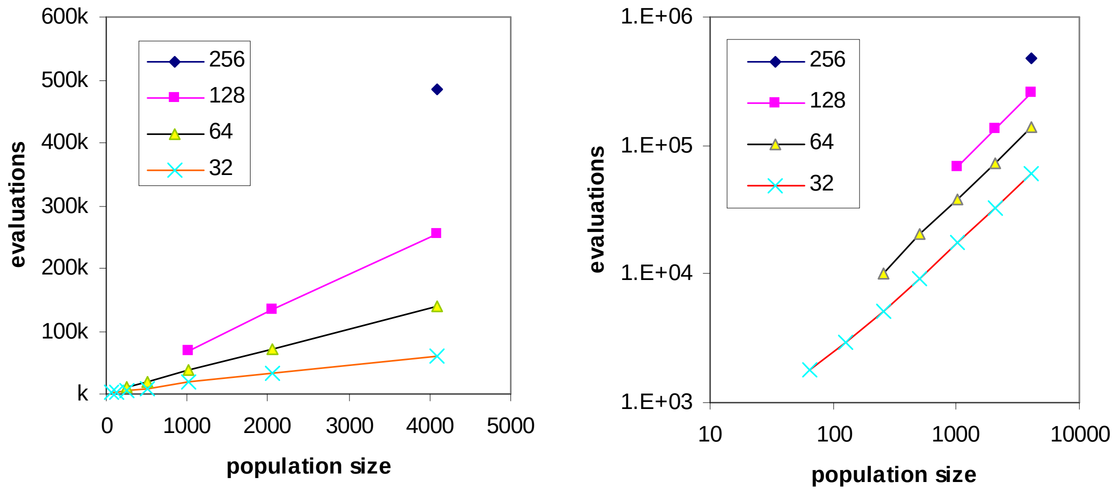
b) Data as per (a), on log log scale.

Figure 6-23: Performance of recombinative hill-climber on H-IFF.

6.7.2 Simulation for Theorem 5: GA with two-point crossover applied to H-IFF2

Theorem 5 is not based on any particular GA and empirically, we find that the algorithm in Figure 6-4, deterministic crowding, works faster and more reliably than GIGA on this problem. This is possibly because deterministic crowding allows some convergence whilst also segregating competition and thereby maintaining diversity.

Figure 6-24 shows the solution times for the algorithm in Figure 6-4 for *N*=32 doubling to *N*=256 and for population sizes from 32 doubling to 4096. Average solution times over 30 runs are shown only for those population sizes that succeeded on at least 90% of runs (in an evaluation limit of 200 times the population size). For example, only the population size 4096 succeeded reliably on *N*=256. We see that the time to solution is approximately linear with population size for those sizes that succeed. We also note that for each doubling of *N*, the minimum population size that succeeds reliably quadruples (See log log scale in Figure 6-24 b).

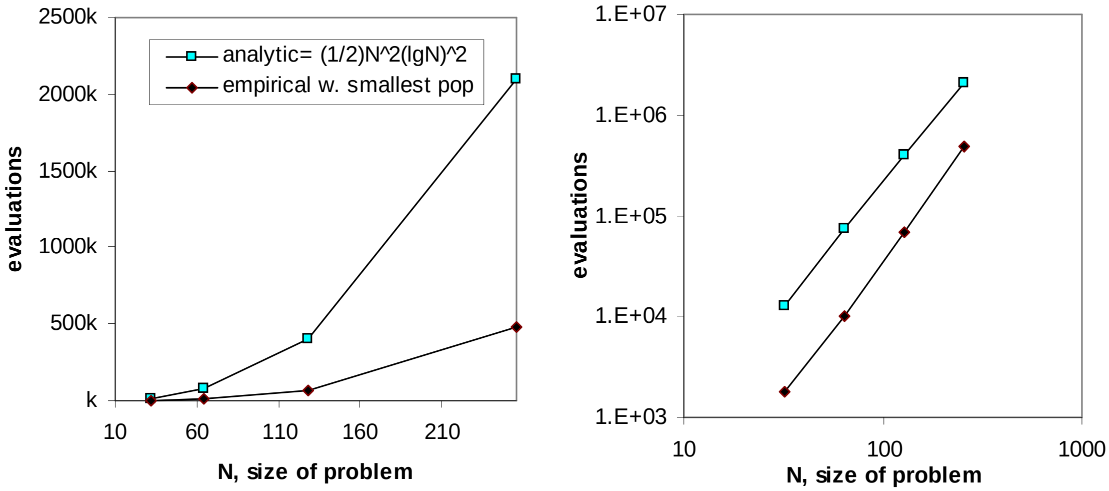


a) linear scale

b) Data as per (a), on log log scale.

Figure 6-24: Performance of GA on H-IFF2, various problem sizes & population sizes.

Since we are interested in whether there is any configuration for the GA such that time to solution is reliably better than our upper bound, we now focus on the smallest population size that succeeds reliably. We extract the average time to solution for each *N* using this population size, i.e. the first point on each curve. These points are compared with analytic time, from Theorem 5 in Figure 6-25. This expected time, $T \leq \frac{1}{2} N^2 \lg^2 N$, provides an overestimate of the experimental time.



a) linear scale

b) Data as per (a), on log log scale.

Figure 6-25: Performance of GA on H-IFF2 using smallest reliable population.

The fact that the empirical time is better than our analytic time does not necessarily mean that Assumption 1 was correct. However, we can make some statements about the operation of the algorithm from this result. Note that the algorithm uses elitist replacement; an individual can only be replaced by a new offspring if the new individual is fitter. An algorithm incorporating such a replacement strategy cannot succeed unless its variation operators successfully manipulate the search space so as to ensure that there is always at least one way in which a fitter individual can be created from the current population. Note also, that in H-IFF, since all blocks within a level are the same fitness, and higher level blocks must contain correct lower-level blocks, superior fitness can only arise from a greater number of correct blocks. Thus we know that since this algorithm succeeds then there must be a progression of individuals with monotonically increasing number of correct blocks, i.e. Condition 1 must be true. Further, we may say that an algorithm that performs better than our analytic time, as this algorithm does, either has more opportunities to find a next step on a path to the optimum, or has a shorter path than we than we thought. These observations suggest that the GA is able to properly exploit the decomposable structure of H-IFF by following the crossover landscape.

6.7.3 Summary of analysis section

This section has provided an analytic time to solution for a recombinative algorithm on a particular non-separable building block problem. The upper bound on expected time is based on proving a path to the optimum and the time for each step on the path. In the limited case of a population of two we have proved that the expected time to solution is at most $\boldsymbol{O}(N\lg^2 N)$, where N is the size of the problem in bits. This is illustrated empirically. For the more general population case we provide a time $\boldsymbol{O}(N^2\lg^2 N)$ based on the assumption that, at any time, the state of the population is such that the algorithm is able to provide recombination steps that follow the path we have described. We see that there is a population size for which the GA succeeds reliably and our analytic time to solution is an overestimate.

6.8 Summary of sexual recombination on hierarchically modular problems

We have seen that there are some circumstances where sexual recombination is able to compose together modules to find larger modules. And we have seen previously that this process is not available to accretive mechanisms such as mutation in this class of adaptive system. The conditions for this successful manipulation of modules include:

- Appropriate modular interdependency structure (e.g. a decomposable structure like that of HIFF). Reasoning in the previous chapter suggests that the structure of HIFF exemplifies the difference in adaptive capacities of accretive and compositional mechanisms. We previously discussed that HIFF is amenable to a divide and conquer problem decomposition in principle, and the operation of sexual recombination in these experiments illustrates exactly this process in operation.

- Population diversity. If different members of the population are not able to maintain competing solutions to modules (different members of M_A for a module A, see 4.3.1), at least as long as is required to test different combinations of modules and resolve their interdependencies (i.e. to test M_AxM_B, for two modules A and B), then the performance of sexual recombination will degrade to approach that of mutation. Thus appropriate diversity in the population is required. We have demonstrated that the diversity provided by the deterministic crowding model of population subdivision is sufficient.

- Strong Genetic Linkage. If genetic linkage is weak or absent, as modelled by uniform crossover, then the genes of a module cannot be transferred to an offspring as a whole to find new combinations of modules. Thus strong genetic linkage is required, and we have demonstrated that one-point or two-point crossover is sufficient.

- Tight genetic linkage. Even if genetic linkage is strong, if genetic linkage between genes does not correspond well with the epistatic dependency between genes then modules cannot be recombined effectively. More generally, the recombination of modules as wholes through sexual recombination is effective only to the extent that gene ordering happens to correspond with the structure of those modules. We demonstrated that such an arrangement is possible for HIFF.

It should be clear that although the GA with deterministic crowding and one-point crossover successfully exploited the hierarchical dependency structure of the HIFF problem in Figure 6-5, the GA did not ‘discover’ this modular structure in any sense. It is only able to utilise the structure when the structure is given to it via the appropriate ordering of genes on the chromosome. Thus it can only exploit the dependency structure of the problem if it happens to coincide with the genetic linkage structure that is given. It can discover the solution to modules and recombine them if the biased way in which strong-linkage crossover samples the space of possible recombinations is suitable to exploit the structure of modules in the problem, but it cannot discover this structure in the problem for itself. In an engineering domain with difficult dependencies between variables where we do not have *a priori* knowledge of the dependency structure, we are not able to order genes appropriately on the chromosome and thus sexual recombination is not likely to provide the algorithmic advantage of compositional evolution.

In the following chapter we investigate a mechanism of composition that is not sensitive to the ordering of genes in the problem as given, or put another way, a mechanism that is able to dynamically discover and explicitly use the learned dependency structure of the genes, so as to recombine modules effectively through successive hierarchical levels.

Chapter 7 - Symbiotic Encapsulation on Modular Interdependency

In this chapter we investigate a compositional mechanism based on symbiotic encapsulation—the joining of entities from different reproductive lineages into a new reproductive whole. This model is based on compositional change in the major evolutionary transitions whereby “entities that were capable of independent replication before the transition can replicate only as part of a larger whole after it” (Maynard Smith & Szathmary 1995). And more specifically, the model abstractly represents a mechanism of “symbiosis followed by compartmentation and synchronised replication”, (ibid.).

Our purpose in this chapter is to show that the adaptive capacity of composition is not dependent on *a priori* knowledge of which genes are dependent of which other genes, or the assumption that this is implicitly provided by favourable gene ordering. In the process, we gain some understanding of the differences in the adaptive capacity of sexual recombination and symbiotic encapsulation.

In the following subsection we outline the basic framework of the symbiotic encapsulation model. In the subsequent subsections we detail the components of the encapsulation model and illustrate its adaptive capacities using simulations on HIFF. Following this we give a brief analysis of the expected time to discover complex adaptations on HIFF. In the subsequent discussion chapters (Chapters 8 & 9) we discuss related work and theories in EC and EB.

7.1 Overview of symbiotic encapsulation model

The composition model that we introduce in this chapter is called the “Symbiogenic Evolutionary Adaptation Model”, or “SEAM”, to invoke the notion of symbiotic union or joining.

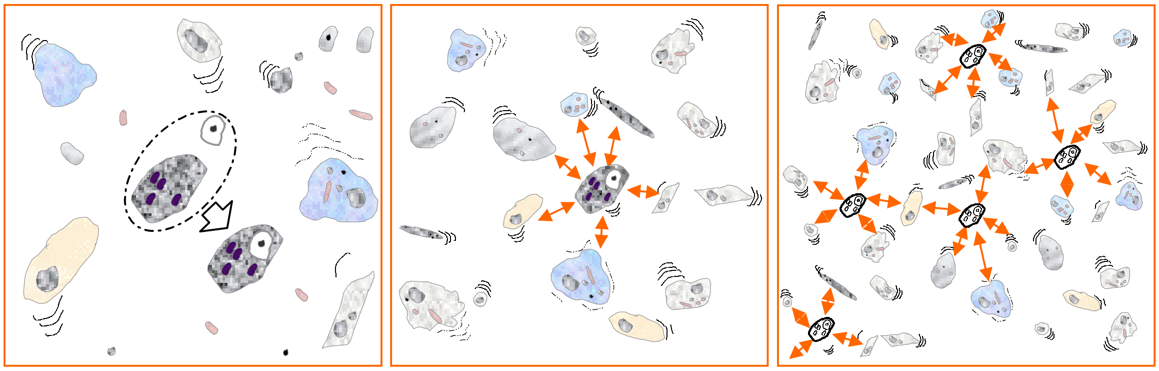
This model includes the idea of integrating the features adapted in different lineages into a new whole simply by encapsulating extant entities into new higher-level entity - i.e. ‘symbiotic encapsulation’ - and this is the fundamental distinction from the model of sexual recombination. But this new integration model has implications for our selection model too. As a consequence SEAM begins to look quite different from the model of sexual recombination in the previous chapter. But, in later discussion we will outline the similarities and analogies between the two models and show that in many respects they are not that different. Specifically, we will see that both can be seen as a model of evolution that acts at more than one level of selection: crudely, on modules and on assemblies of modules. In sexual recombination the modules are represented by tightly-linked schemata and the assemblies are individuals. In SEAM, the modules are represented by individuals and the assemblies are made from the explicit joining of individuals. But in the meantime, by introducing SEAM in the manner that we do below, we provide a stepping stone to a more general understanding that allows us to see the sexual recombination model in a new light.

The three main features of SEAM that we will introduce in the following subsections are:

- Variable-sized entities and a mechanism of creating new entities by joining extant entities.
- Evaluating entities in transient groups of other entities.
- A selection scheme based on testing the ‘stability’ of a proposed join in many environmental contexts.

In overview, SEAM develops as follows. The ecosystem is initialised with many different small entities. Pairs of entities are then picked at random to see if they might form a stable symbiotic join. If the overall fitness of either entity alone could be, dependent on environmental contexts, greater than the fitness of the entity with the proposed symbiotic partner then the composition is deemed *unstable* and the original entities are returned to the ecosystem. Otherwise the composition is deemed *stable* and the two entities always co-occur together in future. The process of building and selecting compositions of entities is repeated, eventually building larger and larger composite entities.

Three main features of SEAM are depicted in Figure 7-1. Frames (a) through (c) in the figure loosely correspond to variation, evaluation, and selection, respectively. These processes, outlined in the figure, are detailed in the subsequent subsections.



a) New entities are created by joining two existing entities together.

b) The fitness of an entity is dependent on its environmental context.

c) An entity is placed in many contexts to test the stability of a new join.

Figure 7-1: A caricature of processes in SEAM.

a) New entities are created from the composition or joining of two randomly selected extant entities (Section 7.2.2, Figure 7-3). b) The fitness of any entity (possibly the result of a previous join) has dependencies with its environmental context, i.e. a random selection of other entities from the ecosystem (Section 7.3.3, Figure 7-4 & Figure 7-5). c) The new pairing is subject to many such contexts. If there is some environment of other entities in which either component of the join is fitter individually than when it is with its proposed partner, then the join is deemed unstable and is dismantled. This follows the assumption that the partnership must be in the 'selfish' interest of the partners involved. In our implementation, the stability of a proposed join is tested in many contexts and is immediately undone if found to be unstable. This models the assumption that competition between joined and non-joined variants of an entity occurs rapidly such that only reliably successful joins persist long enough to be involved in a subsequent join (Section 7.3, Equation 18). A join that persists through (c) is treated as a new entity that may participate in further joins as the cycle of the model repeats.

7.2 Integration model: Entities and their encapsulation

7.2.1 Mutually exclusive characters and the 'overlap' of feature sets

In Section 2.2.1 we listed a spectrum of compositional mechanisms with sexual recombination at one end, and symbiotic encapsulation (e.g. endosymbiosis and symbiogenesis) at the other. Whereas the previous chapter modelled sexual recombination, this chapter is concerned with a model of composition based on symbiotic encapsulation at the other extreme of this spectrum.

The ordering of mechanisms in the spectrum that we emphasised in Section 2.2.1 was based on the similarity or dissimilarity of the entities being composed: specifically, the entities involved in sexual recombination are of the same species, whereas composition can occur between individuals across species. A second distinction is found in the relationship between the genetic material provided by the 'donor' entities and that appearing in the composite. Specifically, whereas sexual recombination involves alignment and crossover exchanging corresponding subparts of the chromosome and taking on average half of the genetic material from each parent, the result of endosymbiosis approaches the union of genetic material from the donors—taking all the genetic material from both parties involved.

The alignment of these two trends in the spectrum of compositional mechanisms—similarity/dissimilarity of entities, and 'half-&-half'/union of genetic material—may not be arbitrary. We illustrate with a toy example. Consider a set of characteristics that can be subdivided into subsets where members within each subset are mutually exclusive but members across subsets are not. For example, the characters {TALL, ROUGH, SMOOTH, SHORT} can be subdivided into {TALL, SHORT} and {ROUGH, SMOOTH} by these criteria. For convenience, let us call each subset of mutually exclusive characters a *feature:* for example, the feature HIEGHT has two possible (but mutually exclusive) characters, namely TALL and SHORT. The features then are a dimensionalisation of the possible character combinations. Now consider the result of combining an entity specifying for {SHORT} with an entity specifying for {TALL}; the result, by our characterisation, cannot be both tall and short—it must be one or the other.[52]

i.e. {TALL} + {SHORT} → {TALL **XOR** SHORT}

Whereas, the result of combining an entity specifying for {short} with an entity specifying for {smooth}, by our characterisation, could be an entity that is both short and smooth.

i.e. {SHORT} + {SMOOTH} → {SHORT **AND** SMOOTH}

A natural interpretation of mutually exclusive characters is that they reside on the same loci and are mutually exclusive in the haploid chromosome: i.e. genes are *features,* and the alleles of a gene are

[52] Biologically, a blending, resulting in medium height for example, is another reasonable possibility in some cases. But still there is no possibility that the result can be both fully-tall and fully-short, and on average, the characteristics of the donor entities can only be 'half inherited' in the offspring. Similarly, a mixture, resulting in some of the stalks within a plant being tall and the remainder being short is also neither fully-tall nor fully-short, and on average, the characteristic of the donor entities can only be expressed in half the offspring.

mutually exclusive *characters* of the feature. But this is a particular instantiation of characters and features which presupposes a mechanism of meiosis which is not strictly necessary. That is, if by some circumstance, alignment and crossover failed and entire duplicate chromosomes resulted in the offspring, as in polyploidy events, then the conclusions about some characters being mutually exclusive and some being compatible would still hold even though all alleles of both the parents were genetically represented in the offspring.[53]

In this scenario then, entities resulting from the combination of entities specifying variations of the same feature set will specify the same feature set themselves and will on average specify half the characters of each parent (assuming unbiased dominance effects, etc.). Whereas, entities resulting from the combination of entities specifying variations of different (non-overlapping, i.e. not mutually exclusive) feature sets could conceivably specify the union of features from the donor entities. If we define entities specifying for the same (or largely overlapping) feature sets as *similar* (regardless of the values of these features) and entities specifying disjoint (or largely non-overlapping) feature sets as *dissimilar* then a similarity/dissimilarity trend will coincide with a half&half/union trend in resulting entities, as seen in the spectrum of composition mechanisms we listed.

From this point of view we see the mechanisms of sexual recombination and symbiotic encapsulation as specific types of a larger class of integrative mechanisms. That is, both mechanisms take the characters of two donor individuals and form a new entity from the combination of their characters. But in sexual recombination, the donor individuals are similar in the sense that their specified feature sets are largely overlapping, whereas in symbiotic encapsulation the donor entities may be dissimilar in the sense that their feature sets are largely non-overlapping. This has the consequence that the result of sexual recombination events will express on average approximately half the characters of either donor, whereas the result of symbiotic encapsulation will express approximately the full union of characters from both donors. This expectation follows logically from the definition of features as sets of mutually exclusive characters, and the assumptions about the similarity of donors in respect to overlap in the set of features they possess, rather than from the nature of the genetic mechanisms involved.

[53] Similarly, if two alleles of a gene are not intrinsically mutually exclusive then gene duplication could conceivably result in an entity with both alleles.

Similarly, this general perspective is useful in understanding the expected results of other mechanisms in the compositional spectrum—See Figure 7-2. For our purposes, it is not the genetic mechanisms themselves that are important in these different phenomena, so much as the expected overlap of features coming from the similarity or dissimilarity of the donor individuals.

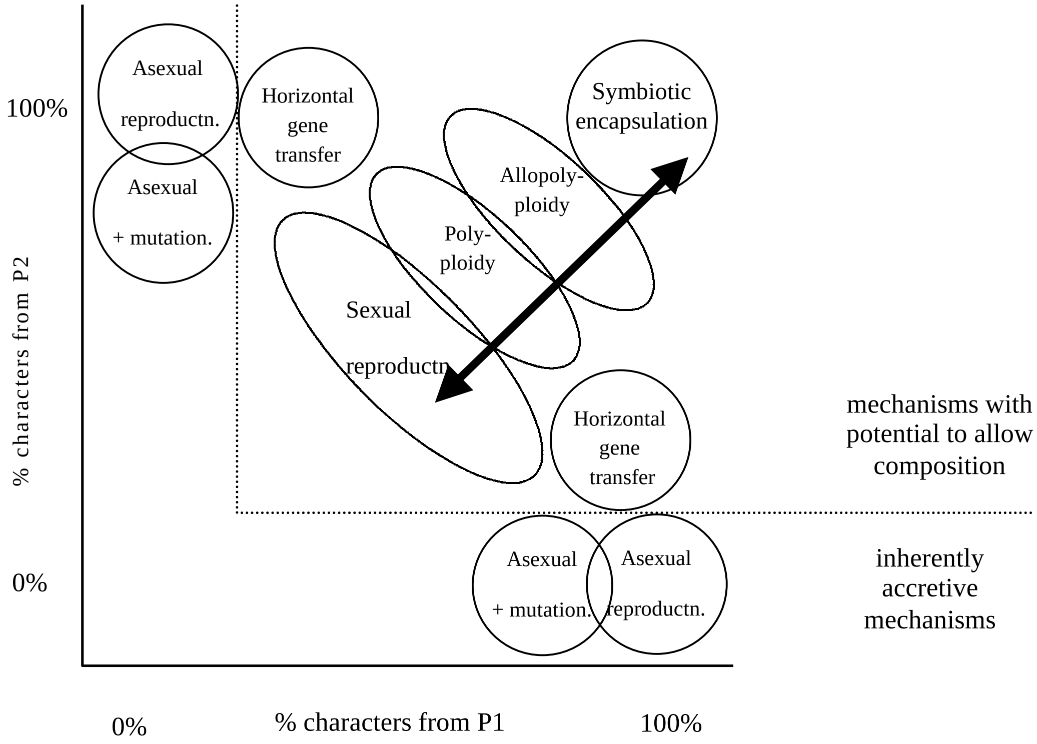


Figure 7-2: Mechanisms of integration from two donor individuals, P1 and P2.

The axes are the percentage of characters from each donor entity that is likely to be exhibited by the result of integration. The arrow indicates the spectrum of compositional mechanisms in discussion. Our assumptions here are as follows: a) Asexual reproduction maintains the characters of the parent, and does not involve characters of a second parent; b) Mutation reduces the heritability of characters from a parent; c) sexual reproduction results in an offspring exhibiting on average half the characters of parent one and half from parent 2; d) horizontal gene transfer, introduces some of the characters from one 'parent' into the other 'parent' but, depending on how similar the parents are, some characters may also conflict and reduce the heritability of characters from the recipient individual; e) Polyploidy, in a manner similar to sexual recombination, will exhibit approximately half the characters from either parent on average, despite the fact that the offspring may contain the union of genetic material from either parent. But, this depends on how similar/diverse the feature sets of the donors are. f) As the feature sets become more different, involving chromosomes from different species (i.e. allopolyploidy), the characters of the parents will be less conflicting and 'offspring' may

> involve on average more than half of the characters from both donor individuals; g) Symbiotic encapsulation, when occurring between wholly unrelated entities, may result in the union of characters from the donor entities. In summary, mechanisms that take characters from only one donor individual are inherently accretive, those that that take characters from two donor individuals are potentially compositional.

Following this framework, the symbiotic encapsulation mechanism we detail in this chapter is much like that of the sexual recombination mechanism detailed previously but we have different assumptions about the nature of the entities involved. Specifically, in the sexual recombination model we assumed (as is quite normal) that the sets of features exhibited by the donor individuals were fully overlapping. Accordingly, recombination of entities produced conflicts in characters (assuming the population was not fully converged) that needed to be resolved to create a new entity.[54] In contrast, in the symbiotic encapsulation model we detail shortly, we assume that the entities are dissimilar from one another and generally non-overlapping in the sets of features they specify. We detail our representation of entities and their encapsulation in the next section.

7.2.2 A model of entities and their encapsulation

The focus of the model in this chapter is the notion of a mechanism that composes together dissimilar entities; entities that are mostly non-overlapping in the features they specify, producing resultant entities with the union of their characteristics. These entities may be interpreted as genes, bacterial cells, more complex cells, or any other higher level of organization—the intent is to model transitions between these levels in an integrated model of ‘entities’. We will use the word ‘species’ to refer to types of entity at any level. Entities are represented only by their *features values*, and for now, species are simply the set of entities with identical feature values. These features may be interpreted as genes, as phenotypic features corresponding to genes, or as higher level features of an organism such as resource usage or a behavioural strategy. In general, they are the set of characteristics that affect the fitness of an entity and the fitness-sensitive interactions of the entity with its environment and other entities. Our model abstracts away all population dynamics within a species and therefore the ecosystem will only incorporate one representative

[54] Specifically, which of the mutually exclusive parental-alleles was to be inherited by the offspring was resolved randomly under uniform recombination, and according to genetic linkage under one-point crossover.

entity of each species. An appropriate level of metaphor is that the set of entities represented in the model is an ecosystem of species (rather than a population of individuals).

The basis of our composition model will be that a composite is created from the joining of features from two different species of entity. Accordingly, it is necessary that different entities will specify different subsets of features (not just different values of the same set of features). To provide a simple example: let each feature take one of two values, “0” or “1”, and let the features be identified by an index, F_n. Then one entity might specify features F_3=0, F_7=0, F_{12}=1, and a second species might specify F_1=1, F_9=0, F_{10}=1, F_{15}=0. Then their join may create the new entity with features F_1=1, F_3=0, F_7=0, F_9=0, F_{10}=1, F_{12}=1, F_{15}=0. (See Messy GA ‘splice’ operation, Figure 3-9).

We will use a large finite set of possible features for simplicity in the implementation of the model,[55] but the maximum number of features could be flexible in alternate implementations. The number of features specified by any one entity may be anything from one to the full set. In this way it is simple to write the specification of an entity using a fixed length string. For example, working in a 16-feature space, we may write the two entities given in the example above as ***A*** and ***B*** in the left of Figure 7-3 below, and their composition may be written as ***A+B***.

The ‘null features’, “-”, in this representation are ‘placeholders’ for features that are not (currently) specified by an entity. We will refer to the ‘size’ of an entity to mean the number of non-null features—for example, the entities used above have sizes 3, 4 and 7, respectively. Figure 7-3 also illustrates how we will deal with conflicting specifications when they arise.

```
  A:  --0---0----1----          A:  ----1----00-1--
  B:  1-------01----0-          B:  --1-0---0-1----
----------------------      ----------------------
A+B:  1-0---0-01-1--0-        A+B:  --1-1---000-1--
```

Figure 7-3: Symbiotic encapsulation

[55] Note that in the GA individuals also generally use a finite set of binary features, ‘genes’, but unlike the entities in SEAM, individuals in the GA must generally specify a value for every possible gene. This is natural for a model of evolution within a single lineage where every individual has basically the same features but varies in the values of these features. The ‘null’ value used in the implementation of SEAM, detailed shortly, cannot reasonably be characterised as a third allele since it is not heritable in the same way as non-null values (see Figure 2).

(an abstraction of Figure 7-1:a). Left) Composition of two variable size entities, A and B, produces a composite, C, that is twice the size of the donor entities with the union of their features. Here we represent unspecified features by "-". The composite is created by taking specified (i.e. non-null) features from either donor where available. Right) Where conflicts in specified features occur we resolve all conflicts in favour of one donor, e.g. the first.

Algebraically, we define the composition of two entities A and B, as the superposition of A on B, below. $A=(A_1,A_2,\ldots A_n)$, is the entity where feature F_i takes value A_i. S(A,B) is the superposition of entity A on entity B, and s(a,b) is the superposition of two individual feature values, as follows:

$$S(A,B)= S((A_1,A_2,\ldots A_n),(B_1,B_2,\ldots,B_n)) = (s(A_1,B_1),s(A_2,B_2),\ldots,s(A_n,B_n)),$$

$$\text{where, } s(a, b) = \begin{cases} a, \text{ if } a \neq \text{null}, \\ b, \text{ otherwise.} \end{cases}$$

Equation 16: Composition of two entities

This composition will be the only mechanism of variation in our model. The intent is that the model will start from 'primtive', i.e. single-feature, entities and compose them together into larger composites, and compose these together, and so on. When small entities are composed with relatively large entities, their effect is like single-feature mutations, but as entities become larger, their composition enables variations that scale-up with their size.

Note that the way we use species in this model has no implication of restricting possible unions based on type—in principle, new entities may be created by the composition of any two existing entities regardless of their species, i.e. regardless of the features they specify.

7.2.3 Consequences for adaptation

Our algorithmic intuition is that different small entities (with a small number of specified features) will adapt to different simple sub-parts of the adaptive domain and that complex adaptations may subsequently be formed by composing them together. In this respect, the algorithmic intuition is largely the same as that for sexual recombination. However, since sexual recombination between entities with overlapping features sets must involve a half&half combination operator, sexual recombination has the problematic aspect that the extraction of *appropriate* subsets of characters from each parent is dependent on strong genetic linkage and the favourable ordering of genes on the chromosome (i.e. 'tight' linkage). That is, the combination operator needs to be biased to take subsets of features that correspond to coherent modules, because otherwise modules cannot be selected for and manipulated as wholes.

In contrast, our intuition is that a mechanism of composition using entities with non-overlapping characters will enable the entities themselves to represent modules, (as distinct from the idea that a module may be represented by a subset of genes *within* an individual). If a whole entity represents a module, and entities can be combined together to provide larger modules, then it is not necessary to extract subsets of genetic material from within a given entity. In other words, the appropriate subsets of genes are 'linked' by virtue of being specified within some entity, rather than by virtue of being proximal to each other on the chromosome, and thus the position of genes along the chromosome within an entity becomes irrelevant.

7.3 Evaluation and selection model

Having defined a variation operator that defines a join of two entities, we need to determine whether such a join would be selected for. Our basic assumption is that the symbiotic relationship must be in the 'selfish' interest of both the component entities involved. That is, if the fitness of either component entity is greater *without* the proposed partner than it is *with* the proposed partner then the composite will not be selected for. If, on the other hand, the fitness of both component entities is greater when they co-occur then the relationship is deemed stable and will persist.

However, in the model we are using, entities are only partial specifications of feature sets and we are assuming that the features of one entity have interdependencies with the features of another. In other words, the fitness of any entity is dependent on its environmental context; possibly, in one environment an entity may be fitter when co-occurring with the proposed symbiont, and in another context the symbiosis may depreciate its fitness. Thus whether a symbiotic relationship is preferred or not depends on what environmental contexts are available. (i.e. the preference for a composite is interdependent with context - see Section 4.6.6).

7.3.1 Evaluating an entity in different contexts

For our purposes, the set of possible environmental contexts is well defined: an environmental context is a complete set of features (in which some partially specified entity, which may be the result of many joins, can be assessed). See Figure 7-4.

`---0-11---110---`	***x***,	an entity specifies a partial set of feature values.
`0110101100010011`	*θ*,	an 'environmental context' is a complete set of feature values.
`011`**`0`**`1`**`11`**`100`**`110`**`011`	***S***(***x***,*θ*),	the entity ***x*** superimposed on the context *θ*.

Figure 7-4: A partially specified entity must be assessed in a context.

We assume that the overall fitness of the entity will be a sum of its fitness over different environmental contexts weighted by the frequency with which each environment is encountered. But, we would not generally suppose that the frequencies with which different environments are encountered by one type of entity would be the same as the frequencies relevant to a different type of entity. That is, we imagine that different species have different distributions over possible environments.

Let us assume that we have a measure of the 'context sensitive fitness',[56] ***csf***(***x***,*θ*), of an entity, ***x***, in any given environmental context, *θ*, and that the overall fitness of the entity ***x***, will be ***F***(***x***) which is the sum of its fitness over all environments weighted by the frequency of that environment for that species, as below.

$$F(p)=\sum_{\theta\in Contexts}\left(\lambda_{(\theta,p)}csf(p,\theta)\right)$$

Equation 17: Weighted sum of context-sensitive-fitnesses

where $\lambda_{(\theta,\,p)} \geq 0$ is the weighting of the environmental context *θ*, for entity *p*.

[56] We will dispense with the need for this in Equation 17.

Now, whether a symbiotic relationship is preferred or not depends on the relative weighting of each context to each entity involved, and many factors could influence this. For example, a biased distribution over environmental contexts may be 'inherited' by virtue of the collocation of parents and offspring in a subdivided population (or ecosystem subdivided into species), or affected by the behavioural migration of organisms during their lifetime, or the selective displacement of one species by another in short term population dynamics. We did not wish to introduce such factors and accompanying assumptions into our model. However, neither did we wish to assume an equal weighting of environmental contexts. In fact, it turns out to be critically important that we take account of the context sensitivity of a modular solution in order to maintain diverse complementary modules (as anticipated in 3.4.6).

7.3.2 Pareto dominance

The concept of *Pareto dominance* is specifically designed for application in cases where the relative importance of a number of factors is unknown (e.g. see Fonseca & Flemming 1995 for recent discussion of the use of Pareto dominance in EAs). Put simply, this concept states intuitively that, even when the relative weighting of factors (or 'dimensions') is not known, the overall superiority of one candidate with respect to another can be confirmed in the case that it is non-worse in all dimensions and better in at least one. More exactly, '***x*** Pareto dominates ***y***' is written '***x*** >> ***y***', and:

$$\boldsymbol{x} >> \boldsymbol{y} \Leftrightarrow (\forall\theta : \boldsymbol{csf}(\boldsymbol{x},\theta) \geq \boldsymbol{csf}(\boldsymbol{y},\theta) \text{ AND } \exists\theta : \boldsymbol{csf}(\boldsymbol{x},\theta) > \boldsymbol{csf}(\boldsymbol{y},\theta).$$

or equivalently, given that ***x*** and ***y*** are different in at least one dimension:

$$\boldsymbol{x} >> \boldsymbol{y} \Leftrightarrow \nexists\ \theta : \boldsymbol{csf}(\boldsymbol{y},\theta) > \boldsymbol{csf}(\boldsymbol{x},\theta).$$

In cases where there is some θ such that $\boldsymbol{csf}(\boldsymbol{x},\theta) > \boldsymbol{csf}(\boldsymbol{y},\theta)$ and some other θ such that $\boldsymbol{csf}(\boldsymbol{x},\theta) < \boldsymbol{csf}(\boldsymbol{y},\theta)$, we say that ***x*** and ***y*** are *non-sorted*. And in cases where $\exists\boldsymbol{x}\textbf{:}\ \boldsymbol{x} >> \boldsymbol{y}$ we say that ***y*** is *dominated*, else ***y*** is *non-dominated*. For our ecological domain, these simple rules are easily interpreted. In the case where ***x*** is better in some environments than ***y***, and ***y*** is better in some environments than ***x***, then we do not know which is fitter overall unless we know the relative weighting of the environments for each entity. But, if ***x*** is always fitter (or at least as fit as) ***y*,** then regardless of the weightings of the environments for each entity, we know that the overall fitness of $\boldsymbol{x}$ is greater than that of $\boldsymbol{y}$ (assuming $\boldsymbol{x}$ and $\boldsymbol{y}$ are different in at least one dimension).

This pair-wise comparison of two entities over a number of contexts will be used to determine whether a symbiotic join produces a stable composite. If we write the composition of entities $\boldsymbol{a}$ and $\boldsymbol{b}$ as $\boldsymbol{a}+\boldsymbol{b}$, then, using the notion of Pareto dominance, $\boldsymbol{a}+\boldsymbol{b}$ is stable iff $\boldsymbol{a+b} \gg \boldsymbol{a}$, and $\boldsymbol{a+b} \gg \boldsymbol{b}$. In other words, $\boldsymbol{a+b}$ is unstable if there is any context in which either $\boldsymbol{a}$ or $\boldsymbol{b}$ is fitter than $\boldsymbol{a+b}$.

i.e. $\boldsymbol{stable}(\boldsymbol{a+b}, \boldsymbol{a}, \boldsymbol{b}) \equiv \boldsymbol{a+b} \gg \boldsymbol{a}$ AND $\boldsymbol{a+b} \gg \boldsymbol{b}$,

i.e. $\boldsymbol{unstable}(\boldsymbol{a+b}, \boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow \exists\theta\in \boldsymbol{Contexts}$: $(\boldsymbol{csf}(\boldsymbol{a},\theta) > \boldsymbol{csf}(\boldsymbol{a+b},\theta)$ OR $\boldsymbol{csf}(\boldsymbol{b},\theta)) > \boldsymbol{csf}(\boldsymbol{a+b},\theta))$

where $\boldsymbol{Contexts}$ is a set of complete feature specifications.

We should note that there is a subtle distinction between 'the fitness of an entity *in* an environment' and 'the fitness of the entity *and* environment together' i.e. $\boldsymbol{csf}(\boldsymbol{x},\theta) \neq \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{x},\theta))$. However, our method precludes the need to separate the former from the latter because the pair-wise comparison of two entities in the same environmental context implicitly 'differences away' the contribution of the environment.[57] That is,

$$\boldsymbol{csf}(\boldsymbol{x},\theta) > \boldsymbol{csf}(\boldsymbol{y},\theta) \Leftrightarrow \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{x},\theta)) > \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{y},\theta)),$$

where $\boldsymbol{f}(\boldsymbol{w})$ is the objective fitness of the complete feature set $\boldsymbol{w}$ as given by the fitness function. This assumes that although we can only measure the fitness of a complete feature specification (organism and environment together) we can determine the information we need by differencing away the fitness contributions coming from the environment by including them in both sides of the inequality.

Thus our condition of instability becomes:

$$\boldsymbol{unstable}(\boldsymbol{a+b}, \boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow \exists\theta\in \boldsymbol{Contexts}: (\boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a},\theta)) > \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a+b},\theta)) \text{ OR } \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{b},\theta)) > \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a+b},\theta)))$$

Equation 18: Stability of a+b.

Equation 18 becomes our abstraction for Figure 7-1 (c).

[57] It is plausible that the non-epistatic contributions of an environment are also 'differenced away' in biological scenarios.

In our simulation we will test the stability of a proposed join and immediately ‘undo’ it if it is ‘unstable’ by these criteria. This seems a little like we are explicitly testing whether a variation is good before we make it. However, this is not the intent of the model. Rather, we are assuming that new associations are made randomly, but, if and only if the new entity is reliably fitter than both of the parties involved will the new entity survive in the ecosystem. In other words, if there is some niche where one of the two symbionts will out-compete the composite then the composite is competitively excluded. We abstract the process of random joining and competition into the stability test. This models the assumption that competition between joined and non-joined variants of an entity occurs rapidly such that only reliably successful joins (by these criteria) persist long enough to be involved in a subsequent join. A join that persists through the ‘stability test’ is treated as a new entity that may participate in further joins as the cycle of the model repeats.

7.3.3 Building environmental contexts: Group evaluation

In our model, the environmental contexts, used in determining Pareto dominance and the stability of a proposed composition, will be formed entirely from other members of the ecosystem. The intent here is that the assessment of a new composition involves selecting between being in permanent association with some particular member of the ecosystem or being in transient association with members of the ecosystem. If we were to employ the naive alternative, i.e. selecting between being in permanent association with some particular member of the ecosystem or remaining in entirely random environmental contexts, then it would be likely that many more proposed associations would be preferred. This would result in many sub-optimal associations. Additionally, if entities are evaluated in transient groups of other entities then there is the potential that they may become co-adapted to one-another, and thereby ‘primed’ to make successful permanent joins by composition. Figure 7-5 illustrates how to build a context from a randomly selected set of entities.

```
               a:  --0---1-
               b:  01------
               c:  -0---0--
               d:  ----1-0-
               e:  ------10
               f:  ---0-00-
Resultant context  01001010
```

Figure 7-5: Building a context from other entities.

(an abstraction for Figure 7-1 (b)). In this example, six entities a through f, are needed to complete a fully-specified feature set of eight features. Where specified features conflict, the specifications of the topmost entity take precedence, as in Figure 7-3.

Algebraically, we define a context, using the recursive function ***S****, from an ordered set of n≥2 entities X_1, X_2,… X_n, as follows:

$$\boldsymbol{S}^*(X_1, X_2,\ldots X_n) = \begin{cases} \boldsymbol{S}(X_1, \boldsymbol{S}^*(X_2,\ldots X_n)), & \text{if } n>2, \\ \boldsymbol{S}(X_1, X_2), & \text{otherwise.} \end{cases} \quad .$$

Equation 19: Building a context

where $\boldsymbol{S}(X_1, X_2)$ is the superposition of two entities as per Equation 16 above.

Some contexts may require more or fewer entities to provide a fully-specified feature set. In principle, we may use all entities of the ecosystem, in random order, to build a context—but, after the context is fully-specified, additional entities will have no effect. This allows us to write a context as S*(E), where E is all members of the ecosystem in random order. Implementationally, we may simply add entities until a fully-specified set is obtained.

7.4 The Symbiogenic Evolutionary Adaptation Model (SEAM)

We may now put together the components we have introduced above to provide a complete model. To summarise, the model includes the following features:

- Variable size entities and a variation operator based on composition.
- Building environmental contexts from other co-adapting entities in the ecosystem.

- Testing (in)stability of compositions by testing for Pareto dominance of the composition over the component entities.

Although each of these features is conceptually somewhat involved, the overall simulation model is not that complicated. Figure 7-6 overviews the operation of SEAM.

```
•   Initialise ecosystem, E, to random, single-feature, entities.[1]
•   Repeat until stopping condition:
      -   Remove two entities at random from the ecosystem → $\boldsymbol{a}$ & $\boldsymbol{b}$.
      -   Produce $\boldsymbol{a}+\boldsymbol{b}=\boldsymbol{S}(\boldsymbol{a},\boldsymbol{b})$, using composition (see Equation 16).
      -   If $\textbf{\textit{unstable}}(\boldsymbol{a}+\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{b})$ return $\boldsymbol{a}$ and $\boldsymbol{b}$ to ecosystem, else add $\boldsymbol{a}+\boldsymbol{b}$ to ecosystem.

    where  $\textbf{\textit{unstable}}(\boldsymbol{a}+\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow$
            $\exists\theta\in \textbf{\textit{Contexts}}: (\boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a},\theta)) > \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a}+\boldsymbol{b},\theta))$ OR $\boldsymbol{f}(\boldsymbol{S}(\boldsymbol{b},\theta))> \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a}+\boldsymbol{b},\theta)))$
            where $\textbf{\textit{Contexts}}$ is a random set of contexts each built by composing together
            other members of the current ecosystem, E, using $\boldsymbol{S}^*(\text{E})$
                                                    (see Equation 18 & Equation 19).

[1] Initialisation needs to completely cover the set of single-feature ‘primitives’ so that all values for
    all features are available in the ecosystem.
```

Figure 7-6: Pseudocode for a simple implementation of SEAM.

7.5 Simulation results

In this section we give simulation results of SEAM applied to a 128-bit Shuffled HIFF. Our intent is to illustrate the qualitative difference in the way that composition operates in this landscape as compared to the operation of accretive evolutionary change and compositional change under sexual recombination demonstrated in the previous chapters. Accordingly, we contrast the operation of SEAM with the results of a mutation only algorithm, Random Mutation Hill-Climbing, (RMHC), shown in 5.3.1, and a GA using sexual recombination (see summary of simulations 6.5).

The pseudocode for SEAM is given in Figure 7-6. The parameters we use for these experiments are: number of features, N=128, alphabet of features, S={0,1}, initial population size 256 one-feature entities covering all alleles at all loci[58], maximum number of contexts used for dominance test, t=200, (empirically, on average less than 20 of these are required on average to reveal the instability of a proposed join). The stopping condition is that $3\cdot10^6$ calls to the fitness function have been used. ***f***, the fitness function, is provided by Shuffled HIFF with 128 binary features.

The population size in SEAM is determined by N - in fact, population_size=2N. SEAM uses no mutation. There is no sexual crossover in SEAM, but the encapsulation operator is applied in all matings. Thus there are almost no parameters to the SEAM model and there is only one curve to be plotted. The remaining parameter, the number of trials, (number of contexts sampled), to determine whether a join is stable is set to t=200 in these experiments - this value was found by preliminary experimentation. The effect of a low t-value is that incorrect joins are passed as correct (see 7.7.2, for related analysis), and an unnecessarily high t-value results in wasted evaluations.[59]

7.5.1 Control experiments

Recall that the three main conceptual features of SEAM are: the use of variable size entities and a variation operator based-on composition; testing (in)stability of compositions by testing for Pareto dominance of composition over the component entities; and, building environmental contexts from other co-adapting entities in the ecosystem.

[58] This may be done systematically for practical purposes, but may in principle be done without knowledge of the encoding dimensions by ‘over-generating’ the initial population and then removing duplicates—more specifically, by removing entities that behave the same (produce identical fitness changes) over a sample of random contexts (see Watson & Pollack 2001b).

[59] A couple of implementational details: The implementation used in these experiments checks for A+B>>A using a separate set of contexts from the check for A+B>>B. This is wasteful of evaluations since it evaluates A+B in twice as many contexts as necessary. Also, this implementation does not test whether A+B==A or A+B==B - if such a test is possible, which it is using a binary representation as we do, then this could be used to avoid 200 wasted evaluations every time this occurs. The following results therefore provide an upper limit on the number of evaluations necessary.

Variable-sized entities and composition vs full-sized entities and crossover

RMHC and the GA provide a control experiment for the use of partially-specified individuals. That is, they use a fully-specified feature set for each entity/individual, and use mutation for variation. The GA with sexual recombination provides a control for the use of symbiotic encapsulation. We already know that these methods fail on Shuffled HIFF.

Pareto dominance of composite over components vs better on average

We also tested the operation of an algorithm that is the same as SEAM except that instead of using the Pareto dominance test, the second feature of SEAM, it simply replaces a component (parent) with the composite (offspring) if the average fitness of composite, over the set of equally weighted environmental contexts, is greater than the average fitness of that component. This provides a control experiment for the second aspect of SEAM.

Co-adapted templates (group evaluation) vs random templates

We also tested a control for the third feature of SEAM by using an algorithm that is the same as SEAM except that it uses random feature sets for the contexts instead of contexts built from other members of the ecosystem.

7.5.2 Simulation results for SEAM and control experiments

Performance is measured by the fitness of the best string evaluated (in the preceding 2000 evaluations) averaged over 30 runs for each algorithm. The strings evaluated are the groups of entities (i.e. an entity with its contextual environment), forming a complete feature specification. The problem size of 128 bits gives a maximum fitness of 1024.

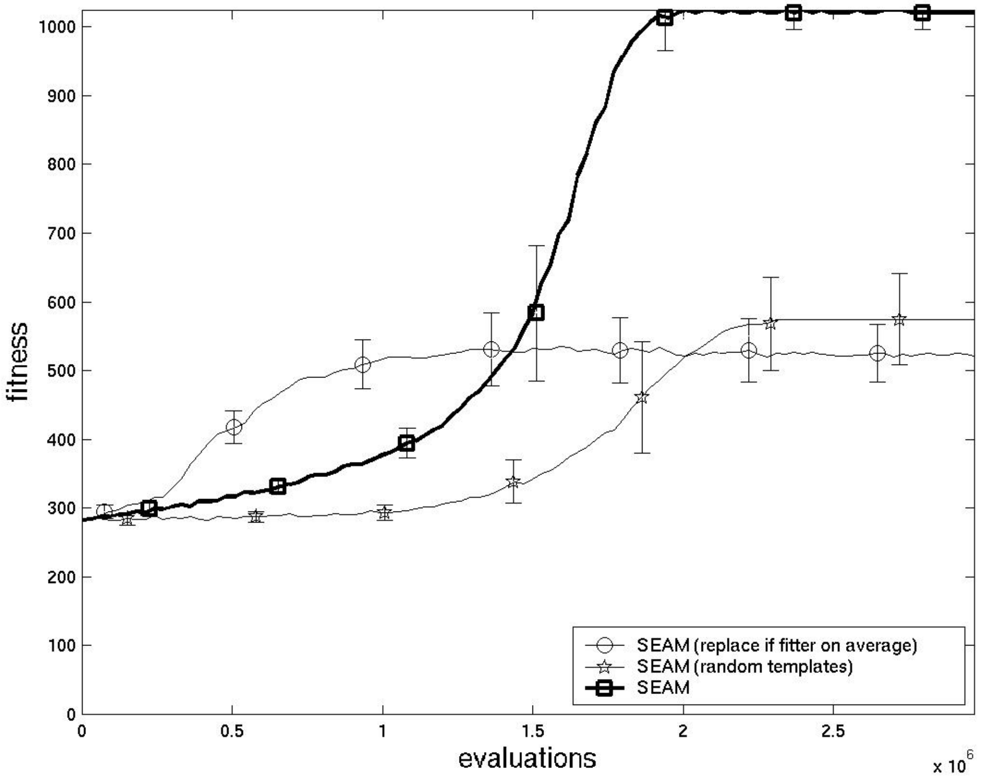


Figure 7-7: Simulation results for SEAM and control experiments.

Figure 7-7 and Figure 7-8 show the performance of SEAM and the two control experiments described above. SEAM finds *both* global optima in *all* 30 runs.

We see that the performance SEAM using replacement when offspring are better on average than the parent is not successful. Performance is much like that of the GA DC with random replaces (Figure 6-7 right). In this experiment, many sub-optimal associations are made and the entities 'fill-up', or 'bloat',[60] with sub-optimal feature values—thus defeating the composition operator (Watson & Pollack 1999c). The 'greedy' selection for whatever configurations appear to be best on average causes the population to quickly converge on bloated sub-optimal generalists, rather than properly exploring optimal specialists that cover

[60] This is a well-known problem in other variable-sized EA representations such as LISP S-expressions in Genetic Programming. We have addressed the use of MOO techniques for reducing bloat in GP in (De Jong et al. 2001) - but we have not yet used the full SEAM method on GP.

the problem together. Although, as we can see in the figure, this is advantageous in the short term, the population is unable to find optimal configurations in the long term.

We also see that the other control using random templates/contexts to test the stability of a join is also unsuccessful. In this experiment, joins between small individuals are not made unless they are correct because random templates are sufficient to distinguish sub-optimal associations. However, as modules become larger, random templates are insufficient to identify inter-module interdependencies correctly and sub-optimal associations result. Once again the individuals fill-up with sub-optimal associations, and the algorithm fails to find high-fitness configurations.

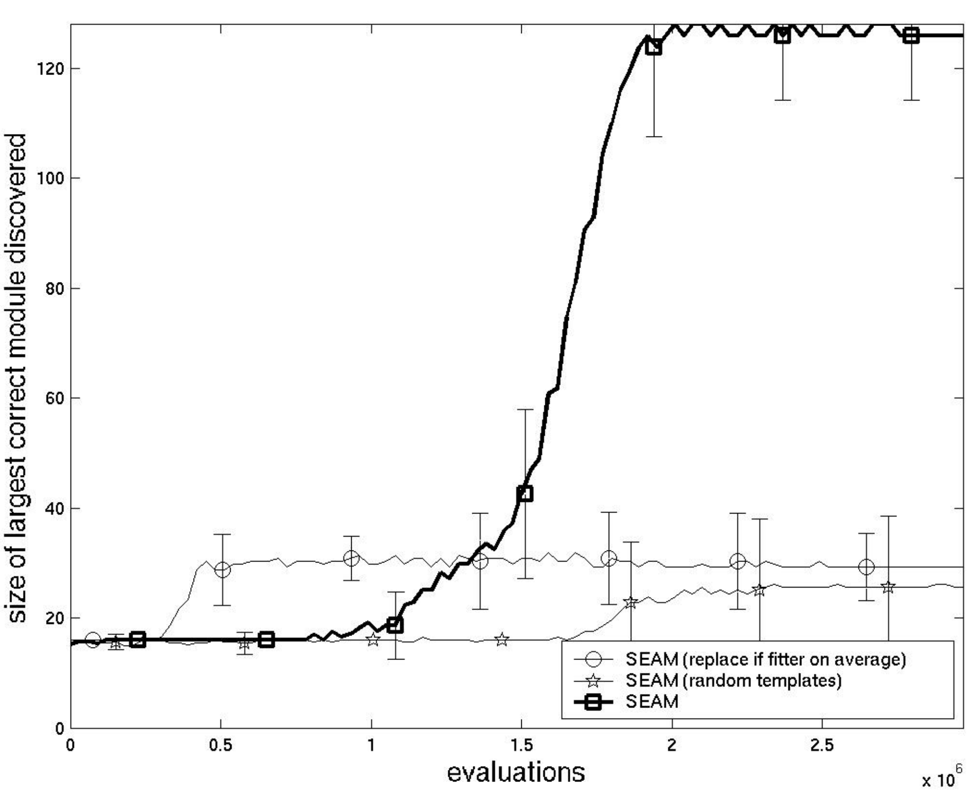


Figure 7-8: Simulation results for SEAM and control experiments: Size of modules discovered.

7.5.3 Summary of main simulation experiments using SEAM and GA.

As Figure 7-9 shows, the results for SEAM are very successful as were the results for sexual recombination with tight linkage. However, SEAM (since it is not able to exploit the assumption of favourable gene ordering) takes about four times longer for this size problem, N=128.[61] But both methods are clearly different from the other algorithms. SEAM and the GA with deterministic crowding (given the assumption of tight linkage) find *both* global optima in at least 28/30 runs. None of the other methods find *either* global optimum in *any* of the 30 runs.

Progress in the non-compositional mechanisms (i.e. mutation only methods and panmictic GA), and scenarios where composition is disrupted (e.g. GA with random linkage), becomes increasingly more

[61] It appears that SEAM takes about twice as long as GA DC to find complete solutions when N=64.

difficult in HIFF as the width of the fitness saddle to the next higher-fitness point becomes further away with each fitness increase. But in scenarios where composition is working effectively, i.e. the GA DC with tight linkage, and in SEAM, progress is not impeded in this landscape.

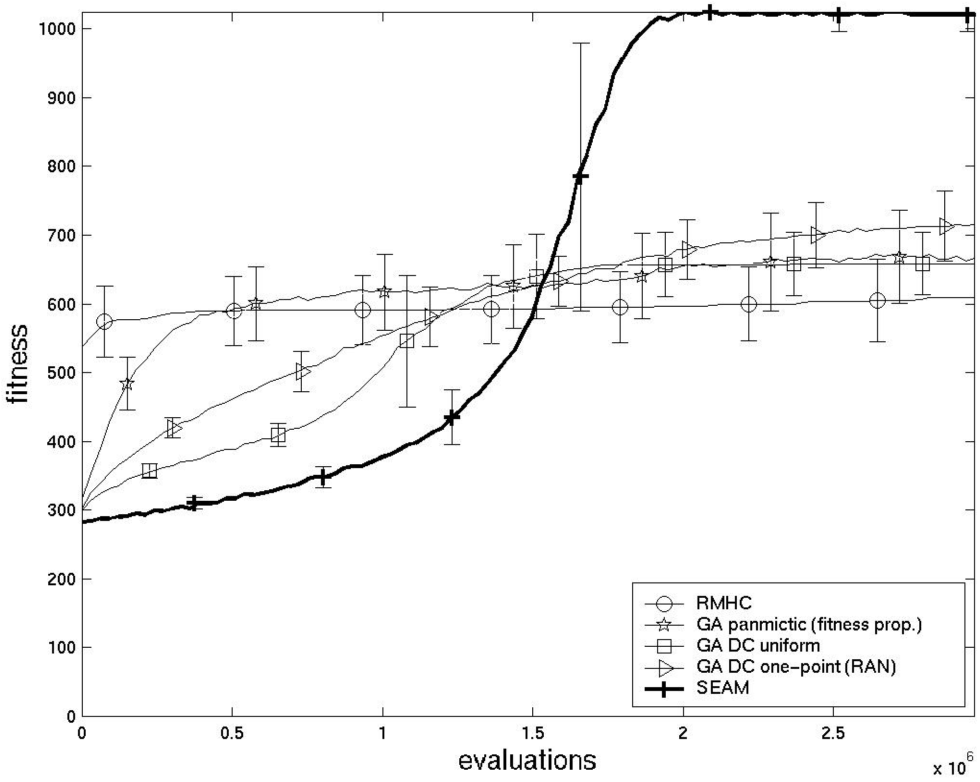


Figure 7-9: Summary of simulation experiments 2 - SEAM and GA.

Simulation results for various algorithms on 128-variable Shuffled HIFF. Best mutation rates (of those sampled earlier) are used for each method that uses mutation. RMHC - Random Mutation Hill-Climber (mut=0.0938), 'GA panmictic' - Genetic Algorithm without any diversity maintenance, using fitness proportionate selection, and one-point crossover (mut=0.0078). 'GA DC uniform' - GA with Deterministic Crowding diversity maintenance method, and no genetic linkage (i.e. uniform crossover) (mut=0). 'GA DC one-point RAN' - GA with DC, and one-point crossover, and random genetic linkage (mut=0.031). 'SEAM' - Symbiogenic Evolutionary Adaptation Model (mut=0).

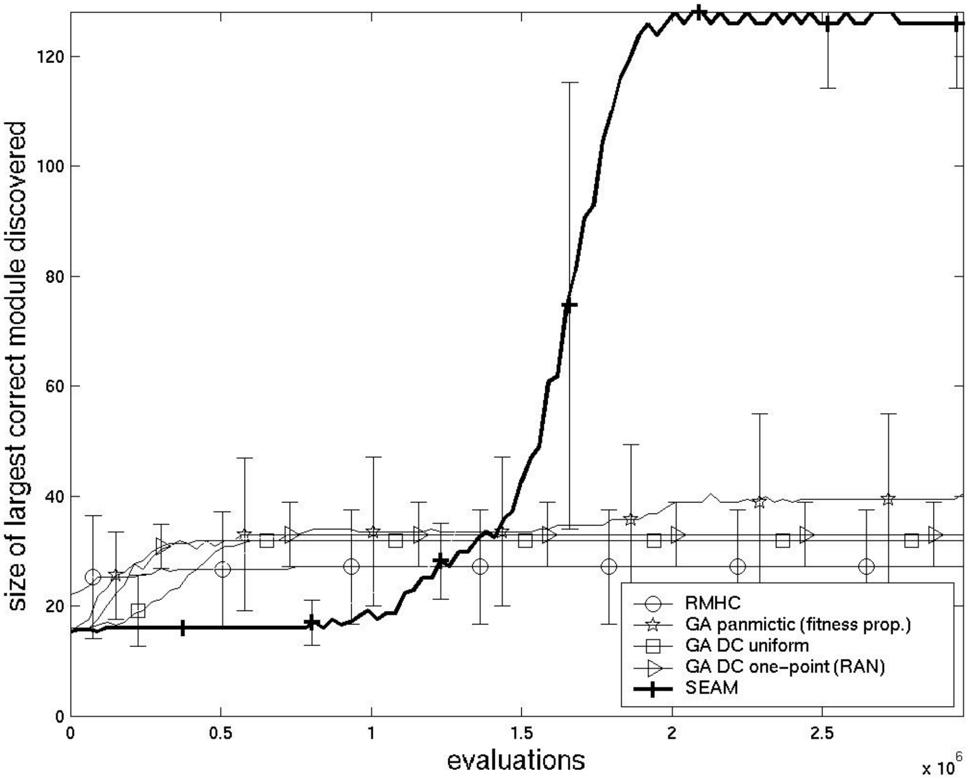


Figure 7-10: Summary of simulation experiments 2: SEAM and GA - size of largest correct module.

Shows size of largest correct module discovered for same algorithms as Figure 7-9.

7.6 Discussion

7.6.1 Coevolution and SEAM

In evolutionary computation terms, SEAM describes an evolutionary algorithm where schemata of all sizes coevolve with one another, as if in a multi-player game, and cooperative groups are found incrementally from individual features through larger and larger schemata. With respect to the biological analogues, SEAM describes an ecosystem of entities that coevolve with one another, finding stable symbiotic relationships that satisfy their fitness dependencies with one another, and progress through successive evolutionary transitions, each occurring via the composition of simpler extant entities into more complex organisations.

7.6.2 Canalisation of successful groups

There is an interesting analogy between SEAM, the Baldwin effect (Baldwin 1896), and ‘Symbiotic Scaffolding’ (Watson & Pollack 1999d, Watson et al. 2000b). That is, these scenarios have in common the feature that rapid non-heritable variation (lifetime learning or the temporary groups formed for contexts) guides a mechanism of relatively slow heritable variation (genetic mutation or composition, respectively). In other words, evaluation of entities in contextual groups ‘primes’ them for subsequent joins, or equivalently, solutions found first by groups are later *canalised* (Waddington 1942) by composite entities (see also Bull 1995). In Figure 7-11, the ‘indiv.’ curve shows how the discovery of correct building blocks by individuals follows behind the discovery of correct building blocks by groups.

We show the size of the largest correct sub-block discovered. The ‘group’ curve for SEAM is the size of the largest correct module in any group of entities when they are evaluated together as a contextual environment, (this corresponds to the curve shown for SEAM in Figure 7-8). The ‘indiv.’ curve for SEAM is the size of the largest correct module in any stable individual entity. We use a log scale on the size axis—thus, if the increase in size is proportional to extant size the curve would appear as a straight line. We can see clearly in this figure that unlike the conventional evolutionary algorithms, innovation by composition continues steadily in this problem, approaching a scale-invariant increase in size of correct modules in individual entities.

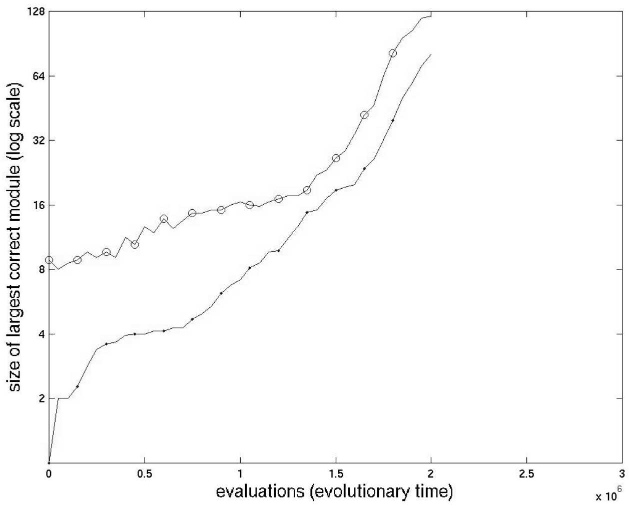


Figure 7-11: Size of modules in individuals and in groups in SEAM. (log scale)

SEAM (group) shown with circle point-markers. SEAM (indiv.) shown with dot point-markers.

7.6.3 Scale-invariant evolutionary processes

The main features of an evolutionary algorithm are: variation, evaluation, and selection. In the RMHC, variation acts at the scale of the primitive problem variables, and evaluation and selection act on complete strings. In the GA with crossover, variation (by crossover) manipulates schemata of all sizes at all stages of search, and evaluation and selection are (at least on the face of it) applied to complete strings. In neither method does the scale of these processes increase progressively as search continues. Thus, even in an adaptive landscape that is scale invariant, like HIFF, it cannot be the case that these mechanisms properly ‘escape’ the primitive variables of the problem domain. Specifically, both of these methods continue to explore variations in the primitive variables throughout the process. For example, even when all individuals contain only large fully-optimised correct modules, crossover is still just as likely to exchange an order-1 schema as it was in the first generation of the run.

In contrast, in SEAM all three of these features scale-up as search progresses (if the problem domain allows):

- **Variation:** Variation is performed by the assembly of extant entities. As extent entities increase in complexity, the variations that are explored are of the same scale. The size of jumps that can be made (with respect to the original mutation-based landscape) scales with extant complexity of entities. This frees the variation operator from the original primitive-space of variations and allows search to move explicitly from the combinations of primitive variables to the combination of schemata of successively higher orders. (Notice that the addition of any explicit mutation breaks the scale-invariant quality of the variation in the algorithm. That is, mutation is inherently tied to the original primitive variables.)

- **Evaluation:** SEAM progresses by the identification of good modules rather than the direct identification of good complete solutions. The identification of good modules requires that we have good assemblies in which to test the value of a module. In SEAM the assemblies that are used to test a module are built from extant modules. Therefore, the quality of contexts used to evaluate modules improves in step with the quality of modules.

- **Selection:** In SEAM, when new entities are defined they encapsulate the pair of entities involved so that they always travel together as a whole in future. Accordingly, the unit of selection is explicitly scaled-up. And since new units are created from the assembly of extant units, the unit of selection are scale-free too.[62]

These observations show that all these major features of SEAM are scale free. When applied to a scale-free modular interdependency problem, such as HIFF, the modularity at all scales is equally utilisable by the algorithm.

[62] A join that is deemed stable is not re-tested for stability in future, thus SEAM explicitly scales-up the unit of selection. However, if we continued to apply selection for composites at lower levels, individuals would re-choose the same associations because the same join was in their selfish interest in the first instance and remains in their selfish interest. Thus we could implement the algorithm using only primitive-unit selection. However, this would miss the point that the selection of the primitives can be accurately abstracted into the selection of the composites in this class of problem.

7.7 Analysis of SEAM on Shuffled HIFF

This section provides an analytic treatment of SEAM, and in so doing helps to explain how SEAM works on HIFF and under what conditions it would fail.

Our analytic treatment of SEAM on HIFF follows that developed for sexual recombination on HIFF given in Section 6.6. Specifically, we calculate the expected time to solution from the product of the number of blocks to be found in a complete solution and the expected time to find each block.

7.7.1 Idealised-SEAM

Let us define a join to be 'correct' when two compatible entities, that are themselves fully-optimised modules, are brought together to create a new higher-level fully-optimised module of twice the size: e.g. "00--"+"--00" → "0000" is a correct join.

We start by proving that SEAM only makes correct joins if the stability of a proposed join is made is tested in all possible contexts. It is not actually the case that all possible contexts are tested in SEAM—a sample of contexts, built from other coevolving individuals, is used to test the stability of a join—but this proof provides an appropriate stepping stone in coming to a more complete analysis of SEAM.

Lemma 1

> Only '*correct*' joins are made in SEAM if all possible contexts are used for testing 'stability', where correct means that the new entity created is a fully-optimised module, and 'all possible contexts' is the set of all strings collectively covering all possible configurations of features.
>
> **proof:** The set of all possible contexts includes the two global optima, i.e. the strings of all-1s and all-0s, which we will call the 'ideal contexts' for reasons which will become clear. An entity, A, entering into a join, is fully-optimised internally, say, all-0s. Thus the fitness the string created by superimposing entity A on one of the ideal contexts will be the maximum fitness possible (since the superposition of 0s on 0s, for example, will have no effect). Clearly, there is no pairing with any other entity, B, which could yield greater

> fitness. Moreover, any pairing with any ‘compatible’ entity, i.e. an entity of the same type (all-0s), will not depreciate the fitness of entity A in this context. That is, f(A+B+ideal_context)=f(A+ideal_context) when A, B, and the ideal context are all all-0s. However, any join of A with some other entity B that is incompatible (of the wrong type, i.e. all-1s, or internally incorrect) will decrease the fitness of A in the ideal context. The same reasoning applies for entities of all-1s. Thus, any compatible join will be deemed stable, and any incompatible join will be deemed unstable. Then, since SEAM is initialised with correct one-variable modules, all modules in the population at all times will be correct. [end].

It is clear from this reasoning that using only the two ‘ideal contexts’ to test the stability of a proposed join is sufficient to ensure than only correct joins are made.[63] Intuitively, any join that *could in some context* be detrimental, *will* be detrimental in one of the ideal contexts. We can easily implement an ‘idealised’ version of SEAM that we can use to illustrate this. The pseudocode is given below - Figure 7-12. Clearly idealised-SEAM is not algorithmically practical because it uses the two globally optimal strings as part of the algorithm. However, it is informative theoretically to consider the operation of idealised-SEAM first. Following this, we will consider the expected time for (the original) SEAM method to cover a *sufficient* number of contexts to reveal an incorrect join.

[63] We can view this as the decomposition of the single-objective HIFF problem into a two-objective problem - specifically, the ‘find modules of zeros’ dimension, and the ‘find modules of ones’ dimension (see Knowles et al. 2001 for an explicit multi-objective treatment of HIFF). These two dimensions can be seen as the ‘ideal trainers’ (Juille & Pollack 1998) for the two underlying dimensions inherent in the HIFF problem (credit goes to Edwin De Jong for this observation).

```
•   Initialise ecosystem, E, to random, single-feature, entities.[1]
•   Repeat until stopping condition:
        -   Remove two entities at random from the ecosystem $\rightarrow \boldsymbol{a}$ & $\boldsymbol{b}$.
        -   Produce $\boldsymbol{a+b}=\boldsymbol{S}(\boldsymbol{a},\boldsymbol{b})$, using composition (see Equation 16).
        -   If $\boldsymbol{unstable}(\boldsymbol{a+b}, \boldsymbol{a}, \boldsymbol{b})$ return $\boldsymbol{a}$ and $\boldsymbol{b}$ to ecosystem, else add $\boldsymbol{a+b}$ to ecosystem.

    where $\boldsymbol{unstable}(\boldsymbol{a+b}, \boldsymbol{a}, \boldsymbol{b}) \Leftrightarrow$
            $\exists \theta \in \boldsymbol{Contexts}$: $(\boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a},\theta)) > \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a+b},\theta))$ OR $\boldsymbol{f}(\boldsymbol{S}(\boldsymbol{b},\theta)) > \boldsymbol{f}(\boldsymbol{S}(\boldsymbol{a+b},\theta)))$
            where Contexts is a set of exactly two strings, all-0s and all-1s.
                                                                        (see Equation 18).
```

[1] Initialisation needs to completely cover the set of single-feature 'atoms' so that all values for all features are available in the ecosystem.

Figure 7-12: Pseudocode for 'idealised-SEAM' i.e. SEAM using 'ideal contexts'.

This algorithm is the same as that in Figure 7-6 except that Contexts is the set of two globally optimal strings, rather than a sample set of contexts built from other coevolving entities.

Note also that any join that 'overwrites' an entity at some loci with allele values that disagree will not be allowed. If an entity A is joined with an entity B, and B is not fully compatible with A, then B will depreciate the fitness of A in one or other of the ideal contexts and the join will not be allowed. Thus it is never possible for a join to make a fully-optimised block become sub-optimal when ideal contexts are available. This, together with the knowledge that the initial population consists only of correct blocks, means that the population always consists of correct blocks only, and the number of correct blocks in the best individual in the population can never decrease.

Note that, since the only variation operator in SEAM is encapsulation, and in idealised-SEAM an entity is never replaced by another entity unless it forms a correct join, this ensures that there can only be a monotonic increase in number of correct modules in the population. Moreover, it is clear that since all primitive modules are available by initialisation and all higher-level modules can be formed by some assembly of them, there is always a means to increase the number of correct modules given an appropriate selection of existing entities.

We now calculate an upper bound on the expected time to solution for idealised-SEAM on HIFF using similar assumptions as we did in Section 6.6. Specifically, given the use of idealised contexts and the reasoning above, we know that at all times the entities in the population provide the necessary components for blocks at the next level in the hierarchy. This means that there is necessarily a change available in the population (a composition) that will increase the number of correct modules in the individual that has the greatest number of correct modules - up until the point that this individual is one of the globally optimal strings. (This is analogous to proving Condition 1 in Section 6.6.2.)

In Section 6.6.2 we used $T \leq BS$, where B is the number of blocks to be found (i.e. steps on the path to the optimum) and S is the expected time to find a block (i.e. time to take a step). In idealised SEAM we know that there is at least one encapsulation operation that will find a new module given some choice of donor entities from the current population. Unlike the conditions for sexual recombination we do not need to factor in the number of possible crossover masks, since given the correct donor individuals there is only one way to compose them together[64] and this will necessarily produce the desired module as a result. So we could simply use S=3CP, and accordingly, $T \leq 3CBP$, where P, is the number of choices of parents/donor individuals, and C=2 for idealised-SEAM is the number of contexts that need to be tested for each choice of parents (corresponding to the two ideal contexts in idealised-SEAM), the factor of 3 comes from the need to do 3 evaluations (A in c, B in c, and A+B in c), for each context c in C. However, as before, we can improve on this estimate by noticing that there is generally more than one encapsulation operation that will find a new module - at a given level (other than the top level) of the hierarchy there are many blocks to be found and many opportunities to increase the number of correct blocks (make a correct join). For example, at the beginning of search there are many size-two blocks to be found and many possible encapsulations that may find a size-two module given the initial population.

So, in a manner analogous to that in Section 6.6.3, we can use an estimate of time to progress along the path to the optimum that takes account of the fact that the expected time for a step changes with each step,

[64] correct donor entities are not overlapping - so, S(A,B)=S(B,A).

as the number of possible ways to find a block changes. The expected time to find one of *q* available blocks is *P/q*, thus;

$$T \leq \sum_{b=1}^{B} \frac{3CP_b}{q_b}$$

Equation 20: Expected time to solution for SEAM

> where *B* is the maximum number of steps in the path to the optimum, P_b is the number of choices of parents at the bth step, *C*=2 is the number of contexts that need to be tested for each choice of parents, and q_b is the number of ways that a new block can be swapped-in at the b^{th} step.

Thus, we may write $T \leq 3Cr$, where *r* is the sum of *P/q* for all steps in the path to the optimum. Theorem 7-1 uses an upper bound on *r* to give a time to solution for idealised-SEAM on H-IFF.

Theorem 7-1

T=$\boldsymbol{O(N^2 lnN)}$, where T is an upper bound on the expected time to find a global optimum in H-IFF using the idealised-SEAM algorithm in Figure 7-12, and N is the problem size in bits.

Proof: To use $T \leq 3Cr$ we must find $r=r_1+r_2\ldots+r_B$ where $r_b=P_b/q_b$, where q_b is the number of blocks that may be discovered at the b^{th} step, and P_b is the number of choices of parents(/donor entities) at the b^{th} step. We will count a step for each correct join the algorithm makes, which equals twice the total number of blocks in either globally optimal solution - i.e. the total number of steps, B=2(N-1). At the first hierarchical level in H-IFF there are *N*/2 size-2 blocks of each type (0 and 1) to be discovered (in the worst case, we may assume that the initial strings have none of their size-2 blocks correct). By the reasoning of Lemma 1, any one of these blocks may be discovered by encapsulation of a particular pair of existing entities. So, q_1=2N (because the order of selected parents does not matter so there are two ways to create each block). In the initial population in idealised-SEAM there are 2*N* entities and thus $P_1=4N^2$ (assuming we do not bother to check that donor entities are different). Thus $r_1=4N^2/2N=2N$. There are now *N*-1 blocks remaining to be found at the first level, and the since the encapsulation of two entities into one reduces the population size by one $P_2=(2N-1)^2$. So $r_2=(2N-1)^2\ /2(N-1)$. More generally, let the hierarchical levels in the problem be indexed with j from lgN to 1, and let the blocks that need to be discovered in that level be indexed with k from 2^j to 1, then for the kth block at the jth level $p=(2^j+k)^2$ and $q=2k$. The overall sum of p/q for all blocks at all levels, r, is

$$r = \sum_{j=1}^{lg\ N} \sum_{k=1}^{2^j} \left(\frac{\left(2^j + k\right)^2}{2k} \right)$$

$$= \sum_{j=1}^{lg\ N} \sum_{k=1}^{2^j} \left(\frac{2^{2j}}{2k} + 2^j + \frac{k}{2} \right)$$

$$\leq \sum_{j=1}^{lg\ N} \left(2^{2j-1}\left(ln\ 2^j + 1\right) \quad + 2^{2j} \quad + 2^{j-2}\left(2^j + 1\right)\right)$$

$$\boldsymbol{r} \leq \boldsymbol{O}\left(2^{2\,lg\ N-1}\ ln\ 2^{lg\ N}\ \right)$$

$$\boldsymbol{r} \leq \boldsymbol{O}\left(N^2\ ln\ N\right)$$

For the algorithm in Figure 7-12, C=2. So, from $T \leq 3Cr$,

$T \leq 6r$,

Thus $T = \boldsymbol{O}(N^2 \ln N)$

[end]

7.7.2 From idealised-SEAM to SEAM

Idealised-SEAM uses exactly the two ideal contexts for testing the stability of a join and is thereby able to ensure that only correct joins are made. In (ordinary, not-idealised) SEAM, a random sample of contexts is used where each context is built from an assembly of other entities in the population.

To expand our proof for idealised-SEAM to SEAM we need to estimate the number of contexts required to correctly identify the correctness of a proposed join. Thus SEAM will only make join stable if they are correct, as did idealised-SEAM, but SEAM will require more evaluations to determine the correctness of joins. The basic observation is that although idealised-SEAM uses the global optima as contexts, it is not necessary that the contexts be globally optimal - only that they contain optimal sub-modules of about the same order as the individuals being tested, and since the contexts are built from other individuals of about the same order as the individuals being tested this condition is feasible for any size module as search progresses.

Lemma 2 provides the number of contexts we expect to need to test to determine the correctness or incorrectness of a proposed join. In SEAM a proposed join is tested in a number of contexts until the first occurrence of a context that reveals it to be incorrect. But when a join is correct, no such context occurs - in this case we must place a limit on the number of contexts we will test before being satisfied that the join is correct (i.e. ***t***). This limit should be based on the allowable risk of accidentally accepting an incorrect join as stable.

Lemma 2

C= $\frac{G \ln P}{\ln(1-p)}+\frac{(1-G)}{p}$ where C is the expected number of contexts required to determine the stability a proposed join, G is the probability that the proposed join is a correct join, P is the acceptable risk (allowable probability) that an incorrect join is passed as stable/correct, and p is the probability that a single context will reveal an incorrect join to be unstable/incorrect.

Proof. C=G***a***+(1-G)***b*** where ***a*** is the expected number of contexts required to determine that a correct join is correct, and ***b*** is the expected number of contexts required to determine that an incorrect join is incorrect. If the probability that a single context will reveal an incorrect join is p, then the expected number of contexts required to reveal an incorrect join is, ***b***=1/p. The probability that an incorrect join will not be revealed to be incorrect in a single context is 1-p. The probability that an incorrect join will not be revealed to be incorrect in k contexts is $(1\text{-}p)^k$. We want this probability to be P, thus P=$(1\text{-}p)^k$. So, k=$\log_{(1\text{-}p)}$P, thus k= $\frac{\ln P}{\ln(1-p)}$, where k=***a*** is the number of trials required to make the probability of an incorrect join passing as correct to be P. Thus C= $\frac{G \ln P}{\ln(1-p)}+\frac{(1-G)}{p}$.

Our intent in this analysis is to show that, with reasonable assumptions, there is a constant upper bound on C with respect to N, the size of the problem. This, in turn, will allow us to show a polynomial upper bound on the SEAM algorithm. Accordingly we need to bound G, P, and p appropriately. We address each term individually.

G, the probability of a proposed join being correct, changes as the algorithm progresses. In general, G is the ratio of 'the number of new blocks that could be created from a pair of existing blocks' to 'the number of possible pairs of existing blocks'. This ratio, and therefore G, cannot be greater than one. P, the acceptable risk (allowable probability) that an incorrect join is passed as stable/correct should be low. Roughly, we want to make O(N) joins without making any incorrect joins. In any case, ln(P), where P is a probability<1, will have a maximum which is negative.

Thus, given these bounds on G and P, if we can show that p, the probability that a single context will reveal an incorrect join to be incorrect, is constant, then we can show that C has a constant upper bound Showing p to be constant is more complicated.

The probability that a single context will reveal an incorrect join to be incorrect

In SEAM, a join between two individuals A and B is incorrect if f(A+c)>f(A+B+c), or f(B+c)>f(A+B+c) for any c tested, where f is the fitness function and c is a context. To start with, let us consider only f(A+c)>f(A+B+c). It is easily seen that the join will be rejected if there is any context tested where the presence of B causes a decrease in the fitness of the string.[65]

Note that any alleles provided by B that are overlapped by A can play no part in these fitness evaluations. Thus any context that might possibly show f(A+c)>f(A+B+c) would also show f(A+c)>f(A+B'+c) where B' is the subpart of B that is not overlapping with A. Let D be the subpart of context c that is exactly covered by B' – i.e. D is located in the same partition of the variables as B'. Let us also consider E, which is defined as the partition of variables that form the complementary part of B' (and therefore also form the complementary subpart of D), making a module at the next hierarchical level in the problem structure – i.e. in HIFF this will be the block neighbouring B' defined in the hierarchical block structure, in Shuffled HIFF, this will be the new randomised position of these same variables. Thus if B' and E are both internally

[65] Note that when A and B form a correct join there cannot be any context in which the superposition B causes a decrease in fitness that has not already been caused by the presence of A.

correct and compatible (i.e. both 0s blocks, or both 1s blocks) then B’+E forms a correct join. Similarly, since D is in the same partition as B’, D+E may or may not form a correct join.

As stated, we are interested in the probability of finding a context where the presence of B causes a decrease in fitness. In HIFF, a change in fitness resulting from the superposition of a module on a context can occur either because of the independent fitness contribution of that module (if it is a correct block or contains correct blocks), or from the fitness dependencies between that module and other modules in the context. If B is a correct block then its independent fitness contribution cannot be lower than that of D. However, if B is incompatible with E, but D is compatible with E, then B will cause a fitness decrease. In other words, if D+E is a correct block, and B’ $\neq$ D, then f(A+c)>f(A+B+c). Thus the probability of revealing an incorrect join to be incorrect from a given context, c, is $p \geq q$ where q is the probability that c contains a correct block twice the size of B’ and located at the partition covered by B’ and its ‘neighbouring’ block, E. In this case, D+E makes a correct block, and the superposition of B ‘breaks’ this block causing a fitness decrease. (Note that when A and B form a correct join, E is the same partition as A and thus D+E can only be a correct block when D=B’, thus D+E does not confer greater fitness than D+B’ and there can be no context where f(A+c)>f(A+B+c).)

Note that when using random contexts, p is inversely related to the exponent of the size of the module B. But in SEAM contexts are built from other coevolving individuals. If all individuals in the population, including B, are correct modules of size k, and if the 0s kind of module and the 1s kind of module is equally represented in each partition, then the probability that a context built from this population will reveal an incorrect join to be incorrect is just the probability that the context will contain a correct module of size 2k in the B’+E partition that is incompatible with B’. This probability is independent of blocks in other partitions since the partitions of correct modules are non-overlapping. Thus p=1/4 in this case: i.e. there are four possible block combinations of size k that make candidate blocks of size 2k, only one of which is a correct size-2k blocks and incompatible with B’.

Since the test is symmetric, i.e. we are looking for a context where f(A+c)>f(A+B+c), or f(B+c)>f(A+B+c), the number of contexts that need to be tested is limited by the size of whichever module is the smaller out of A or B. In general, the probability of a context assembled from other individuals revealing an incorrect join to be incorrect is $p=1/2^{(2K/k)}$, where K is the size of the smaller of the two blocks in the proposed join, and k is the average size of blocks in the population in the partition of that block (assuming 0s and 1s are equally represented).

In practice, we cannot ensure that all members of the population are of about the same size because it may be the case that different partitions and different types of block within a partition are progressing at different rates. Let us assume that in general, different modules/blocks in the problem may be m hierarchical levels apart in size, i.e. when the smallest block of A and B is of size K, the average size of other entities specifying blocks in the partition of that block and its neighbouring partition is no less than $k=K/2^m$. This assumption asserts that different solutions to a module never get more than m hierarchical levels ahead of one another.

If some members of the population are of size $k=K/2^m$, then $2K/k=2K/(K/2^m)=2^{(m+1)}$, and the probability of a context revealing an incorrect join to be incorrect is, $p=2^{-2\wedge(m+1)}$, for a join between modules of any size in the operation of SEAM.[66] Thus, p is constant if the ratio of sizes of different solutions to a module in the population is constant. This is not an unreasonable assumption in HIFF and Shuffled HIFF since the discovery of different solutions to a module are equally likely. Thus, given a constant upper bound for p, we arrive at a constant upper bound for C, the expected number of contexts required to determine the stability a proposed join. Finally, a bound for C, allows us to give an upper bound on the expected time to find a global optimum in H-IFF using SEAM.

Theorem 7-2

$T \leq \boldsymbol{O}(\boldsymbol{N}^2 \boldsymbol{lnN})$, where T is an upper bound on the expected time to find a global optimum in Shuffled H-IFF using the SEAM algorithm in Figure 7-12 given a constant bound on the ratio of different sized entities in the population as described above, and N is the problem size in bits.

Proof: Using $\boldsymbol{O}(\boldsymbol{N}^2 \boldsymbol{lnN})$ from Theorem 7-1, which was based on $T\leq 3Cr$, we see from the above argument that in SEAM (as well as idealised-SEAM) $C\approx\boldsymbol{O}(1)$, and r is the same for SEAM as idealised-SEAM, thus $T\approx\boldsymbol{O}(N^2 \ln N)$. [end].

[66] Note that this contextual testing scales-up appropriately as the size of the modules being tested scales up - but testing joins in random contexts cannot do this, as seen in our control experiments.

7.7.3 Summary of analytic results

The gist of the above analysis is that by testing an appropriate number of contextual groups we can ensure that only correct joins are made. This number of contexts is limited, regardless of the size of the modules being processed, simply because contexts are built from individuals of about the same order as the individuals being joined. Thus the operation of SEAM continues unhindered through successive hierarchical levels. Given this invariant property of the SEAM population, the expected time to solution in HIFF or Shuffled HIFF is therefore, polynomial in N, the size of the problem, despite there being difficult interdependencies between modules of size N/2.

Note that in general, analytical results for complex population-based algorithms such as those we have analysed are very difficult because the way in which new individuals are created is dependent on the current state of the population. However, this dependency on the state of the population cannot be ignored since it is exactly this that allows compositional algorithms to change the way in which they move in the search space as evolution progresses. However, both here and in the proof for the GA on HIFF (see 6.6.4), we have been able to provide an analytical time by making assumptions about certain properties of the population rather than the exact composition of the population. In the GA we needed to assume that appropriate diversity was maintained and given this invariant property of the population we can prove a polynomial solution time. In SEAM, the necessary invariant property of the population is less vague - we need only assume that the discovery of different solutions to a module do not get too far ahead of one another. If this holds, then a polynomial bound holds.

Of course this result is dependent on the problem class, HIFF. The property of HIFF that is critical here is that the dependencies of a module with its context are limited, and thereby it is not necessary to test a proposed module in all possible contexts but merely a sample of contexts. Nonetheless, the interdependencies that do exist between a module and its context are high-order and must be resolved to find optimal solutions.

The important point to note is that the result given above for SEAM is true for Shuffled HIFF as well as HIFF and thus the algorithmic advantage available from compositional mechanisms in this class of modular interdependency systems is *not* dependent on knowing in advance which genes are dependent on which other genes so as to order them appropriately on the chromosome.

The important point to note is that the result given in the previous chapter for composition based on sexual recombination showed the potential for compositional mechanisms to provide a divide and conquer style problem decomposition in this class of modular interdependency systems, but that result was conditioned on having *a priori* knowledge of which genes are dependent on which other genes so as to order them appropriately on the chromosome. The result above for SEAM is true for Shuffled HIFF as well as HIFF and thus shows that the algorithmic advantage available from compositional mechanisms is *not*, in general, dependent on knowing this information in advance.

7.8 Summary

In the previous chapter we saw that under some circumstances sexual recombination is able to provide compositional change appropriate for adaptation on landscapes built from modular interdependency like HIFF. However, we also saw that one of the provisos for this was that the ordering of genes on the chromosome must correspond well with the epistatic dependency structure of the adaptive domain.

Results of the previous chapters showed that mutation and sexual recombination are unable to exploit the decomposable structure of HIFF or otherwise overcome the large-scale fitness saddles in the landscape except when tight linkage is assumed for sexual recombination. The provisos for the successful action of sexual recombination were: Appropriate modular interdependency structure, population diversity, strong genetic linkage, and favourable gene ordering on the chromosome (tight linkage).

In contrast, the variable-sized entities in SEAM are able to each identify and represent a correct assembly of compatible features forming a useful module for the next hierarchical level, regardless of gene ordering. Also the use of Pareto dominance in the selection scheme provides an appropriate tool for maintaining diversity that does not depend in any way on genotypic similarity measures. Thus only the first proviso remains for the successful operation of SEAM: Appropriate modular interdependency structure.

Since SEAM has no gene-position-sensitive features, there is no difference in the operation of SEAM on HIFF or Shuffled HIFF. Other variants of HIFF, such as Biased HIFF (where 0s and 1s confer different fitness contributions), and H-XOR (based on exclusive-OR rather than IFF), (see Watson & Pollack 1999a, 1999b), are also solved by SEAM.

These results using symbiotic encapsulation provide a second set of sufficient conditions under which complex systems of this kind are easily evolvable under compositional mechanisms. In particular, SEAM shows that a mechanism from the other end of the spectrum of compositional mechanisms given in Figure 2-5 is not dependent gene ordering in the way we saw for sexual recombination, or on any *a priori* knowledge of which genes are dependent on which others.

Chapter 8 - Implications for Evolutionary Biology

In this chapter we discuss the impact of the previous chapters with respect to our understanding of biological evolution, and discuss ongoing and future research on related issues.

8.1 Impact for evolutionary biology

This subsection discusses the impact of the models in the previous chapters with respect to our understanding of EB concepts. We discuss how our common assumptions and intuitions about evolutionary processes and evolutionary difficulty are affected.

8.1.1 Evolutionary difficulty and gradualism

We have argued that common concepts associated with evolutionary difficulty such as the ruggedness of a landscape, the existence of broad fitness saddles, the number of local optima, and notions of irreducible complexity are all based on an underlying assumption of accretive mechanisms. The model landscape and compositional mechanisms that we have discussed, illustrated, and analysed, show that accretive change is not the only non-teleological possibility for evolutionary mechanisms. Compositional mechanisms can, under certain conditions, enable efficient evolution in landscapes that exhibit all these characteristics.

In this manner we have shown a class of systems, normally considered unevolvable, that are easily evolvable under compositional mechanisms. One conclusion from this, showing general contradiction with the above notions of evolutionary difficulty, is that certain kinds of complex adaptations may be easily evolvable despite the fact that there is no path of small changes conferring monotonically increasing fitness approaching these systems. Our experiments have included scenarios where gradualism is neither sufficient nor essential to explain the evolution of a complex adaptation. The insistence on a path of small changes approaching a complex adaptation is therefore not essential in evolutionary explanation.

Note that although, in general, it is clear that each kind of adaptive mechanism is well-suited to different classes of adaptive landscapes (Wolpert & Macready 1997), we have done more than simply show that accretive and compositional mechanisms are different. For example, we have learned a lot about the class of landscapes that discriminate the adaptive capacities of these mechanisms and why the different mechanisms behave differently in this class, about the properties of complex systems that create such landscapes, and about the conditions under which the adaptive advantage of compositional mechanisms is shown.

8.1.2 Symbiosis as source of evolutionary innovation

The intuition that symbiotic composition might afford large, relatively sudden, changes in the characters of an entity is not new (e.g. Margulis 1993a, Margulis & Fester 1991, Mehrezkovsky 1909, Wallin 1927). And many authors accept that symbiosis is a mechanism available for increasing complexity (e.g. Maynard Smith and Szathmary 1995, Ridley 1985). However, as we have discussed (2.2.5), there has been an underlying assumption that gradual mechanisms are, in some sense, ‘primary’ - in that they provide the components that are later assembled. However, our models illustrate that this adherence to the primacy of gradual change is not required. First, the fact should not be overlooked that although the components may have been evolved gradually, the entity that they were assembled into, was not evolved gradually, and we have shown that in principle there are conditions, where complex adaptations could not have been evolved gradually. Moreover, it is not necessary that the components that are assembled were provided by gradual change; they may themselves have been assembled from sub-components. This kind of ‘symbiosis all the way down’ (or perhaps ‘up’) is reminiscent of Serial Endosymbiosis Theory (Margulis 1993a). Using the notion of hierarchical encapsulation, we have provided a clear model of a class of systems that typify those that cannot be evolved accretively but can be evolved compositionally, and we have operationalised the intuitions of compositional change.

In accord with the intuitions of Margulis, Wallin and the earlier Russian Biologists, our models support the possibility that symbiosis may be a fundamentally different source of evolutionary innovation from the accepted norms of evolutionary change. We reinforce the conclusion that symbiotic encapsulation *does* present a challenge to Darwinian *gradualism* - though it does not present a challenge to Darwinism in general, since it is perfectly reasonable for natural selection to act on large changes (Maynard Smith 1991).

Moreover, it is therefore not true that the Darwinian "must" reject a system of large changes (Ridley, notwithstanding) and it is not the case that all systems must have evolved *only* through the gradual accumulation of small changes. Finally, it is not essential that there be paths of small changes approaching a complex adaptation (Behe, notwithstanding), nor that we find or posit the existence of such paths (e.g. Dawkins 1976 notwithstanding).

8.1.3 Multiple-inheritance allows large non-random genetic changes

Compositional mechanisms afford adaptive capacity that is not available under accretive mechanisms because they employ multiple inheritance. That is, under accretive change, characters are inherited from one parent only (or very similar parents), and under compositional mechanisms, characters are inherited from more than one parent (two, in all the models we have looked at). In addition to stochastic mutation, there are several possible genetic mechanisms that may produce highly non-random genetic variations from a single parent, such as inversion, translocation, or gene duplication. However, when non-random genetic material is introduced across lineages it may be wholly unrelated to the genetic material in the recipient lineage, not a rearrangement of existing genetic material. This may be good or bad - it may be bad because new genetic material, selected for in a different genetic context, may be wholly useless or detrimental in the genetic context of the recipient lineage. On the other hand, it provides an opportunity for innovation that is not available through mutational change or the rearrangement of genetic material within a lineage.[67]

The important point is that the opportunity for innovation through the introduction of genetic material that has been pre-adapted in a parallel lineage is likely to be higher than it is through any system of stochastic variation within the lineage if there is limited context dependency for the genetic material involved. In

[67] As an aside: this dichotomy between a decrease in the probable fitness of resulting organisms and an increase in opportunity for innovation, highlights the distinction between mechanisms that are beneficial to the interests of the individual, and mechanisms that are beneficial for the adaptation of the species. This in turn begs the question of how a mechanism that might be beneficial to the species might arise if it is detrimental to the individual. However, such questions are outside the scope of this thesis. Thus far, we know that sexual recombination and symbiotic encapsulation mechanisms are present in natural systems, and in this thesis we have addressed the adaptive consequences afforded by different adaptive mechanisms when present.

cases where epistatic dependencies do not restrict the fitness contribution of a sequence to a unique context, that sequence may confer positive fitness contribution in some other context, and more to the point, may be more likely to confer a positive fitness contribution than a random change in genetic material.

8.1.4 Assumptions concerning genetic systems

Our models suggest that the following characteristics of genetic systems may benefit from careful consideration:

- **Consideration of non-additive effects**: When mechanisms of genetic variation only provide small genetic changes it cannot be the case that selection acts on anything other than additive effects of alleles (or small combinations of alleles). However, when mechanisms of genetic variation afford large genetic changes (such as sexual recombination of tightly linked genes, or symbiotic encapsulation) selection may well act on non-additive effects of allele combinations. Accordingly, multi-locus non-linear epistatic dependencies should be included in our considerations.
- **Intra-genomic dependencies and inter-genomic dependencies**: When mechanisms of genetic variation act only among the individuals of a single lineage it is reasonable to treat the genetic makeup of other lineages as merely providing a selective context for the lineage of interest. However, when genetic mechanisms permit the transference of genetic material among lineages we must consider the genetic dependencies among lineages not just within them. This requires that we address intra-genomic and inter-genomic dependencies in a unified fashion.

8.2 Related results

8.2.1 Hierarchical encapsulation, the Baldwin effect, and symbiotic scaffolding

We mentioned earlier (7.6) that there is an interesting analogy between the operation of SEAM and the Baldwin effect (Baldwin 1896). Let us elaborate. In the Baldwin effect, lifetime plasticity guides genetic variation, and genetic variation canalises acquired traits. In SEAM, the evaluation of transient groups guides the genetic encapsulation of new entities, and the genetic encapsulation of new entities canalises features of the more successful groups. In both cases, there is a mechanism of rapid non-heritable variation guiding a relatively slow, heritable variation mechanism.

We know that one of the conditions for the Baldwin effect is that the space of variations available to these two mechanisms are in some sense correlated (Mayley 1996). If they are not, then the fact that the mechanism of non-heritable variation might discover a fit configuration does not provide a gradient that the mechanism of heritable variation can follow. It needs to be the case that the slow variation mechanism can incrementally reduce the amount of rapid variation that is required in the lifetime of an organism (and thereby reduce the cost of lifetime learning) - and if the space of variations in the two mechanisms is not correlated, then it is not possible for the heritable variation to canalise the non-heritable variation.

In Hinton and Nowlan's model (1987) of the Baldwin effect, lifetime variation is modelled by random completion of the partially specified string, and genetic variation is modelled by random variation in the specified features. Both mechanisms produce variants that are neighbours in the mutation neighbourhood metric. Accordingly, genetic variation is able to canalise acquired characteristics effectively.

In a different model in our previous work, we began investigating the possibility of 'symbiotic scaffolding' (Watson & Pollack 1999d), (see also Wiles et al. 2001, which uses HIFF and symbiotic scaffolding), a symbiotic analogue of the Baldwin effect. The idea is that, instead of augmenting innate abilities with lifetime learning, symbiotic scaffolding might augment the innate abilities of one organism with lifetime interaction with other organisms in its vicinity. Under symbiotic scaffolding an organism may come to exhibit the characteristics of a symbiont without direct genetic transfer in a manner analogous to the Baldwin effect where an organism may come to exhibit acquired characteristics without Lamarckian inheritance. The basic model demonstrated a symbiotic scaffolding effect, and subsequent investigations illustrated some interesting phenomenon (Watson et al. 2000b). This model illustrates that it is in principle possible for a complex adaptation to be discovered first by a symbiotic group, and subsequently be canalised by the heritable features of a member of that group. The bivalve Solemya reidi may be a natural example (Watson et al. 2000b, Powell & Somero 1986).

However, this work did not address the need for the rapid and slow variation mechanisms to move in correlated spaces. Fortunately, in the 'needle in the haystack' adaptive landscape that we used, following

Hinton and Nowlan, there are no local optima, and the effect was robust. But, in general, it should not be expected that the variations in feature combinations that can be produced by composing together symbiotic groups can be canalised by accretive change within the entity of interest.

In SEAM, this limitation is alleviated. The mechanism of rapid variation is the assembly of transient groups, and the mechanism of slow heritable variation is the pairwise assembly of some of those same entities. Accordingly, the two spaces of variations are appropriately correlated and the canalisation of successful groups is robust. The difference then between the Baldwin effect and the symbiotic scaffolding implicit in SEAM is that whereas the former canalises accretive variation with heritable accretive changes, the latter canalises compositional variation with heritable compositional changes. That is the Baldwin effect operates in the space of accretive variations, and SEAM operates in the space of compositional variations.

More than providing an interesting analogy - there are important adaptive consequences of this shift from canalisation of accretive variation to canalisation of compositional variation. Specifically, the latter enables a form of adaptation that uses the results of prior selection to inform subsequent selection. That is, in the Baldwin effect, the mechanism of exploration is fixed, whereas in SEAM, the mechanism of exploration is shaped by prior selection by virtue of the fact that the groups are formed from other coevolved individuals. (See ‘stochastic lookahead’ in 9.3.6.)

8.3 Future research

The models that we have introduced and the issues that we have touched upon in the subject of this thesis raise many questions and have a rich interface with other areas of study. In this subsection we briefly list some of subjects that deserve further attention. We do not intend to be comprehensive in sharing these ideas - we provide them for the purpose of encouraging further discussion and investigation.

8.3.1 The relationship of accretive and compositional mechanisms

In the models of previous chapters we have provided an operationalised model of compositional mechanisms. However, there is much to be done to properly integrate the findings of these models with existing frameworks of evolutionary change, not least accretive mechanisms. For example:

- **Integrating macro and micro evolutionary change.** A general model of evolutionary change might provide a framework that incorporates both accretive and compositional mechanisms, perhaps acting on different time scales or across different ecosystem scales. The role of compositional mechanisms in the major evolutionary transitions suggests that accretive and compositional mechanisms might correspond to intra- and inter-transitional mechanisms of change respectively. In such a framework, accretive mechanisms and competitive selection between non-complementary entities, might provide the common picture of micro-evolutionary change; whereas compositional mechanisms and coexistence of complementary entities, might provide a more macro-scale framework into which the former can be fitted.

- **The diversification and integration of lineages.** In the SEAM model we have presented thus far, we have illustrated mechanisms to maintain coevolving complementary lineages and their genetic integration into composites. However, a full model of evolutionary processes should also include mechanisms of speciation that allow both the usual ‘differentiation from a common ancestor’ and also the compositional ‘integration into a new whole’.

- **Integrating sexual recombination and symbiotic encapsulation.** Margulis (1993a) suggests we may view sexual recombination and symbiotic encapsulation as analogous mechanisms - the main difference being that the donor entities that create an offspring under sexual recombination share a closer common ancestor than the donor entities that create a new individual under symbiotic encapsulation. In SEAM we suggested that another important is seen (therefore) in the expected overlap of characters. More work is necessary to properly integrate our understanding of the two mechanisms and provide a general model of compositional change.

- **Hierarchical selection.** In 7.6.3 we briefly discussed the scale invariant properties of SEAM. Intrinsic to these ideas are the ideas of scaling-up the unit of variation and scaling-up the unit of selection. Obviously, hierarchical selection is a highly controversial topic in evolutionary biology. Our findings here do not intend to address the many issues underlying such discussion – we have focussed on understanding what difference it would make to the adaptive capacity of evolutionary processes *if* changes in these units were available through compositional mechanisms. We have shown that the difference is significant in some circumstances and we have come some way in identifying those circumstances.

- **Integrating multi-species models and population genetic models.** Often community models incorporate the interaction of numerous species but do not model evolutionary changes in the interaction coefficients of the underlying 'community matrix' (e.g. Arthur 1987) – that is, the competitive (or otherwise) relationship of species to one another is generally fixed, whilst the stability (or otherwise) of population dynamics are examined. In contrast, population genetics models do include changes in the genetic composition of a species but generally do not model the interaction of multiple species. In order to properly understand the balance of diversity and competitive exclusion necessary to properly model the diversification and integration of species under compositional mechanisms, it is necessary to integrate models of inter-species interaction and intra-species genetic change. In the models we have presented, some of the necessary machinery has been embedded in mechanisms such as deterministic crowding and selection based on Pareto dominance. A proper integration of the compositional mechanisms presented in this work with more conventional population genetics sub-division models on the one hand, and community stability models on the other is required.

- **Pareto selection, Nash equilibria, and evolutionary stable strategies.** The multi-dimensional treatment of fitness used in SEAM has some potential to assist in modelling the balance of competitive exclusion and stable coexistence that is required in a general model of evolutionary change incorporating compositional and accretive mechanisms. The relationship between the Pareto-dominance-based selection in SEAM, Nash equilibria, and mixed strategies (e.g. see Ficici in preparation), and evolutionary stable strategies (Maynard Smith 1982) warrants attention.

- **Towards a general model of subdivision and integration.** The general idea of subdivision and integration is well established in EB concepts through Shifting Balance Theory. However, in the GA models we have used the subdivision model is based on deterministic crowding. In SEAM the subdivision model is also largely the result of restricting competition to offspring and parents. In both cases the diversity of demes is largely the result of random initialisation. A more conventional model of spatial subdivision and diversification via genetic drift would properly relate the potential of compositional mechanisms with existing models of subdivision and integration. This would allow us to translate what we have learned about the adaptive capacity of sexual recombination and symbiotic encapsulation, and the class of adaptive landscapes to which they are well suited, into more conventional models. Meanwhile, the models we have presented suggest that Shifting Balance Theory and Serial Endosymbiosis Theory are similar in the sense that both are concerned with evolution of interaction systems in different lineages and their subsequent integration. The difference between them at this level of abstraction is that the former addresses subdivision (usually spatial subdivision) within a population of a single species and their integration via sexual recombination, whereas the latter addresses the adaptation of multiple species and their integration via symbiotic encapsulation. There is a lot of work to do to develop a better understanding of the differences and similarities in these mechanisms.

8.3.2 The open-endedness of evolutionary processes

The separation of a local optimum from the next best configuration of features is a fundamental limiting characteristic of adaptive landscapes, and saddle-crossing is a useful way to conceptualise the ability of an adaptive mechanism in some circumstances. But, what scale of fitness-saddle should we expect in a natural adaptive landscape? Intuitively, we might suspect that as one scale of ruggedness is overcome, a larger scale of ruggedness becomes the limiting characteristic of the adaptive landscape. If this is so, then there is no fixed scale of saddle-crossing ability that is sufficient for open-ended evolvability. In contrast, an adaptive mechanism that scales-up as evolution continues, enabling larger and larger ‘jumps’ in feature space by the composition of extant entities, is not necessarily limited in the same way.

In our experiments using a scale-invariant fitness landscape, we find that, as expected, accretive mechanism have a limit to the size of fitness saddle that they can cross. More exactly, as adaptation continues and the distance to the next-best optimum increases, adaptation by these methods becomes increasingly difficult. In contrast, SEAM is able to discover the epistasis structure in the problem, use collections of features in different entities to represent it explicitly, and by searching combinations of these entities is able to continue to find successful combinations of features through many hierarchical levels. Accordingly, these experiments show that on this class of adaptive landscape, evolvability under accretive modes of adaptation is inherently limited, whereas innovation by composition offers the possibility of inherently scalable, open-ended evolvability. Future research will seek to understand the generality of this result.

8.3.3 The inherent tension of innovation and reproductive fidelity (change and non-change)

Heritable variation is one of the fundamental axioms of evolutionary theory. However, it is a familiar irony that random variation is the source of new innovation but also inherently opposed to the maintenance of extant complexity. We might refer to this as the inherent tension of innovation and reproductive fidelity - or the tension of change and non-change. Evolution has created mechanisms, such as enzymatic repair, that reduce the *error rate* (Nowak & Schuster 1989) and increase reproductive fidelity, but still, the question remains: How can it be the case that variation may be suppressed (by whatever mechanism) without also suppressing the opportunity for innovation? Differential reproduction is also not such a simple concept as it might first appear. Specifically, it requires us to delineate the entities involved—to identify the entities whose reproduction could be differentiated. There are many biological cases where the relevant reproductive units are not so obvious—and it may be in principle inaccurate to draw distinct boundaries.

Symbiotic encapsulation offers an intriguing perspective on these issues. It is perfectly reasonable that a number of entities may each be individually stable and yet, via the discovery of successful compositions of these entities, there is still opportunity for innovation at a higher-level of organisation. Thus, composition presents no opposition between the stability or reproductive fidelity of the component entities, and the opportunity for innovation in entities at the next level of organisation. And significantly, this view is enabled by a willingness to repeatedly re-define the boundary of the entities involved in processes of differential reproduction.

The tension between reproductive fidelity and opportunity for innovation perhaps suggests an inherent limit to the complexity that can be evolved by accretive mechanisms – these notions being allied to the complexity ceiling coming from reproduction fidelity (Eigen & Schuster 1979). In contrast, it seems from the above observations that compositional mechanisms are not necessarily subject to the same limitations and may be better able to provide continued increases in complexity. Indeed, the solution provided by Eigen and Schuster for the reproduction fidelity problem is the formation of hypercycles from the assembly of formerly independent self-replicating molecules which is a compositional event. A formal treatment of closed-endedness and open-endedness in accretive and compositional mechanisms would be valuable.

8.3.4 Selfish genes and the evolution of cooperation

In the selection scheme of SEAM, all entities enter into associations for selfish motives (see 7.3). So, from this perspective, entities do not cooperate ‘out of the kindness of their hearts’ - i.e. they are not behaving altruistically. So, when entities discover a successful mutually beneficial group, they remain in it because it is in their own best interest (no single point mutation, or equivalently, no change in strategy for a primitive player) provides a fitness improvement. To this extent, we are in accord with the selfish gene model (Dawkins 1976).

However, we have also suggested that we can see the formation of high fitness configurations through composition as the formation of hierarchical cooperative groups (see 4.6.5). This is because it should be noted that a naïve model of individual selection would not discover high-fitness configurations. Specifically, a model based on the (appropriately named) ‘greedy’ selection strategy, selecting for an association whenever it is better on average than not having the association, fails to find high fitness configurations in HIFF (7.5.2).

The point then is not whether entities remain in an association for altruistic or selfish motives – the point is how the ‘immediate’ fitness landscape of the individual is modified such that the selfish fitness choices it makes now lead it to discover the mutually beneficial configurations rather than the mutual defect configurations. In SEAM this is achieved by the mechanism of group evaluation which guides the formation of associations towards those which are optimal rather than just immediately satisfying. We discuss the guiding effect of group evaluation in other sections (8.2.1, 9.3.5) but its exact relationship to notions of selfish genes and the evolution of cooperation is thus far at the intuitive level.

Related issues here include the possibility of evolutionary selection for different evolutionary mechanisms - a subject we have not addressed in this dissertation. The tension between mechanisms that provide effective adaptation for the population, and those that provide immediate benefits to the individual, are relevant here.

8.3.5 Modular interdependency in natural systems

In Chapter 4 we defined a system exhibiting modular interdependency precisely because that was the class of system that would distinguish the adaptive capacity of accretive and compositional mechanisms. It is not part of our claim that natural adaptive landscapes may exhibit these characteristics.

However, there are some observations to make that may be of interest on this topic. For many, (e.g. Holland 2000, Hartwell et al. 1999) the modular nature of biological systems, and other systems (Simon 1969), is ubiquitous and obvious. For others, different assumptions are normal - for example, we might suppose that gene networks are largely random, (as in the NK model of Kaufmann 1989, 1993). In Figure 8-1 below, we sketch some different possibilities.

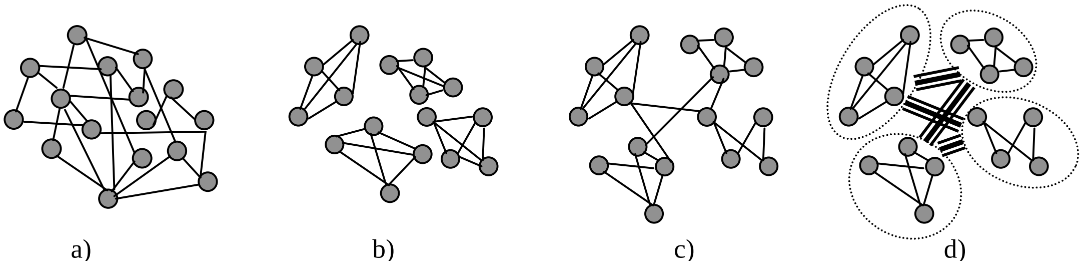


Figure 8-1: Possible interdependency structures

Nodes represent genes, edges represent dependencies between genes (see Figure 1, Case 3). a) random structure (as in N-K landscapes), b) separable sub-groups with only internal interdependencies (as in Royal Roads, concatenated trap functions), c) clustered dependency structure (as in ‘hdf’s, Holland 2000, and NKC landscapes), d) modular interdependency structure as in HIFF.

An important question to be asked is: When might the dependencies between variables in a biological system be clustered and organised in such a way that the interaction between clusters (something like c) can be abstracted into the interaction of higher level aggregate features (as in d)? We list some possibilities:

1. One possibility is that in a system of interdependencies (like that described in Section 4.3), the strength and structure of interactions between genes might itself be the result of adaptation. In this case a modular substructure might, for example, be the result of indirect pressures for ontogenic stability or robustness (Wagner 1995).

2. Another possibility is that modular epistatic interactions might be the natural result of expression in physical substrates. For example, consider two groups of nucleotides coding for two proteins that interact. The epistatic interdependency of many nucleotides is thereby ‘implemented through’ the interaction of the two proteins they code for. If the manner by which proteins interact is dependent on high-level or aggregate characteristics, such as their 3D shape or particular binding sites on the periphery of that folded shape, then the interdependencies of the two groups of nucleotides can be described by the relatively low-dimensional interaction of their aggregate effects. Such a hierarchical understanding of interdependencies is natural in our everyday understanding of biological systems (e.g. Holland 2000, Hartwell et al. 1999).

3. A third possibility is that modular interdependency structure in adaptive domains might be an indirect result of the modular structure of natural physical systems. Specifically, to the extent that the natural environment of an organism is neither an undifferentiated ‘soup’ nor a collection of wholly unrelated objects, but rather is a system of more-or-less differentiated but interacting objects, the problem of adapting to, interacting with, controlling, or modifying natural environments is also modular in structure. (See also Bak 1996, Gordon 1993).

Some other observations concerning modularity in natural systems include the fact that scale-invariance and fractal structures are often associated with biological systems (e.g. Mandelbrot 1982) and the self-organisation of dynamical systems (Kauffman 1993). ‘Small world networks’ (Watts 1999) have scale-invariant properties, occur in many different types of systems, biological and man-made, and seem to result from simple properties of incremental network construction (Slanina & Kotrola 2000). Lastly, it is interesting to note that spatially distributed dynamical systems with quite simple properties exhibit modular interdependency: For example, a hierarchically clustered distribution of communicating agents with a tendency to synchronise is sufficient. Consider a two-party voting system in humans with a tendency to shift allegiance toward the current majority and who communicate through local, state, and national elections. This defines a system where the stability of voting patterns has a modular interdependency structure.

These are only intuitions. But now, armed with a working definition of modular systems, and freed from naïve notions of ‘nearly separable’ systems for example, we are better equipped to search for and measure modular interdependency in natural systems. This may help formalise the intuitive ideas about the sources of modularity in natural systems listed above.

8.3.6 The ubiquity of compositional mechanisms

Although the symbiogenic origin of eukaryotes, including all plants and animals, is now accepted wisdom. The impact of symbiogenic events on evolutionary processes as a whole might seem minimal since such events appear to be rare. However, we have argued that symbiogenesis may be viewed as a particular kind of more general compositional mechanisms common to several of the major evolutionary transitions. Evidence for the involvement of compositional mechanisms is available for many of the transitions summarised by Michod (1999), for example (See 2.2.3).

So, composition is implicated in many of the most fundamental events in evolutionary history, but still, what about the complex adaptations that occur within a transitional level - the adaptation of metazoan features, for example, the eye or the wing? Maynard Smith and Szathmary 1995 clearly exclude this type of adaptation from the list of ‘fundamental’ evolutionary events precisely because they do not involve compartmentation or a change in the mechanism of information exchange (the units of inheritance). However, when we include mechanisms such as sexual recombination between diverse lineages, and allopolyploidy, in our list of compositional mechanisms we see that compositional mechanisms may act, albeit in a more subtle manner, on adaptations in between transitions as well as at the transitions themselves.

We also suggest that the notion of acquiring symbiont characters without direct genetic transfer, as in the model of ‘symbiotic scaffolding’ (Watson and Pollack 1999d, and Section 8.2.1), potentially broadens the influence of compositional processes to situations where there are no mechanisms of direct genetic transfer available.

Lastly on this topic, it is worth mentioning that the influence of compositional mechanisms should not necessarily be restricted to scenarios that ultimately result in full genetic integration. We have grown very accustomed to defining organisms by the common centre of genetic material, but more general notions perhaps based on reproductive interdependence might be sufficient – consider the complex siphonophores (e.g. Gould 1985 p.82), not to mention mitochondria.

Chapter 9 - Implications for Evolutionary Computation

9.1 Impact for Evolutionary Computation

This subsection lists the main areas of EC where the results of the previous chapters make an impact on existing understanding and theory.

9.1.1 Building Block Hypothesis

Given that the BBH is about the Simple GA, and is supposed to describe when it will work well in general, we cannot support the general form of the BBH. However, we have shown that a form of GA on a particular class of problem can assemble together modules in a manner similar to the intuition of the BBH. (see also 9.1.9)

9.1.2 Building blocks and modules - genetic linkage vs. interdependency

We have shown that the intuition of building blocks, and the idea that a problem can be reduced in dimensionality from the combination of primitive variables to the combination of building blocks of higher-order, does have some value. However, we have shown that the naïve interpretation of a building block as a separable partition of the variables (e.g. the Royal Road problems and Concatenated Trap functions) is inappropriate. Jones (1995) has already shown that a macro mutation hill climber, MMHC, can solve this class of problem.

Other methods in the literature defeat the MMHC by using random gene ordering (e.g. Harik 1997). Certainly, this makes the problem hard for MMHCs but it also makes the problem hard for the GA and crossover. Specifically, random genetic linkage prevents schema recombination under crossover.

In this thesis we have shown that a different and logical way to make a problem hard for the MMHC but not for the GA is to introduce higher-order interdependencies between modules. We have shown that this does not make the problem hard for crossover if diversity can be maintained and linkage is tight. Thus, when modules are short, and low-order (and identifiable via their fitness contributions as we have shown), then there is a form of GA that can optimise them and recombine them as the BBH suggests.

Having properly defined the concepts of modularity and difficult epistasis, and conceptually separated the concepts of decomposability and separability, we can show the adaptive capacity of ordinary crossover (using HIFF). We are then able to re-introduce difficult gene ordering (in Shuffled HIFF) to see if there is some (other) kind of EA that can optimise and manipulate this more general (gene-order independent) concept of a module (as distinct from the tight-linkage concept of a building block described by, for example, Holland and Goldberg). The resulting SEAM algorithm shows clearly that the concept of a module is not dependent on gene ordering but on the structure of epistatic dependencies.

9.1.3 GA-easiness

There is no good reason to suppose that a problem needs to have an upper bound on the highest-order non-linearity it exhibits in order to be solvable by a GA. Furthermore, problems in the literature often equate order-k delineation with order-k separability (e.g. the Royal Road problems and Concatenated Trap functions),[68] and there is certainly no good reason to suppose that a problem needs to have separable sub-modules in order to be solvable by a GA. Specifically, HIFF is not order-k delineable nor order-k separable - it involves interdependent modules and order N fitness contributions. Thus the order-k delineable and order-k separable concepts of GA easiness are not required for GA success.

Moreover, we have shown that the use of hierarchical modular interdependency exemplifies the distinction between the adaptive capacities of accretive mechanisms like mutation and compositional mechanisms like crossover. Accordingly, we suggest that modular interdependency is a useful concept of GA-easiness since it is, at the same time, provably difficult for accretive mechanisms.

[68] To see the distinction, note that Equation 5 defines a version of modular interdependency defined on order-2 dependencies and is therefore ‘order 2 delineable’ - however, this problem is not separable into order-2 blocks because fitness contributions come from overlapping pairs of variables.

9.1.4 Interdependency and NK landscapes

Equation 5 defines a version of modular interdependency defined on only pairwise dependencies and it is therefore ‘order-2 delineable’ (Equation 7 and Equation 8 could similarly be described using only order-2 dependencies in the style of Equation 5). However the family of NK landscapes with k=1 (where each variable is dependent on itself and one other) does not include this function. However, Equation 5 is built only from Pairwise interactions so it could be defined with something like an NK formalism with k=1 but where the fitness contribution of each variable is the sum of N fitness contributions dependent on k other variables and itself, (rather than just one fitness contribution dependent on k other variables and itself). Still Equation 5 defines a very particular instance of this class of landscapes. We know that most NK landscapes with k=1 are not very difficult for accretive mechanisms (Kauffman 1993). Similarly we expect that most order-2 delineable functions may also be easy. Thus we see that the particular structure of interdependencies, not just the number or order of dependencies, is essential in understanding problem difficulty.

Moreover, it has been noted that as K increases in NK landscapes the average height of optima decreases. Also the expected time for random search to find a (relatively) high-fitness point also decreases because many points are close to the optimum fitness value although they may be genotypically distant from the globally optimal configuration. In Equation 5 the globally optimal configurations are only fitness contribution=1 higher than the next best points, and so on down through the hierarchical levels. In fact, all local optima for even a bit-flip hill-climber have a fitness of at least half the fitness of the global optima in this landscape. If, in an attempt to increase the fitness difference between the global optima and other optima, we were to make the fitness contributions for larger modules larger then inter-module dependencies then these would ‘over-power’ intra module dependencies (see Figure 5-3). This helps us understand the low likelihood of broad fitness saddles in NK landscapes with low k where fitness contributions between all variables are drawn from a uniform distribution.

In contrast, in HIFF we use explicit higher-order dependencies between variables. This allows us to make the fitness value of optima arbitrarily higher than the fitness values of sub-optima.

9.1.5 Module acquisition and linkage learning

In SEAM, we have shown an equivalence between concepts of linkage learning and module acquisition. In other words, by solving Shuffled HIFF with random linkage we discover where the modules are in the problem. We represent these explicitly as modules that can be composed together. This overcomes the in principle limitations of representing dependencies in linear form (i.e. by their arrangement on the chromosome - e.g. Harik 1997. See also 6.4.4). By explicitly representing the interdependent variables as modules we are both ‘learning the linkage’ in the problem and performing automatic module acquisition.

9.1.6 Crossover is not just macro-mutation

It should be clear from our experiments that crossover cannot and should not be approximated as a source of random variation. The multiple inheritance feature of crossover (2.5.1) and the assembly of higher-order building blocks from low-order building blocks (as Holland originally suggested) is intrinsically unavailable in single-inheritance methods like hill-climbing, the GA without crossover, or the GA with strongly converged populations (i.e. the GA without a diversity maintenance mechanism).

This puts GAs using crossover and appropriate diversity maintenance in a fundamentally different class of algorithms from mutation only algorithms (Culberson 1995).

9.1.7 Competing conventions problem

We have shown that the competing conventions problem is not necessarily a problematic issue for crossover. HIFF has strongly competing (mutually exclusive) solutions to each module in every partition of the variables, but a GA with tight linkage can overcome the disruption caused by the recombination of incompatible sub-solutions of different ‘conventions’ if it has an effective means of selecting against poor offspring and for identifying valuable new schemata. We have shown that the GA using deterministic crowding is sufficient for this purpose in this class of problem.

9.1.8 Credit assignment

We have shown that there are principled methods for determining the value of a schema for the purposes of assembling schemata together. The method we use is not based on schema fitness - the average fitness of strings included in that schema - instead it is based on the influence that that schema has on groups of other schemata in group evaluation. A schema is selected because it is non-dominated rather than because it is good on average. (see 9.1.9, and 9.3.5).

9.1.9 Separation of the Building Block Hypothesis from the Schema Theorem

It should be clear that since we do not use fitness proportionate selection or any other method that allows inferior schema to be propagated - we are therefore not interested in schema disruption except in so much as it suggests a lower rate of schema creation. Similarly, we do not require useful schemata to be duplicated in the population. Thus, we are not interested in generating multiple copies of fit schemata, or in assessing the average value of a schema (see previous point). Therefore the Schema Theorem is simply not relevant to the kind of GA that we are examining. And thus the validity of the Schema Theorem is not required for the supply of building blocks for recombination in this type of algorithm.

Accordingly, we have separated the validity of the Schema Theorem for GAs in general, from the intuition of building block assembly in the BBH.

9.1.10 Evolution and coevolution in the GA with crossover

SEAM transforms a single objective optimisation problem into an explicit coevolutionary game amongst schemata. It is not so obvious that the regular GA is also exploiting a similar idea - but in fact, it is. Specifically, in the GA, we are coevolving schemata in groups (i.e. many schemata reside in each individual), and exchanging schemata between one individual and another to, implicitly, try and find a location for that schemata where they are more stable (confer higher fitness and are less likely to change again).

Acknowledging that even the regular GA with crossover is inherently a coevolutionary system, and needs to be treated as a coevolutionary system, helps explain why in the past the GA with crossover has often failed to perform better than the GA without crossover. Specifically, if the practitioner does not realise that the GA with crossover is a coevolutionary system then they will probably not respect the need for the diversification of lineages that collectively cover the space of interesting schemata. Accordingly GAs with crossover are often used without any diversity maintenance mechanisms, and even when diversity maintenance schemes are used they are often seen as a secondary ‘add-on’ to the GA.

Treating the GA with crossover as a coevolutionary system, it becomes obvious that we require a system of diversity maintenance to make the recombination of schemata valuable. Purely competitive selection pressures do not allow diversity. Thus fitness sharing methods or some kind of diversity method is required. Methods such as these respect the need for complementary individuals (too many similar individuals is undesirable) - and this is therefore not purely competitive.

However, there is a subtle difference between recognising that things should not be all the same (implying that even arbitrary diversity is desirable), and recognising that we need things to cooperatively cover complementary parts of the problem domain (which is more specific in the kind of diversity required). We suggest that a cooperative coevolution approach to diversity maintenance is valuable. For example, the Pareto Coevolution idea explicitly recognises that the value of an individual is related to its ability to fill a gap in the problem domain and that selection should be competitive within a niche and cooperative across niches. The notion of the population as the Pareto front, and the multi-dimensional treatment of fitness explicitly represent this.

9.1.11 Optimisation at several scales vs. optimisation at any one scale

One way to understand the operation of the GA and SEAM is in terms of building block assembly through many hierarchical levels of organisation - a view that we have emphasised repeatedly. Another way to understand the operation of these algorithms is as mechanisms that incrementally increase the scale of optimisation, as follows.

Let us call optimisation based on selection for fully-specified strings ‘large-scale optimisation’, and in contrast, let us call optimisation based on selection for individual alleles/ individual feature values ‘small-scale optimisation'. [69] These ordinarily correspond to selection on the organismic scale and selection on the genetic scale respectively.

We have seen that in a problem with a highly rugged landscape such as HIFF, small-scale optimisation fails to find optimal configurations. For example, uniform crossover in the GA allows individual alleles to be exchanged between individuals and fails to find optimal configurations in HIFF. This is as would be expected for a landscape with significant epistasis between alleles. [70] (See Potter & De Jong 1994, for an explicit implementation of small-scale optimisation using selection on the individual variables of a function optimisation problem.)

We have also seen that large-scale optimisation fails in problems over a large number of variables such as the HIFF problems that we have used. For example, Random Mutation Hill Climbing is based on selection for fully-specified strings, and repeated runs of RMHC fail to find optimal configurations of variables in HIFF. Again, this is not surprising given the high epistasis in HIFF and the large search space of large HIFF problems. Note that in principle, optimisation of fully-specified strings explicitly allows for selection on coadapted sets of alleles (similarly, the GA without crossover), and thus deceptive allelic fitness contributions can in principle be overcome - i.e. since the individual is reproduced as a whole, the selfish interests of the genes can be overruled by selection on the individual. However, the combinatorics of multi-variable problem spaces with interdependency make optimal solutions in large-scale optimisation infeasible in all but small problems.

[69] These are not the same as the terms 'local optimisation' and 'global optimisation' – both small scale and large scale optimisation may follow local gradients and become stuck at local optima depending on the interaction of variation and selection.

[70] Note that the alleles that are promoted in small-scale selection are selected for because of their fitness in the context of the other alleles in individuals 'built from' other alleles in the population (see the 'rowers' analogy of Dawkins 1976). Accordingly, the (sub-optimal) strings that are evolved under small-scale selection may include alleles that are significantly coadapted to one another. For example, in HIFF, each allele at every locus is of equal value on average over all contexts, thus a string built from a distribution of alleles biased only by the independent average fitness contributions of individual alleles will be a random string. However, in the strings that result from small-scale optimisation, the particular alleles that the population converges to will be better adapted than a random string. For example, a sub-optimal string consisting of sub-blocks of ones and zeros might result from small-scale optimisation (depending on the mutation rate, population size, how many evaluations are used in search); and such a string will confer higher fitness than would a random string on average – i.e. such a string will show fitness contributions coming from the coadaptation of alleles.

In summary, neither small-scale nor large-scale optimisation is successful in a problem with high epistasis like HIFF. Specifically, the particular configuration of alleles that small-scale optimisation settles on will include alleles that are coadapted to the distribution of alleles in the population but, if variables have difficult interdependency then this will generally be sub-optimal. In large-scale optimisation, an optimal configuration could be selected for if it occurred even in cases where the distribution of alleles in the population as a whole conflicted with the configuration of alleles in such an individual (because complete specifications are promoted as a whole) but the chances of finding such a configuration in large search spaces is very small.

Now, in SEAM and the GA with tight linkage, there are many different scales of optimisation that are employed. In the GA with crossover this occurs simply because crossover allows the transference of building blocks of different defining lengths at any time. In SEAM this occurs because the algorithm explicitly searches combination of extant entities and thus proceeds from searching combinations of alleles, to combinations of low-order schemata, to combinations of high-order schemata, and so on. This utilisation of both small-scale optimisation and large-scale optimisation allows these algorithms to extract the advantages of optimisation at many scales. Specifically, to the extent that candidate solutions to low-order modules can be identified independently of context, small-scale optimisation utilises divide and conquer algorithmic advantage. This is not sufficient to find globally optimal solutions, but it can reduce the number of possibilities that need to be considered in subsequent search (4.3.1). Then continued adaptation at the next scale of optimisation can enable selection on internally interdependent modules at the next scale, and so on.

Thus we see that at one extreme, searching for solutions by selecting for the individual fitness contributions of alleles exploits the decomposability of the problem where available, but is sub-optimal on its own. And at the other extreme, searching for solutions by selecting for the fitness values of complete sets of alleles can be optimal in principle, but is combinatorially prohibitive. However, a process that moves incrementally from small scale to large scale can gain the benefit of problem decomposition where available to reduce the dimensionality of search at the next scale of optimisation and so on - thus providing optimal solutions in this class of modular interdependency problems where neither small-scale nor large-scale optimisation can alone.

The ideas of optimisation at several scales are clearly apparent in the intuition of the BBH, in the ideas of automatic module acquisition, in approaches to hierarchical classifier systems, and such like. In SEAM we have defined a principled mechanism for progressing between levels, for using the entities of previous levels in subsequent levels, and continuing the process over many hierarchical levels in an explicit and consistent fashion.

9.2 Understanding modularity

This section briefly overviews the relationship of modular interdependency to work in complex dynamical systems, Herb Simon's work on structured systems, and concepts of problem difficulty developed in the EC literature.

9.2.1 Modular interdependency, Ising models, and renormalisation groups

In Chapter 4 we introduced the notion of a fitness dependency between genes as a static measure of their dependencies. A complementary notion of dependency may be formulated as a dynamic interpretation of state-updates in a dynamical system. In this view, instead of interpreting an edge of the graphs like those depicted in (Figure 4-3), as a direct representation of a fitness dependencies, we may for example, interpret an edge as a gene expression interaction where one gene up-regulates or down-regulates the expression of another. This exactly parallels the relationship between the dynamics of Random Boolean Networks (Kauffman 1993) and the statics of NK landscapes (ibid.).

In the dynamic model, we are interested in the attractors of the system (expression patterns that will be common in the lifetime of the cell) and properties of the system such as the number of attractors in the system, the nature of the attractors (such as fixed point or cyclic), the stability of the attractors (stable or unstable), and the sensitivity of the attractors within the system to the external conditions of the cell's environment. In this analogy we can draw association between the dynamic properties of the network and the static properties of a fitness landscape built from a similar structure of interdependencies. For example, if the stability of a configuration in the dynamical system is related to high fitness, then the attractors of a dynamical system are optima in the corresponding fitness landscape.

A question that follows from this interpretation is: What are the properties of a dynamical system built on a modular interdependency structure, and how does this relate to the fitness landscape defined on the same modular interdependency structure?

If we translate each edge in the dependency graphs to a state update rule (e.g. (0,0→0,0), (1,1→1,1), (0,1→0,0 or 1,1) and (1,0→ 0,0 or 1,1))[71] then the configurations with high fitness contributions are the stable configurations in the dynamical system, and vice versa. This particular state-update table is that used in Ising models (see e.g. Van Hoyweghen et al. 2001b). And the 'energy' of a configuration, (measuring the number of edges in the graph that connect nodes with unequal states) is a direct measurement of the likelihood of change, and is inversely related to the system's stability. Thus, the energy function of an Ising model with the connectivity structure of HIFF, is structurally equivalent to the HIFF fitness function.[72] (see Van Hoyweghen & Naudts 2001 for a discussion of HIFF and Ising models.)

So, to answer the above question, there are strong associations between the dynamics of Ising models built on scale-invariant lattices, and the fitness landscape of systems with modular interdependency like that defined by HIFF. In particular, such a dynamic model has attractors corresponding to the same configurations of states as local optima in HIFF, and the size of the perturbation required to move the system to a more stable configuration is the same as the width of the fitness saddles for that configuration in HIFF. Thus the lowest energy, most stable, configuration of the dynamical model, and the highest fitness configurations of the static model are coincident - i.e. all ones and all zeroes. Configurations of intermediate stability, such as where all large sub-domains are internally in agreement but not in agreement across sub-domains, correspond directly to configurations in HIFF where large modules are fully-optimised internally, but inter-module dependencies are not resolved.

[71] where (s1,s2→s3,s4) connected by an edge and taking values s1 and s2 will have tendency to change to the values s3 and s4 in the next time step. The actual new state of a variable in the system can be taken to follow the majority of these tendencies coming from all connections. (s1,s2→s3,s4 or s5,s6) means that two variables will change to either the values s3 and s4 or the values s5 and s6 with equal probability.

[72] Ising models are usually defined with a regular 1, 2, or 3D lattice connectivity structure. But Ising models on scale-invariant lattices, such as on a graph described by the edges in a Sierpinski triangle, are an item of study in and of themselves.

Recognising this equivalence between what started as a hierarchical building block function and a particular class of Ising models is interesting in itself, but importantly, it also provides access to an extensive repertoire of mathematical treatments and analysis, not least the renormalization group developed by Wilson (1979). The renormalization group is an analytical tool that explicitly abstracts the interdependency of low-level variables into the interdependency of higher-level variables. In scale-invariant lattices, this renormalization results in a higher-level system over aggregate states that is the same as that which described the interaction of the primitive states. And in fact this can be applied recursively. In regular lattices, the high-level model is only accurate when the 'temperature' of the system (controlling the likelihood of state changes) is at a particular value - the critical temperature. At this temperature, even regular lattices exhibit interesting scale-invariant phenomena.

Renormalization groups have been enormously successful at describing many types of dynamical systems. The connections between modular interdependency of the type used in HIFF, the dynamics of simple dynamical systems like Ising models, and the ubiquity of systems that are amenable to renormalization group analysis, suggests, loosely, that the type of dynamical systems occurring in many physical systems may, under certain circumstances, exhibit characteristics in common with functions of modular interdependency like those we have defined here. This hypothesis deserves further investigation.

9.2.2 Simon and nearly decomposable systems

Herb Simon (1986) presented some important ideas on modularity, hierarchy and the general structure of many different types of system. It may be valuable to clarify the distinctions between Simon's ideas and those presented in this dissertation. Simon's basic ideas about hierarchical modularity, interactions based on aggregate effects, and representing interdependency with an almost-diagonisable matrix, are clearly present in the modular interdependency framework we presented in Chapter 4. Taking a dynamical interpretation of HIFF (see 9.2.1) Simon's description of the properties of 'nearly decomposable systems' given in 4.4.1 are true of HIFF. However, we have also built-upon and clarified some issues that were not so clear in Simon's work as described below.

The difficult and the easy kind of interdependency

The only concrete example of a nearly decomposable system that Simon provides in (Simon 1969) with more than qualitative description is a system of heat exchange in a building containing a number of rooms each containing a number of cubicles. Unfortunately, this system has only one attractor – eventually all the variables in the system approach the mean temperature of the initial condition. Let us see what this would mean if we interpret this system as a problem to be optimised. For example, let a candidate solution be a configuration of temperatures, and let the fitness of a configuration be its ‘stability’ or the inverse of its internal heat disequilibrium – specifically, the fitness of a temperature configuration will be the sum of inverse absolute temperature differences between all areas. We may describe this system using the general form of modular interdependency described in Equation 9, with ***t***(p,q)=(p+q)/2 and ***f***(p,q)=1/|p-q|. A simple hill climber or annealer following the local gradients in this function will follow approximately the same trajectories through configuration space that the temperatures would in the physical system. However, the important property to note is that this function has no local optima – there is no configuration of temperatures for which there is no small change in temperature that reduces the energy of the system (increases the fitness of the system). In short, Simon has not described his class of systems in such a way as to distinguish between systems that have easy and difficult interdependencies (See 4.2).

The distinction between decomposable and separable

Although Simon describes the interaction between modules as interaction which “depends in only an aggregate way on the behaviour of the other” modules – which is fine – he is not careful to clarify that this interaction may be strong and highly significant. Simon repeatedly refers to the idea that “interactions among the subsystems are weak but not negligible”. For example, he refers to “a set of stable subsystems, each operating nearly independently of the detailed processes going on within the other subsystems” (p.193) and “Intracomponent linkages are generally stronger than intercomponent linkages.” (p.204). This interpretation of ‘decomposability’, approaching what we have termed ‘separability’, is unnecessary. In HIFF the interdependency between modules is strongly non-linear and highly significant – the modules are not separable – yet they are still easily decomposable. When decomposability is interpreted as nearly-separable (which is what Simon means by the term nearly decomposable) it mistakenly confuses the strength of interactions for the structure of interactions.

In summary, we see that Simon contributed a significant number of valuable concepts to the ideas we have developed in this thesis, but we have developed Simon's notions in terms of evolutionary difficulty – in particular, the notion of decomposable but not separable systems and concepts like 'modular interdependency' (4.3.1) replace the idea of 'nearly decomposable' systems based on systems with weak sub-system interaction.

9.2.3 Deception

'Deception' has been proposed as a concept of evolutionary difficulty (e.g. Deb & Goldberg 1992a, 1992b), and has been used extensively in building block functions such as the concatenated trap functions. As Deb and Goldberg indicate, there are many different definitions of deception but the basic idea is that high-order schemata of high fitness (e.g. the global optimum) are not contained in low-order schemata of high fitness– in other words, if search focuses on high-fitness low-order schemata (as one supposes it will in the GA) then it will not by this means be led to discover high-fitness high-order schemata. We have already indicated that the concatenated trap functions lack building block interdependency and that the fact that the building blocks in this function are separable makes them an inappropriate test of building block recombination for crossover in the GA. But let us look at the idea of deception that is used to make each independent building block in the problem difficult for the hill-climber.

One way to understand deception is as follows. In a deceptive problem there are different competing schemata in a partition – generally, the 0s kind and the 1s kind. Assembling together one kind of schemata leads to the global optimum, but assembling together the other kind does not (see also 9.1.7). Now if, the kind of low-order schemata that does not lead to the global optima actually has a higher schema-fitness than the kind that does lead to the global optima, then we can understand why this would create a problem for an algorithm that uses the heuristic of assembling together high-fitness low-order schemata.

Often the misleading schemata in deceptive problems are semantically associated with the idea of an arbitrary 'aberration' in the fitness contributions. Indeed, it seems *un*reasonable to suppose that the fitness contributions of low order schemata will always be reliably informative of their potential to provide high fitness high order schemata. However, it would not make sense to abandon the idea that the fitness contribution of schemata are genuinely informative – this is the necessary heuristic of fitness-guided search techniques. Accordingly, there must be some limit on the occurrence of arbitrarily misleading schemata in the problem; this leads to the idea of 'bounded difficulty' where schemata up to a certain order k are deceptive but schemata above order k are not; and thus we arrive at the concatenated trap functions.

The ideas of problem difficulty in HIFF are related to these ideas of deception but are different in important respects. First note that HIFF does include the idea of different competing schemata in a partition – specifically, the 0s kind and the 1s kind. However, in HIFF each kind of low-order schemata has the same fitness contribution and assembling together either kind of schemata leads to one of the global optima. In Biased-HIFF (Watson & Pollack 1999a), the fitness contribution of schemata built from 1s are higher fitness than those built from 0s and the problem therefore has only global optimum at all 1s. In Biased-HIFF we might term the 0s schemata as deceptive. However, even in biased-HIFF the 'misleadingness' of low-order schemata is non-arbitrary in the sense that the fitness of low-order schemata accurately reflects the value of the high-order schemata that they contain. More specifically, the ratio of fitness contributions coming from low-order schemata of the two types is the same as the fitness of the size-N complete strings of those types. This follows directly from the hierarchical consistency of the constructions functions used in HIFF (Equation 9). The idea of hierarchical consistency is intrinsically opposed to the idea of low-order fitness contributions being arbitrarily over-ruled by high-order fitness contributions as seen in 'fully-deceptive' problems (Deb & Goldberg 1992a).

However, there is a more informative way to look at the misleadingness of low-order schemata in HIFF. Specifically, note that although modules built from 0s and modules built from 1s both confer fitness contributions and these always accurately reflect their potential for involvement in high fitness higher-order schemata, the difficulty arises from the need to search for the correct combination of these modules. Thus, although for example, --00---- and 11------ both confer independent fitness contributions and both include one of the two global optima, the 1100---- schema does not include either of the global optima. Accordingly, we see that the difficulty coming from competing schemata in HIFF does not come from the fact that one kind of schema is 'correct' and the other kind of schema is 'misleading' – rather the difficulty comes from the incompatibility of some types of schemata with other types of schemata in the assembly of a complete solution.

We agree that the notion of arbitrarily deceptive fitness contributions, seen in fully deceptive trap functions for example, is a possible source of problem difficulty in applied domains. In this view a problem is made up of a mixture of correct and incorrect fitness contributions. However, the idea of schema competition in HIFF is more accurately understood as a mixture of types of schemata each of which are correct in some sense, and we suggest this is a useful alternative interpretation of problem difficulty.

9.2.4 Relationship of HIFF to some related concepts of problem difficulty

Figure 9-1 shows the relationship of various pieces of existing work to the HIFF function and problem definition we have developed in this thesis.

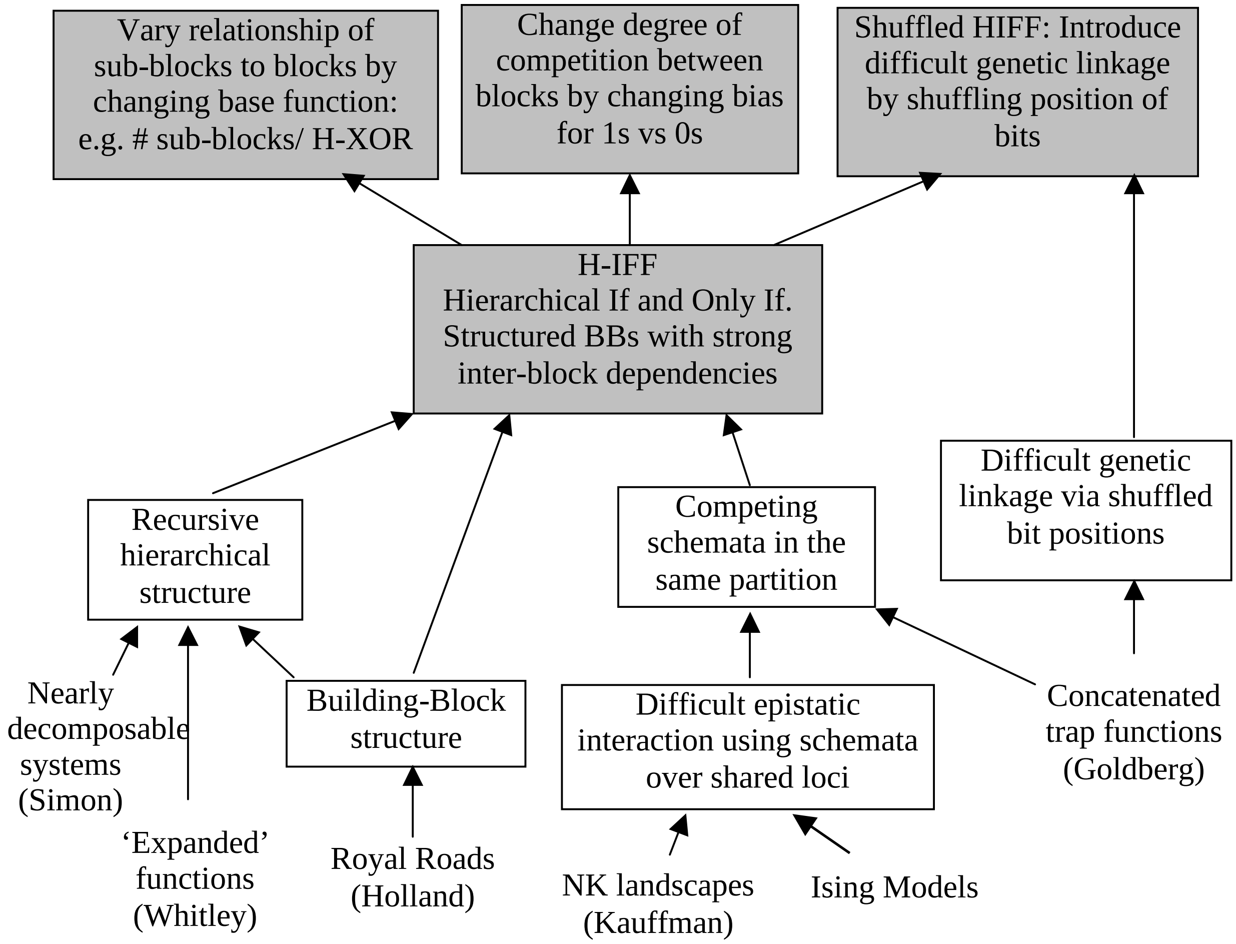

Vary relationship of sub-blocks to blocks by changing base function: e.g. # sub-blocks/ H-XOR
Change degree of competition between blocks by changing bias for 1s vs 0s
Shuffled HIFF: Introduce difficult genetic linkage by shuffling position of bits
H-IFF
Hierarchical If and Only If.
Structured BBs with strong inter-block dependencies
Recursive hierarchical structure
Competing schemata in the same partition
Difficult genetic linkage via shuffled bit positions
Nearly decomposable systems (Simon)
'Expanded' functions (Whitley)
Building-Block structure
Royal Roads (Holland)
Difficult epistatic interaction using schemata over shared loci
NK landscapes (Kauffman)
Ising Models
Concatenated trap functions (Goldberg)

Figure 9-1: Relationship of HIFF to other test problems

Shows relationships of problem components; previous work, key concepts and new research. This map shows one way to interpret the combination of features developed in HIFF and the derivation of those features from existing problems.

9.3 Understanding composition

In this section we briefly outline some of the preliminary work that was involved in the development of SEAM detailed in Chapter 7. This is provided to familiarise the reader with some of the underlying concepts, and their derivations in pre-existing work in evolutionary algorithms. In actuality, the basic idea of symbiotic encapsulation (Watson 1996) pre-dates this background work. However, this work provides a series of intermediate results and investigations that help position SEAM in the broader context of existing EC methods.

9.3.1 Recombining versus combining

We saw in Chapter 6 that when gene ordering is randomised, crossover is unable to exchange meaningful subsets of variables and recombine them in an offspring. As we have mentioned there are various different approaches that may be employed to overcome this problem, such as crossover masks, moving-locus representations, and other linkage learning methods. The basic problem is the problem of determining which part of an individual constitutes a module - which part of an individual is a good part to transfer to the offspring. When gene ordering is random, the parts taken by one-point crossover, for example, do not correspond to meaningful modules.

The Messy GA uses a combination of (at least) two features to address this problem. One is the use of partially specified individuals, the other is the use of a moving-locus representation (See Section 3.4.2). In previous work (Watson & Pollack 1999c) we showed that the moving-locus aspects of the Messy GA are perhaps less important, (at least, not essential), to enable successful recombination in a modular interdependency problem like HIFF. By using just the partial-specification features of the Messy GA it is, in principle, possible to allow each individual to represent exactly one module explicitly (See Figure 3-10). (This is in contrast to the regular GA where an individual is a fully-specified feature set and therefore represents a collection of modules.) Using partially specified individuals in this manner there is no problem in extracting the appropriate parts from a parent in crossover because a parent *is* a part.[73] In this sense, we see a shift from *re*-combination, where modules are moved from one assembly to another assembly, to *combination* where modules are put into an assembly explicitly for the first time.

9.3.2 Problems with partial evaluation, bloat, and diversity

However, the algorithm presented in previous work (Watson & Pollack 1999c) had some serious limitations. First, we needed to evaluate partially specified individuals; Second, we needed a way to prevent sub-optimal associations from being made; Third, we needed an appropriate way to maintain diversity. These issues were handled in that work, but only in a limited manner.

[73] “From square one, we would like to be able to evaluate the fitness of a part without possessing a whole.” (Goldberg et al. 1989).

It so happens that in HIFF, there is a natural interpretation of a partially specified string - we can simply fill in the unspecified loci with “nulls”. But this may not be an available strategy in more realistic domains that require fully-specified strings in order to perform an evaluation. With regard to the second limitation, we used a size penalty that prevented a premature increase in the size of strings until optimal joins had been formed. However, this penalty function required explicit knowledge of how the fitness of strings may grow with size. We have such knowledge in HIFF, of course, but again, in general, such knowledge will be unavailable. With respect to the third limitation, we used the resource-based fitness sharing method as in earlier work (Watson et al. 1998). This works very well, but again uses explicit knowledge of the modules in the problem that is usually unavailable. It is not clear how the deterministic crowding method that we used to maintain diversity in Chapter 6 should be adjusted to work on partially specified individuals (e.g. what is the distance between two individuals that are non-overlapping in the features they specify?), and moreover, the deterministic crowding method assumes a genotypic similarity metric.

At this stage, our three problems were: evaluation of partially specified individuals, limiting exponential growth of strings (and premature commitment of unspecified variables to sub-optimal values), and diversity maintenance.

Note that these limitations are not trivially solved in the Messy GA or other methods either. In the Messy GA, the growth of strings is limited to order k in the first phase of the algorithm - which assumes that there is a small k equal to the highest order non-linearity in the problem (which there is not in HIFF), and it assumes that have knowledge of this value. Goldberg et al. (1989) suggest the use of ‘thresholding’ as a diversity maintenance technique (where individuals that are too dissimilar are not allowed to breed together). This method of diversity assumes that a distance metric is available to discern how similar two parents are. And genotypic Hamming distance is not always an appropriate measure of functional dissimilarity.

However, there are some valuable clues available in previous work also. Goldberg et al. (1989) suggest the use of ‘templates’ to permit partially specified individuals to be evaluated. These are fully-specified strings used to fill-in unspecified features of the individual being evaluated. In later versions of the Messy GA, Goldberg et al. (1990, 1993) suggest that templates for one round of evaluation could be built from strings generated in previous rounds, hence re-using evolved blocks to make templates for subsequent evolution.

Meanwhile, in Cooperative Coevolution, Potter (1997), suggests the use of a shared domain model that enables the evaluation of an individual, fulfilling one particular role in a collective solution, to be evaluated in the context of other individuals coadapted to fill-in other roles in the collective solution. This is not so different from the idea of templates in the Messy GA. Both use the idea of using coevolved individuals to build templates to evaluate subsequent individuals. However, Cooperative Coevolution does not use a join operator to explicitly make higher-level composites - it is a single-level decomposition of the problem domain. Since Cooperative Coevolution does not use a join operator, there is no problem of individuals growing in size. However, a single level of problem decomposition also means that the problem can only be optimised to the extent that the individual fitness interests of individuals in the different roles do not conflict with the optimal configuration for the group as a whole.

So, these methods suggest mechanisms that evaluate individuals in the context of other coevolving individuals. SEAM develops this idea: that is, it evaluates individuals in templates built from other coevolved individuals. SEAM evaluates partially specified individuals in groups that collectively cover the problem variables. Thus, we only evaluate fully-specified feature sets and do not require the direct evaluation of partially specified individuals. However, we use the information from these group evaluations to evaluate the merit of a proposed individual in the group.

However, Goldberg et al. (1989) caution that the use of templating introduces the problem of 'low signal to noise ratio' in the evaluation of individuals (i.e. noise in the evaluation coming from the fitness contribution of the templates is high relative to the signal coming from the fitness contribution of the individual being evaluated), and also that an arbitrary template may not reveal the necessary dependencies that a schema has with its genetic context.

9.3.3 Template differencing

Our solution to the problem of low signal to noise ratio in SEAM involves something we might call 'template differencing' that is allied to the method of evaluation in Cooperative Coevolution.

Naively, we might use group evaluation proposed above and simply share-out the fitness of the group to the members of the group. However, this necessarily suffers from the problem of bloat where sub-optimal feature specification accumulate or 'hitch-hike' along with fit specifications. An alternative, is to examine the fitness difference that an individual makes to a group when it is added or removed to/from the group, as in the shared domain model of Cooperative Coevolution.

So, the idea of template differencing is that we will measure the fitness of a group with and without the individual of interest and difference the two. This will give us an indication of how much the individual contributed to the group. However, because there are fitness dependencies between the individual and the group, using just one group will not give us a proper sample of an individual's possible contribution. Accordingly, we must test an individual in many groups.

However, this does not solve the problem of how to prevent exponential string growth (see the credit assignment problem/'hitch-hiking' problem 3.4.4): i.e. a sub-optimal join is usually better than a random join so individuals fill-up with sub-optimal feature specifications (even when we use template differencing to reduce noise).

9.3.4 The pressure to fill-in missing features

In (Watson et al. 2000b) we observed that when the missing features a partially specified individual are 'scaffolded' by the features of other coadapted individuals there is in some circumstances a tendency for individuals to be, we might say, 'complacent' about whether or not they fill-in for the missing abilities themselves. Briefly, if symbionts are reliably available in the ecosystem then there is no pressure for individuals to acquire the missing features for themselves (see Harvey 1993 for the equivalent phenomenon in the Baldwin effect). In a model of genetic assimilation of acquired characteristics this is a problem. But for our purposes, it offers an opportunity to alleviate the pressure for partially specified individuals to fill-in with sub-optimal features.

However, preliminary work using group evaluation (with or without template differencing) failed. If we took the average fitness an individual in many groups, or the maximum, (or selected for minimising detrimental effects to a group), individuals either never allowed additional features to be specified, or allowed too many additional features to be specified as before. At this stage we needed a principled way to determine when a join really created a good higher-level module rather than merely a collection of better-than-average modules. The solution in SEAM can be understood by two shifts in perspective. First, we use a more principled definition of what makes of a 'good module' - What does it mean to be a good module and how would we find one? Second, rather than develop an evolutionary process that moves from poor solutions to good solutions, we will develop an evolutionary method that moves from specialised solutions to general solutions. The ideas of ‘stochastic lookahead’ and Pareto Coevolution, respectively, were critical in addressing these issues.

9.3.5 The Baldwin effect and ‘stochastic lookahead’

In a different piece of work (Watson & Pollack 1999d, Watson et al. 2000b), we were investigating the role of symbiosis in an effect allied to the Baldwin effect (Baldwin 1896), which we called ‘symbiotic scaffolding’. The idea was that if we evaluate partially specified individuals in groups then this lifetime interaction would guide the evolution of fit individuals to indirectly acquire the characteristics of the group (see 8.3.1). The general idea is that a mechanism of rapid non-heritable variation may guide a mechanism of relatively slow heritable variation. In the Baldwin effect, lifetime learning guides genetic variation. In symbiotic scaffolding, lifetime interaction with symbionts guides genetic variation.

We call the general idea of allowing a fast non-heritable variation mechanism to guide a slow heritable variation mechanism ‘stochastic lookahead’. This sounds ‘unevolutionary’ for the same reason as the Baldwin effect sounds unevolutionary at first, but it is not, for the same reason that the Baldwin effect is not. (See Watson et al. 2000b).

SEAM uses the idea of stochastic lookahead to enable us to identify good modules, as follows:

> We want to select for ‘good modules’. But, what makes a ‘good module’? A good module is precisely the thing that, when used in assemblies in future, will produce good assemblies.
>
> So, how do we find such an entity? - By trying it in a number of assemblies and seeing how it does.

Clearly, if we try a proposed module in a number of assemblies and it ‘does well’ then it is precisely what we want from a module. But this raises two questions: 1) How can we expect to make a good assembly, with which to find good modules, before we have good modules? If we make poor assemblies then they will not reveal the potential utility of the proposed module. 2) If we already made a good assembly, then we are ‘done’ - why should we be interested in the module? Why do we not just keep the good assembly?

These questions and their answers are interrelated. First, the simple answer to the first question is that we do not know how to make good assemblies, but we can make many assemblies from the modules we have so far, and hopefully some will be sufficient to reveal the utility of a proposed new module. Note that the better are our modules so far, the better assemblies we can make. Second, the answer to the second question is that the best assembly we find initially will be poor. To keep the best assembly we have found so far is the ‘greedy’ method that finds sub-optimal configurations. We should not suppose that the assemblies we make are perfect, just that they are good enough to reveal good modules. Then when we have better modules, we can use these to build still better assemblies, which in turn are used to find larger modules, and so on.

These ideas are very similar to the ideas of SAGE (Juille 1999, p. 50-51). SAGE uses a partial solution as the beginning of a complete solution that is filled-in at random or by using an incremental construction heuristic. The complete solutions built in the early stages are not very good, but they are good enough to determine the next incremental step for the partial solution.

These notions are useful for our partial evaluation problem. Specifically, in SEAM, we see that the transient formation of many non-permanent groups guides the evolution of relatively rare permanent symbiotic associations - i.e. non-permanent joins guide permanent joins.

9.3.6 Stochastic lookahead avoids need for ‘backtracking’ in greedy optimisation

Following the analogy of search in the space of associations developed in 9.3.5 we can contrast the operation of stochastic lookahead and greedy approaches to finding good associations. Specifically, if we created new associations by selecting for any association that was better than no association (simple hill-climbing), or even by selecting for associations that were the best on average (steepest ascent), then this greedy approach would lead to sub-optimal associations. For example, the module ----1111 might provide a large fitness increase to the module --00----, perhaps the largest we can find at a given point in the search process, but it is not a correct join and making this join will prevent future joins that would have been higher fitness.

The classical AI approach to overcome this problem is to ‘backtrack’ and try a different path in the search space - in this case a different association. Evolution, of course, cannot backtrack - if it finds itself at a suboptimal configuration that it cannot escape from then that is the end of the story. However, we can see the operation of stochastic lookahead as providing a means to lookahead to avoid the need to backtrack. That is, by evaluating the fitness of both a proposed join and the non-joined components in many groups of individuals we are implicitly providing fitness feedback about the future consequences of the proposed join. We are not really making joins and backtracking, we are just evaluating a proposed join in many transient contexts. It just so happens that since these contexts are built using the same join operator over the same individuals, they provide an accurate reflection of the future consequences for a join. This has the effect of discounting the immediate/short-term benefit of a proposed join with the future/long-term consequences of a join. And this in turn means that a greedy optimisation in this modified fitness landscape coincides with an optimal optimisation. That is, the future-discounted fitness, the fitness coming from many group evaluations, modifies the fitness function for the individuals concerned such that they do not form the association that maximises their immediate selfish interest, but instead they maximise their long-term selfish interests. Since the association that is to their long term selfish interest is the cooperate/cooperate strategy the individuals make what appears to be cooperative associations but only because their selfish interests under the modified fitness coincide with the cooperative behaviour.

This use of 'future discounted' fitness is analogous to the use of lookahead in traditional game playing. In a board game, for example, heuristic search for a best move involves the use of a static evaluation function that assesses the value of static properties of the players pieces and their positions. A common technique in game strategies is to look ahead in the space of possible moves following after a proposed move to arrive at many possible future board states and apply the evaluation function to these board states instead of the immediate board state. Such lookahead is then used to assess the value of each proposed move. In SEAM, we do not explicitly assess the value of each possible association to find the best one – which would be unevolutionary – instead for each proposed join we stochastically assess whether there is any other join that would be better than this one and reject it if there is. This models the competition between a proposed composite and its free-living variants. But algorithmically, this is probably wasteful of evaluations and suggests a deterministic algorithm that may perform better.

When 'a little lookahead plus the fitness function' is better than 'the fitness function applied right now' (as it often is in game playing scenarios) then SEAM is better than greedy optimisation under the same representation and variation operators. The HIFF function shows exactly when this will be the case. Of course, this will not always be the case. Like any heuristic, filling-in the unspecified bits of a partial solution with other individuals and evaluating the full string is an imperfect heuristic for determining the value of a module, with or without lookahead. However, Lemma 1 (in Section 7.7.1), shows that, in principle, lookahead that covers all possible future joins is guaranteed to prevent sub-optimal joins in this class of problem. The question then becomes – how much lookahead does a given problem require to prevent sub-optimal joins? In separable problems, any join between correct modules makes a 'correct' join because there are no inter-module dependencies to be resolved. Thus no lookahead is required and a simple greedy method provides optimal solutions. In a problem containing arbitrarily high-order deceptive schemata, an arbitrarily large amount of lookahead is required to prevent sub-optimal joins. But in problems with modular interdependency, where the context sensitivity of a module is significant but limited, a small amount of lookahead is sufficient (See 7.7.2).

9.3.7 Pareto Coevolution

We mentioned briefly earlier that coevolution can be regarded as a multi-objective optimisation problem where each individual in a population represents a dimension to be optimised. Thus, desirable individuals are not those that do well on average against the other individuals in the population, but those that are non-dominated. Specifically, if A can beat everybody that B can beat (and at least one more) then A *dominates* B, and B may be discarded. But if A beats some opponents that B cannot beat, and B beats some opponents that A cannot beat, then A does not dominate B, and neither B (nor A) should be discarded on the basis of these comparisons. (See Ficici & Pollack 2000, Watson & Pollack 2000, Ficici & Pollack 2001, Noble & Watson 2001, De Jong & Pollack 2002, Bucci & Pollack 2002).

The idea here is that a coevolving entity should be retained because it is 'good for something' rather than merely 'good on average'. The intuition is that a selection scheme based on dominance will allow individuals to survive because they cover some part of the problem domain that other individuals do not. In contrast, selecting for individuals that are good on average is the 'greedy' approach that will cause the population to converge on the best solution found so far. Such a selection scheme is conceptually similar to fitness sharing methods, but here we abstract away all of the population dynamics inherent in fitness sharing. That is, rather than adjust the fitness of individuals according to how many other individuals are similar and then allow everybody to reproduce according to their adjusted fitness, we essentially transform fitness values into either 1 or 0 - either an individual is non-dominated, and it survives, or it is dominated, and it is discarded.[74] In this respect, our method is somewhat similar to implicit fitness sharing (Smith et al. 1993), except that in implicit fitness sharing it assumed that the problem sub-domains are given and in SEAM the problem sub-domains are defined by other coevolving individuals. Further, Parteo Coevolution and (genotypic) fitness sharing methods are different in that the latter often require a similarity metric in order to ascertain who is similar and with whom fitness must be shared. In Pareto Coevolution individuals are implicitly deemed similar if they perform similarly against different opponents.

The Pareto Coevolution idea is useful for explicitly adversarial coevolutionary scenarios (Ficici & Pollack 2001, Noble & Watson 2001), but in SEAM we are concerned with something that, on the face of it, is not a coevolutionary problem. We are merely trying to optimise a single function, HIFF. However, via the use of partially specified individuals that represent partial feature specifications, and templates built from other coevolving individuals that build a complete feature specification, we have transformed an evolutionary problem into a coevolutionary problem. Moreover, by the use of Pareto Coevolution, we have transformed a single objective optimisation problem into a multi-objective optimisation problem.

If we apply the Pareto Coevolution idea to the problem of finding good modules then the ‘dimensions’ of the problem are each represented by a different group - i.e. an individual is retained because there is some group it is good for. More exactly, a join between two individuals will be allowed if it creates a composite that dominates both parents. The SEAM model details the operation of this mechanism. This will work to find modules that cover the problem domain appropriately, and we will not have to specify where the modules are in the problem *a priori*, rather, the coevolution of modules will allow them to be discovered. Further, this means of promoting diversity does not rely of a genotypic similarity metric like Hamming distance. In essence, diverse modules are identified by virtue of having different performance profiles over

[74] See the Pareto hill-climber (Knowles & Corne 1999, 2000, Knowles et al. 2001), and the Multi-objective Messy GA (Van Veldhuizen & Lamont 2000).

a sample of groups. This is a purely ‘behavioural’ metric of similarity that does not depend on knowing an *a priori* measure of similarity, nor does it assume any access to the properties of the genotype directly.[75]

9.3.8 Relationship of SEAM to some other algorithmic methods

In summary, the partial evaluation problem is dealt with by the use of group evaluation, (in a manner sharing some properties with Messy GA templating, the Cooperative Coevolution shared domain model, and symbiotic scaffolding). The problem of maintaining diversity is dealt with by using Pareto Coevolution. And finally, the problem of string growth or bloat is really the problem of preventing selection for associations that appear immediately advantageous but are sub-optimal in the long-term, and this is dealt with by using stochastic lookahead.

[75] This is an interesting point that should be emphasised. In a ‘purist’ frame of mind, the only information we have about a candidate solution is its fitness. However, many methods used in EAs also assume access to genotypic information for the purposes of fitness sharing/diversity maintenance, restricted breeding etc. This seems like a necessary evil perhaps - we would rather that we did not have to examine the genotype directly, but how else could we modify the fitness of an entity appropriately? In SEAM we present an alternative. We do not examine the genotype of any individuals directly, and the only measure of a combination of features that we will use will be the objective fitness function. But nonetheless, we are able to ascertain important additional information about an individual and its characteristics. This is done not by examination of the genotype but by examining its effect on fitness measures in a sample of different contexts.

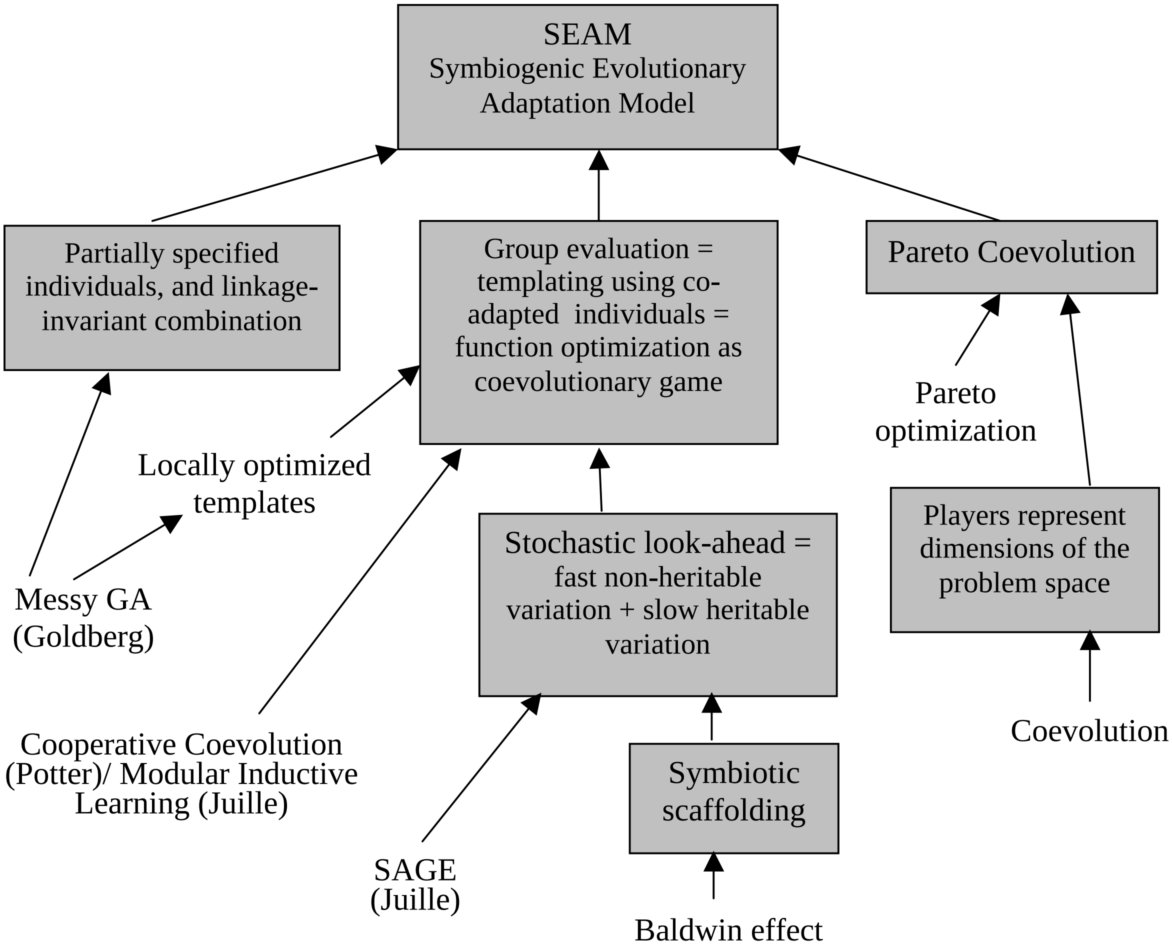

SEAM
Symbiogenic Evolutionary Adaptation Model
Partially specified individuals, and linkage-invariant combination
Group evaluation = templating using co-adapted individuals = function optimization as coevolutionary game
Pareto Coevolution
Pareto optimization
Locally optimized templates
Players represent dimensions of the problem space
Stochastic look-ahead = fast non-heritable variation + slow heritable variation
Messy GA (Goldberg)
Coevolution
Cooperative Coevolution (Potter)/ Modular Inductive Learning (Juille)
Symbiotic scaffolding
SAGE (Juille)
Baldwin effect

Figure 9-2: Relationship of SEAM to other EA methods

Shows relationships of algorithmic components – both existing work and new research. SEAM combines the use of partially specified individuals, with co-adapted templates and stochastic look-ahead, with Pareto coevolution.

9.4 Future research

9.4.1 Theoretical

There are a number of other theoretical works comparing mutation and crossover landscapes, for example (Culberson 1995, Spears 1992, Aizawa 1997, and Gitchoff & Wagner 1996). Now armed with the concept of modular interdependency and the proofs we have developed for compositional mechanisms on HIFF, it would be beneficial to return to these works and make a thorough examination of the relationships and implications. It seems likely that it would also be possible to do a formal comparison of the compositional mechanisms we have described with traditional divide and conquer methods such as dynamic programming.

Another avenue for future work is to continue to formalise a description of the class of problems for which compositional mechanisms are well-suited. To date we have focussed on a particular instance of this class, and a more general way to generate similar hierarchically decomposable problems in the same sub-class, but we could continue to elucidate the defining features of the class that are essential for the success of compositional mechanisms.

We would also like to develop and clarify the many relationships to other work we have outlined in the previous discussion. Not least:

- The relationship to renormalisation groups, Ising models and complex dynamical systems.
- Other notions of problem difficulty such as deception.
- The multi-dimensional treatment of fitness.
- Pareto coevolution as a general coevolutionary tool, as a selection scheme for cooperative coevolution, and as a diversity maintenance method.
- Formal analysis of ‘stochastic lookahead’ and its affordances.

9.4.2 Possible applications of SEAM

Another natural avenue of future work is to take the theoretic and principled illustrations given in this work and apply them to engineering problem domains (as well as biological ones). Preliminary investigations suggest that an algorithm like SEAM may work well on a particular sub-class of traditional hard combinatorial problems. For example, the subclass of Travelling Salesperson Problems where the distribution of delivery points is clustered at several scales into regions, districts, cities, for example. In this class, if the inter-cluster distance is large with respect to the intra cluster distance, then the number of possible sub-tours within a cluster that may be optimal (depending on how the remainder of the tour is completed) may be quite low but greater than one (see 4.3.1).

In general, SEAM can in principle be applied to any substrate that is amenable to a compositional construction of solutions. There is nothing in the algorithm that is dependent on, or makes use of the fact that, individuals are represented in bit strings. Neither is SEAM restricted to fixed length representations – components can be tested in many groups without the idea that the group provides a fully-specified template. Accordingly, future work may in principle exploit aspects of modularity and interdependency in any number of different substrates. For example:

- Graph partitioning (see Karypis & Kumar 1995, Karypis 1999, Toulouse et al. 1999).
- Neural networks (see Potter & De Jong 1995).
- Genetic Programming (See Rosca 1997, Juilles & Pollack 1996, O'Reilly 1997).
- Sorting networks (Juille 1999).
- Electronic circuit design (see Thompson 1998).
- Robot controllers and morphologies (see Sims 1994, Lipson & Pollack 2000).
- 3D sketch interpretation (Lipson 2002[76]).

Applying SEAM to these domains potentially provides the opportunity to identify functional modules in the domain as well as the optimisation of the given objective function.

9.4.3 Extensions of SEAM

There are a number of algorithmic aspects to SEAM that may need attention in order to be applied as an engineering optimisation method:

- **Relaxed stability criteria.** It may be the case that there is no join between a pair of components that is *always* in the interest of the two parties involved. In which case it may be necessary to relax the Pareto dominance criterion and allow joins which are 'nearly always good' or 'preferred in nearly all contexts' to become stable.

[76] SEAM has already been implemented with some modifications and applied to this domain (personal communication 2002). Further investigation is required.

- **Initialisation.** In the current version of SEAM, initialisation covers all the primitive components of the problem domain. It should be noted that the equality of components should be based on comparing the fitness effects they have on different groups of components, rather than on genotypic similarity, but nonetheless, in some domains, especially continuous substrates, it is not likely to be feasible to cover all primitive compoents by intialisation. Accordingly, it may be necessary to allow the creation of new primitive elements, perhaps accretively derived from existing elements. (See '*The diversification and integration of lineages'* in 8.3.1). Alternatively, it may be possible to define 'sub-primitives' that can be used to compose primitives – e.g. using aritmetic operators with only the values 1 and 0 to make real-valued constants (as in some genetic programming work).

- **Component re-use.** In the current version of SEAM, a primitive component is used only once. It seems likely that in general we will want to keep a component for re-use several times. If 'parent' components are retained in the population after a join in SEAM then it is still able to solve HIFF effectively, but not so quickly. It will be seen from 7.7.2 that retaining small individuals in the population disrupts the scale-invariant property of the algorithm. However, it may be useful pragmatically in some domains. An alternative might be to use a finite number of copies of each primitive in the initial population.

- **Discovering modules and dynamic landscapes.** In the current version of SEAM, structural information about component modules within a composite individual are not maintained in the individuals after a jopin is made. However, this information could easily be stored and output so that SEAM could be used as a method to provide information about the modular structure of a problem domain, rather than just optimise it. Additionally, SEAM's ability to maintain competing solutions to sub-problems using Pareto dominance selection, perhaps together with mechanisms to maintain internal module information, may be useful in providing adaptation in dynamic problem domains, and in exchanging information between one problem instance and another within a problem class.

- **Soft joins.** SEAM evaluates many groups of individuals in assessing the stability of a proposed join. If the join is eventually deemed unstable, then all the information from these evaluations is discarded. We know that it is not appropriate to keep a group just because it appears to give high fitness (this defeats the possibility of future compositions that may have found a higher fitness) but perhaps it may be possible to compromise in some circumstances – to use some of this information to bias future joins without excluding future exploration of alternatives. Specifically, it seems likely that a probabilistic form of association, or a 'strength' of association, might be better able to utilise the feedback from evaluations than the discrete all-or-nothing kind of joins used in the current version of SEAM. Interestingly, this begins to converge with algorithms such as BOA (Pelikan et al. 1999).

Chapter 10 - Summary and Conclusions

The intuitions that motivated this thesis work are quite straightforward. It has been observed by many that symbiosis is involved in some of the most important transitions in evolutionary history (Maynard Smith and Szathmary 1995, Buss 1987, Michod 1999, Sapp 1994). However, symbiosis is not generally incorporated in models of adaptation, and it is not properly understood how changes like the evolutionary transitions should be integrated with our common understanding of evolutionary change. The basic intuition is that if two coadapted symbionts are joined together into a new entity then this may be a means toward increases in complexity (e.g. Margulis 1993a). But it has not been at all clear whether there are any circumstances where entities evolved in this manner could not have been evolved otherwise, and what difference this might make to our understanding of evolutionary difficulty.

Meanwhile, in artificial evolution techniques, most implementations ignore the possibility of coadaptation altogether. Those that do utilise coevolution are generally concerned with explicitly adversarial scenarios and use purely competitive selective pressures (e.g. Hillis 1992). Some methods employ fitness sharing techniques to alleviate competitive exclusion but only a few models incorporate explicitly cooperative coevolution (e.g. Cooperative Coevolution, Potter 1997) where coadapted entities collectively cover a problem domain. However, there has been considerable interest in the idea of combining together partial solutions into more complete solutions via sexual recombination (e.g. Building Block Hypothesis, Holland 1975, 2000, Goldberg 1989). However, it has proved difficult to demonstrate the assembly of partial solutions into complete solutions in the Simple GA (e.g. Forrest and Mitchell 1993b). Alternative methods using more explicit representations of partial solutions have been developed (e.g. The Messy GA, Goldberg et al. 1989) but these generally assume that sub-problems are separable and have been unable to show a principled method of hierarchical assembly.

Cooperative Coevolution and the Messy GA exemplify two different complementary parts of the picture. In Cooperative Coevolution the emphasis is on the separation of roles to enable divide and conquer problem decomposition by explicit coadaptation to complementary parts of the problem domain. In the Messy GA the emphasis is on the assembly of partial solutions into whole solutions. Cooperative coevolution is missing the idea of assembling partial solutions into wholes recursively, and the Messy GA is missing the explicit notions of cooperative coadaptation that are required to prevent competitive exclusion of one type of specialist by a different type a specialist when roles are not predefined. Each provides essential parts of the model we develop in this thesis but neither of these approaches has a principled way to balance cooperative and competitive pressures to allow complementary specialists to coexist and at the same time also allow good composites to exclude sub-optimal composites.

A number of other techniques in evolutionary computation provide other important pieces of the puzzle. For example: In many evolutionary scenarios practitioners employ some kind of fitness sharing or crowding method to promote and maintain diversity; Multi-objective optimisation techniques provide tools for promoting a set of solutions each with a different balance of characteristics; And, mechanisms of automatic module acquisition introduce important ideas of encapsulation and re-use of sub-solutions. Each of these has their analogues in biological thought too: for example, frequency dependent fitness effects, niches defined by a different balance of characters, and symbiogenesis, respectively. However, in both disciplines the connections between these mechanisms has not been previously fully realised.

Meanwhile, the ideas of hierarchy and modularity are highly pervasive in our thoughts about biological and artificial systems, and Simon (1969) provides some concepts that have been very important in this work. Specifically, we have developed his intuitions about ‘nearly-decomposable systems’ into the more formal notion of ‘modular interdependency’, by clearly separating the idea of separability from decomposability. This has been critical in identifying the class of systems that are amenable to hierarchical problem decomposition or multi-level composition.

In the experiments presented in this work we have operationalised the intuitive notions coming from evolutionary biology into a working model using components from EC. In the process we make three main contributions to the computational work:

- First, we clarify the class of problems that could in principle be solved by compositional mechanisms but cannot be solved by accretive mechanisms. The principle insight here is the definition of modular interdependency where the features of a complex adaptation can be decomposed to reduce the dimensionality of the problem, but the resultant subparts are not separable and thus cannot be solved accretively.

- Second, we show clearly how sexual recombination in a form of genetic algorithm can, in some circumstances, enable the evolution of complex adaptations exhibiting modular interdependency that cannot be evolved using accretive mechanisms. We also show the dependency of this result on the conditions of genetic linkage.

- Third, we provide a new model, the Symbiogenic Evolutionary Adaptation Model, of an evolutionary process that solves this class of problem without the need for favourable gene ordering. This model combines the following features:

    a) A mechanism of composing sub-solutions together to find whole solutions: using partial specification and symbiotic encapsulation.

    b) A mechanism to ensure that entities co-adapt to cover complementary parts of the problem domain: using Pareto coevolution and the multi-dimensional treatment of fitness.

    c) A mechanism to determine the value of a proposed module in terms of its potential to make good assemblies (to distinguish between good modules and not-so-good sub-optimal associations): using group evaluation, and stochastic lookahead.

These elements together provide a principled way to balance cooperative and competitive pressures to allow complementary specialists to coexist and at the same time also allow coadapted specialists to join together into composites that may legitimately exclude sub-optimal composites. This model shows that the potential of compositional mechanisms to exploit modular interdependency in a problem domain is not dependent on biases coming from the assumption of favourable gene ordering. It also shows the assembly of partial-solutions into whole solutions through many hierarchical levels in a consistent, scale-invariant manner.

Returning to the biological observations that motivated these models, we have learned that there is a meaningful distinction to be made between compositional and accretive mechanisms of evolutionary change.

- We use the term ‘compositional’ to refer to evolutionary mechanisms that combine together systems or subsystems of genetic material that have been semi-independently pre-adapted in different lineages. Examples include sexual recombination (in subdivided populations), natural hybridization, horizontal gene transfer, and endosymbiosis.
- In contrast, we use the term ‘accretive’ to refer to mechanisms that accumulate random variations in genetic material, (i.e. the new genetic material introduced by such changes has not been pre-adapted elsewhere as a set). Examples of accretive mechanisms include genetic mutation, and sexual recombination (in unstructured populations).

These mechanisms enable effective adaptation in different classes of adaptive domains, and confer different understandings of evolvability and the class of landscapes in which continued adaptation is possible, different intuitions about what is evolutionary possible or likely, and conversely different intuitions what is unevolvable and unlikely. In short:

> Certain kinds of complex systems, considered unevolvable under normal accretive change, are, in principle and under certain circumstances, easily evolvable under compositional change.

In fact, in general, it is clear that different kinds of adaptive mechanism are well-suited to different classes of adaptive landscapes, but we have done more than simply show that accretive and compositional mechanisms are different. In supporting this claim in the course of this dissertation we have:

- Clarified the fundamental algorithmic distinction between accretive and compositional mechanisms by drawing analogies with different algorithmic paradigms - namely greedy optimisation methods such as hill-climbing, and divide and conquer problem decomposition, respectively. This helps us identify the class of complex systems to which each is well-suited.

- Described a class of complex systems based on modular interdependency that exemplifies the difference in the adaptive capacities of these mechanisms by utilising intuitive concepts of interdependency, modularity, and hierarchical modularity.

- Shown how a system possessing modular interdependency may exhibit all the usual characteristics of evolutionary difficulty: Creating a fitness landscape which is highly rugged possessing an exponential number of local optima; where the width of fitness saddles large with respect to the number of variables in the system; where high-fitness configurations of the system appear to be irreducibly complex - i.e. any small change is catastrophically deleterious; and finally, where there is no path of gradual changes conferring monotonically increasing fitness that approaches the optima in the landscape.

- And yet, we have illustrated a sufficient set of conditions under which such systems are easily evolvable via compositional mechanisms. First, we illustrated this using sexual recombination - investigating necessary conditions of population subdivision and genetic linkage. Second we illustrated a compositional mechanism based on symbiotic encapsulation - which requires appropriate diversity conditions provided by a particular selection model. However, the latter model, SEAM, shows that the adaptive advantage of compositional mechanisms does not, in principle, depend on *a priori* knowledge of genetic dependencies as is required to provide favourable gene ordering under sexual recombination.

In conclusion, the prevalent notions of evolutionary difficulty are dependent on the assumption of accretive change. Accordingly, in the presence of compositional mechanisms, the existence of a path of small genetic changes conferring monotonically increasing fitness is not necessarily required to explain a complex adaptation. Accordingly, when faced with the existence of a particular complex system in nature, perhaps possessing many complex interdependent parts, even those where any small change in the system causes the system to cease functioning, it is not strictly necessary to show that there exists a succession of proto-systems that are gradually increasing in function. If such a succession can be found, or reasonable hypotheses about the plausible existence of such a path can be upheld, then all well and good. But other explanations are possible.

Subject Index

A

accretive mechanism, 1, 4, 11, 15, 16, 17, 21, 22, 37, 38, 59, 63, 67, 113, 119, 124, 126, 127, 132, 133, 141, 142, 151, 155, 157, 162, 163, 164, 166, 217, 258, 260, 263, 264, 265, 266, 267, 273, 274, 304, 305, 306
additive effects, 13, 42, 47, 50, 51, 72, 118, 161, 261
aggregate effects, 13, 141, 152, 188, 270, 282
allele, 42, 44, 50, 65, 71, 74, 87, 88, 104, 107, 118, 161, 185, 191, 226, 247, 261, 278
allopolyploidy, 224, 271

B

backtracking, 293, 294
Baldwin effect, 98, 242, 261, 262, 263, 291, 292
bloat, 99, 237, 288, 290, 297
Building Block Hypothesis, 7, 24, 29, 30, 32, 35, 92, 115, 118, 168, 272, 276, 303

C

canalisation, 263
Classifier Systems, 93, 98, 100, 319
coevolution, 35, 80, 81, 106, 107, 108, 109, 115, 116, 117, 151, 276, 277, 295, 296, 298, 299, 303, 304, 305, 311, 317
competing conventions, 101, 105, 106, 275
competitive exclusion, 20, 75, 81, 107, 117, 176, 265, 303, 304
compositional evolution, 1, 6, 9, 11, 14, 23, 31, 34, 35, 37, 63, 218
compositional mechanisms, 2, 3, 4, 7, 9, 14, 17, 18, 20, 21, 22, 23, 24, 32, 33, 34, 36, 37, 48, 56, 57, 59, 63, 66, 67, 68, 71, 75, 96, 101, 115, 119, 126, 127, 132, 133, 136, 142, 143, 151, 161, 163, 164, 170, 217, 221, 222, 224, 239, 255, 256, 258, 259, 260, 263, 264, 265, 266, 267, 269, 271, 273, 298, 299, 304, 305, 306, 307
concatenated trap functions, 111, 147, 269, 284, 285
context sensitive, 71, 229
cooperative coevolution, 35, 81, 106, 107, 277, 299, 303
Cooperative Coevolution, 32, 93, 94, 95, 100, 105, 106, 107, 108, 117, 290, 291, 297, 303, 312, 318
credit assignment, 106, 107, 291

D

decomposition, 9, 14, 18, 24, 29, 32, 34, 35, 88, 90, 91, 92, 93, 94, 95, 96, 97, 99, 100, 106, 110, 115, 116, 119, 132, 151, 163, 164, 168, 192, 217, 246, 256, 279, 290, 304, 306
demes, 26, 72, 74, 176, 265
deterministic crowding, 170, 174, 177, 178, 179, 187, 191, 214, 217, 218, 239, 265, 275, 289
Deterministic Crowding, 174, 175, 176, 194, 196, 240
dimensional reduction, 134, 136, 151
divide-and-conquer problem decomposition, 24, 35, 92

E

encapsulation, 3, 4, 6, 7, 18, 20, 21, 22, 23, 25, 27, 28, 30, 31, 34, 37, 38, 55, 56, 57, 65, 66, 75, 93, 97, 99, 116, 167, 179, 219, 221, 223, 225, 226, 235, 236, 247, 248, 249, 259, 260, 261, 264, 266, 267, 287, 304, 305
endosymbiosis, 3, 4, 6, 7, 18, 20, 49, 53, 54, 55, 59, 221, 222, 306
epistasis, 11, 13, 23, 42, 44, 47, 52, 66, 71, 82, 117, 121, 122, 123, 129, 149, 151, 161, 266, 273, 278, 279
eukaryotes, 53, 55, 56, 271
evolutionary algorithm, 3, 7, 18, 25, 29, 35, 38, 77, 80, 84, 94, 107, 241, 242, 243, 287
Evolutionary Algorithm, 76, 78, 115, 311, 313, 315, 320
evolutionary computation, 2, 7, 10, 19, 20, 22, 26, 35, 116, 241, 304
Evolutionary Computation, 2, 28, 32, 38, 39, 76, 272, 312, 313, 315, 316, 317, 318, 319, 320, 321
evolvability, 2, 7, 14, 16, 23, 38, 83, 121, 146, 266, 267, 306, 313, 320

F

fitness peak, 44, 45, 47
fitness saddles, 15, 16, 44, 66, 116, 120, 124, 126, 129, 133, 141, 144, 148, 153, 155, 157, 158, 167, 191, 256, 258, 274, 281, 307
fitness sharing, 81, 99, 107, 108, 117, 177, 277, 289, 295, 297, 303, 304

G

gene duplication, 59, 61, 223, 260
genetic algorithm, 34, 78, 86, 96, 106, 111, 193, 305, 314, 315, 316, 317, 319
genetic linkage, 4, 5, 6, 18, 19, 28, 34, 35, 51, 52, 85, 99, 101, 102, 103, 113, 115, 116, 118, 162, 169, 170, 179, 181, 182, 184, 186, 187, 188, 192, 193, 194, 196, 218, 225, 227, 240, 256, 272, 305, 307
GIGA, 199, 203, 205, 211, 214
gradualism, 1, 2, 3, 40, 57, 58, 59, 181, 258, 260
greedy optimisation, 293, 294, 306
group evaluation, 236, 268, 276, 290, 291, 294, 297, 305

H

Hamming distance, 15, 43, 69, 108, 129, 132, 133, 154, 155, 157, 175, 193, 289, 296
heritability, 224
hierarchical cooperation, 149
hierarchical encapsulation, 18, 55, 259

Hierarchical-Equality, 143, 144, 145
Hierarchical-if-and-only-if, 119, 143, 145, 152, 168
hierarchically consistent, 29, 111, 114, 140, 141, 148, 152
hierarchy, 99, 100, 110, 111, 115, 138, 146, 148, 159, 163, 168, 188, 192, 248, 282, 304
hill-climbing, 8, 24, 35, 43, 44, 47, 64, 78, 89, 92, 203, 275, 293, 306, 317
hitch-hiking, 106, 107, 291
hopeful monsters, 41, 42, 47
horizontal gene transfer, 4, 23, 74, 97, 99, 224, 306

I

Idealised-SEAM, 245, 250
innovation, 54, 56, 59, 86, 116, 191, 242, 259, 260, 267, 317
interaction systems, 26, 50, 72, 73, 74, 118, 266
interspecific genetic integration, 49, 53
irreducibility, 47, 67, 75, 153
irreducibly complex, 15, 16, 22, 34, 46, 67, 159, 160, 161, 167, 307
Ising model, 280, 281, 282, 299

L

lateral gene transfer, 52
linkage learning, 31, 103, 192, 275, 288

M

macro-mutation, 24, 79, 83, 85, 148, 159, 162, 205, 275
major evolutionary transitions, 18, 22, 55, 56, 60, 75, 219, 264, 271
Major Transitions in Evolution, 55, 317
Messy GA, 93, 100, 102, 103, 104, 105, 106, 107, 117, 192, 226, 288, 289, 290, 296, 297, 303
modular interdependency, 9, 10, 11, 12, 13, 14, 16, 17, 18, 20, 21, 29, 32, 34, 37, 38, 59, 71, 113, 114, 126, 129, 139, 140, 141, 142, 143, 146, 147, 151, 152, 153, 156, 157, 168, 188, 217, 244, 256, 269, 270, 271, 273, 274, 279, 280, 281, 282, 283, 284, 288, 295, 298, 304, 305, 307
modularity, 10, 12, 14, 27, 29, 31, 68, 92, 93, 96, 110, 112, 115, 126, 134, 137, 244, 270, 271, 273, 280, 282, 300, 304, 307
module acquisition, 23, 25, 30, 35, 93, 100, 116, 275, 280, 304
moving-locus, 93, 102, 103, 104, 288
multiple-inheritance, 68

N

natural hybridization, 306
nearly decomposable, 12, 115, 132, 134, 282, 283, 284
NK landscapes, 83, 99, 112, 114, 274, 280
NKC landscapes, 31, 114, 147, 148, 269
non-random variation, 70
non-separable, 112, 114, 130, 197, 198, 204, 217

O

ontogenesis, 60
ontogenic, 59, 60, 61, 62, 269
open-ended, 35, 59, 96, 266, 267, 268
organelles, 53, 55

P

pairwise interactions, 12, 16, 124, 125, 126, 127, 130, 137, 139, 144, 148, 151
panmictic, 50, 72, 73, 101, 169, 170, 171, 172, 194, 196, 239, 240
parasites, 106, 107, 314
Pareto coevolution, 108, 109, 117, 298, 299, 305, 317
Pareto dominance, 31, 230, 231, 232, 234, 235, 236, 256, 265, 300, 301
partial specification, 30, 87, 99, 101, 103, 104, 228, 305
paths of small changes, 47, 260
population sub-division, 35, 169
premature convergence, 83, 107, 115
pseudocode, 37, 235, 246

R

random linkage, 145, 186, 212, 239, 275
recombinative hill-climber, 198, 205, 208, 211, 212, 214
renormalisation group, 280, 299
RMHC, 162, 164, 165, 166, 167, 194, 196, 234, 236, 240, 243, 278
Royal Roads, 91, 111, 126, 129, 147, 148, 269
ruggedness, 22, 44, 47, 66, 82, 83, 116, 119, 155, 258, 266

S

scale-invariant, 29, 146, 242, 244, 266, 270, 281, 282, 301, 305
schema disruption, 102, 276
Schema Theorem, 30, 87, 118, 276, 311
selfish gene, 27, 50, 268
Serial Endosymbiosis Theory, 7, 54, 259, 266
Shifting Balance Theory, 7, 26, 35, 50, 72, 118, 265
Simple GA, 78, 79, 81, 82, 83, 87, 88, 92, 99, 107, 170, 173, 272, 303
single inheritance, 159, 167
steady-state, 175, 319
stochastic lookahead, 263, 292, 293, 294, 297, 299, 305
subdivided populations, 6, 306
successive slight modifications, 4, 41, 47, 48, 57
symbiogenesis, 20, 22, 23, 28, 32, 33, 34, 35, 37, 53, 54, 55, 96, 221, 271, 304
Symbiogenic Evolutionary Adaptation Model, 20, 219, 233, 240, 305
symbiosis, 32, 35, 49, 53, 56, 57, 58, 96, 98, 99, 219, 228, 259, 292, 303, 317
symbiotic encapsulation, 1, 4, 6, 18, 20, 21, 22, 23, 25, 27, 30, 37, 38, 49, 56, 57, 65, 66, 75, 99, 167, 179,

219, 221, 223, 225, 236, 259, 260, 261, 264, 266, 287, 305, 307
symbiotic groups, 6, 18, 263
symbiotic scaffolding, 261, 262, 263, 271, 292, 297

T

template, 106, 290, 291, 300
templating, 108, 290, 297
tight linkage, 19, 102, 113, 194, 206, 239, 240, 256, 275, 279

U

uniform crossover, 84, 85, 170, 182, 183, 184, 185, 186, 187, 190, 191, 192, 194, 196, 218, 240, 278
unstructured populations, 306

References

Aizawa, A, 1997, “Fitness Landscape Characterization by Variance of Decompositions”, in *Foundation of Genetic Algorithms 4*, Belew RK, & Vose MD, eds., Morgan Kaufmann, San Francisco. p.225-245.

Altenberg, L, 1995, “The Schema Theorem and Price’s Theorem”, in *Foundations of Genetic Algorithms 3*, Whitley D, & Vose MD, eds., Morgan Kaufmann, San Francisco, pp. 23-49.

Angeline, PJ, & Pollack, JB, 1993, “Evolutionary Module Acquisition”, *Procs. of The Second Annual Conference on Evolutionary Programming*, Fogel, DB, and Atmar, W, eds., Evolutionary Programming Society, San Diego CA, pp. 154-163.

Arthur, W, 1987, *The niche in competition and evolution*. Wiley and Sons, Chichester, NY.

Arthur, W, 1997, *The Origin of Animal Body Plans: A Study in Evolutionary Developmental Biology*, Cambridge University Press.

Back, T, Hoffmeister, F & Schwefel HP, 1991, “A Survey of Evolutionary Strategies”, in *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Inc.

Bak, P, 1996, *How nature works: the science of self-organized criticality*, NY, USA: Copernicus.

Baldwin, JM, 1896, “A New Factor in Evolution”, *American Naturalist*, 30, 441-451.

Barricelli, NA, 1957, “Symbiogenetic Evolution Processes Realised by Artificial Methods”, *Methodos*, vol. 9, pp. 143-182.

Barry, A, 1996, “Hierarchy Formulation Within Classifiers System -- A Review”, in *Proceedings of the First International Conference on Evolutionary Algorithms and their Application* EVCA'96, Moscow, 1996, Goodman et al. eds., The Presidium of the Russian Academy of Sciences, pp. 195-211.

Barton, NH, 1993, “The Probability of Fixation of a Favoured Allele in a Subdivided Population”, *Genet. Res*. 62, pp 149-157, Cambridge University Press.

Behe, MJ, 1996, *Darwin's black box: the biochemical challenge to evolution*. The Free Press: New York.

Bermudes, D, & Margulis, L, 1985, “Symbiosis as a Mechanism of Evolution: Status of the Symbiosis Theory”, *Symbiosis* 1: 101-124.

Bucci, A, & Pollack, JB, 2002, “Order-theoretic analysis of coevolution problems: Coevolutionary statics,” in *Procs. of Understanding coevolution: theory and analysis of coevolutionary algorithms Workshop*, (in GECCO 2002), to appear.

Bull, LW, & Fogarty T, 1995, “Artificial Symbiogenesis”, *Artificial Life 2*(3): 269-292.

Bull, LW, & Fogarty, TC, 1996, “Horizontal Gene Transfer in Endosymbiosis”, in Langton CG, & Shimohara T, eds., *Artificial Life V*, MIT Press, pp. 77-84.

Bull, LW, 1997, “On the Evolution of Eukaryotes: Computational Models of Symbiogenesis and Multicellularity”, In *Mathematical and Computational Biology: Hierarchical Complexity and Digital Evolution*, Nehaniv CH, ed., American Mathematical Society, pp. 31-46.

Bull, LW, 1999, “On the Evolution of Multicellularity and Eusociality”, *Artificial Life 5*(1):1-15.

Bull, LW, Fogarty, TC, & Pipe, AG, 1995, “Artificial Endosymbiosis”, in *Advances in Artificial Life - Proceedings of the Third European Conference on Artificial Life*, Moran, F, Mereno, A, Merelo JJ, & Chaon, P, eds., Springer Verlag, pp. 273-289 .

Buss, LW, 1987, *The Evolution of Individuality*, Princeton University Press, New Jersey.

Cavicchio, DJ, 1970, *Adaptive Search Using Simulated Evolution*, Ph.D.Thesis, University of Michigan, Ann Arbor, MI.

Chen, S, 1999, “Is the Common Good? A New Perspective Developed in Genetic Algorithms”, PhD diss., Robotics Institute, Carnegie Mellon University.

Cormen, TH, Leiserson, CE, Rivest, RL, 1991 *Introduction to Algorithms*, MIT Press, Camb. MA, McGraw-Hill Book Company, NY.

Coyne, JA, Barton, NH, and Turelli, M, 2000, “Is Wright’s Shifting Balance Process Important in Evolution”, *Evolution* 54(1), pp.306-317.

Culberson, JC, 1992, “Genetic Invariance: A New Paradigm for Genetic Algorithm Design”, technical report TR92-02, University of Alberta.

Culberson, JC, 1995, “Mutation-Crossover Isomorphisms and the Construction of Discriminating Functions”, *Evolutionary Computation*, 2, pp. 279-311.

Darwin, C, 1859, *On the Origin of Species by means of natural selection*, John Murray, London.

Dasgupta, D, & Forrest, S, 1999, “Artificial immune systems in industrial applications”, *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials*. IPMM'99, Honolulu, HI (July 10-14, 1999).

Davies, J, 1996, “Origins and evolution of antibiotic resistance”, *Microbiologia* 12:9-16.

Dawkins, R, 1976, *The Selfish Gene*, Oxford University Press, NY.

Dawkins, R, 1996, *Climbing Mount Improbable*, New York: WW Norton & Company.

De Jong, ED, & Pollack, JB, 2002, (working title “Beyond Competition”), in preparation.

De Jong, ED, Watson, RA, & Pollack, JB, 2001, “Reducing Bloat and Promoting Diversity using Multi-Objective Methods”, in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO-2001, Spector, LE, Goodman, Wu, A, Langdon, WB, Voigt, HM, Gen, M, Sen, S, Dorigo, M, Pezeshk, S, Garzon, M & Burke, E, eds., Morgan Kaufmann. pp. 11-18.

De Jong, KA, 1975, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD thesis, University of Michigan, Ann Arbor, MI.

De Jong, KA, 1988 “Using Genetic Algorithms to Learn Task Programs: the Pitt Approach”, *Machine Learning*, 3, 2-3 (1988).

De Jong, KA, 2002, *Evolutionary Computation: A Unified Approach*, MIT Press.

De Jong, KA, & Potter MA, 1995, “Evolving Complex Structures via Cooperative Coevolution”, In *Proceedings of the Fourth Annual Conference on Evolutionary Programming,* McDonnell, JR, Reynolds RG, & Fogel DB, eds., MIT Press, pp. 307-317.

De Jong, KA, & Spears, W, 1990, “An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms”, in Proceedings of the First Int’l Conf. on Parallel Problem Solving from Nature, Dortmund, Germany, October 1990. Springer Verlag. pp. 38-47.

Deb, K, & Goldberg, DE, 1989, “An investigation of Niche and Species Formation in genetic Function Optimization”, in *Proceedings of the Third International Conference on Genetic Algorithms,* (ICGA3), Schaffer JD, ed., San Mateo California: Morgan Kaufmann. pp. 42-50.

Deb, K, & Goldberg, DE, 1992a, “Analyzing Deception in Trap Functions”, in Whitley, D, ed., *Foundations of Genetic Algorithms 2*, Morgan Kaufmann, San Mateo, CA. pp. 93-108.

Deb, K, & Goldberg, DE, 1992b, “Sufficient conditions for deceptive and easy binary functions”, *Annals of Mathematics and Artificial Intelligence 10*, pp. 385-408. (also IlliGAL Report No. 91009), University of Illinois, IL.

Demur, R, 1995, “Evolution Through Cooperation: The Symbiotic Algorithm”, *Procs. of Artificial Evolution Euro. Conf. 1995 (AE95)*.

Doolittle, WF, 2000. “Uprooting the tree of life”, *Scientific American*, 282:90-95.

Dorigo, M, Di Caro, G. & Gambardella LM, 1999, “Ant algorithms for discrete optimization”, *Artificial Life*, vol. 5. no. 2, pp. 137-172.

Ebner, M, Shackleton, M, Shipman, R, 2001, “How neutral networks influence evolvability”, *Complexity* 7(2), pp. 19-33, Wiley Periodicals.

Ebner, M, Watson, RA, and Alexander, J, 2000, “Co-evolutionary dynamics on a deformable landscape”, in *Proceedings of the 2000 Congress on Evolutionary Computation*, San Diego Marriott Hotel, La Jolla, CA, Volume 2, pp. 1284-1291, IEEE Press, 2000.

Eigen, M, & Schuster, P, 1979, “The Hypercycle: A principle of Natural Self-Organisation”, Springer, Berlin.

Eshelman, LJ, Mathias, KE, & Schaffer, JD. 1996, “Convergence controlled variation”, in *Foundations of Genetic Algorithms 4*, Belew, RK, & Vose MD, eds., Morgan Kauffman, San Mateo, CA., pp. 203-224

Ficici, SG, PhD thesis, Brandeis University, in preparation.

Ficici, SG, and Pollack, JB, 2000, “A Game-Theoretic Approach to the Simple Coevolutionary Algorithm”, *Parallel Problem Solving from Nature VI*, Schoenauer, M, et al, eds., Springer Verlag.

Ficici, SG, and Pollack, JB, 2001, “Pareto Optimality in Coevolutionary Learning”, *Sixth European Conference on Artificial Life*, Kelemen, J, ed., Springer, 2001.

Fisher, RA, 1930. *The genetical theory of natural selection*. Dover publisher, Inc., New York.

Fogel, LJ, Owens, AJ, & Walsh, MJ, 1966, *Artificial Intelligence through Simulated Evolution*, New York: Wiley.

Fonseca, CM, Fleming PJ, 1995, “An Overview of Evolutionary Algorithms in Multiobjective Optimization”, *Evolutionary Computation*, Vol. 3, No.1, Massachussets, MA: MIT-Press, pp.1-16.

Forrest, S, & Mitchell, M, 1993a, “Relative Building block fitness and the Building block Hypothesis”, in *Foundations of Genetic Algorithms 2*, Whitley, D, ed., Morgan Kaufmann, San Mateo, CA.

Forrest, S, & Mitchell, M, 1993b, “What makes a problem hard for a Genetic Algorithm? Some anomalous results and their explanation” *Machine Learning 13*, pp.285-319.

Frank, SA, 1997, “Models of Symbiosis”, *American Naturalist*, 150, p80-99

Gitchoff, P, and Wagner, GP, 1996, “Recombination induced hypergraphs: a new approach to mutation-recombination isomorphism”, *Complexity* 2: pp. 37-43.

Goldberg, DE, & Horn, J, 1994, “Genetic Algorithm Difficulty and the Modality of Fitness Landscapes”, in *Foundations of Genetic Algorithms 3*, Whitley D, ed., Morgan Kaufmann, San Mateo, CA.

Goldberg, DE, 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading Massachusetts, Addison-Wesley.

Goldberg, DE, & J. Richardson, 1987, “Genetic Algorithms with Sharing for Multimodal Function Optimization”, *Genetic Algorithms and Their Applications: Proc. of the 2nd International Conference on GAs*, July 28-31, MIT, pp. 41--49.

Goldberg, DE, Deb K, & Korb B, 1990, “Messy Genetic Algorithms Revisited: Studies in Mixed Size and Scale”, *Complex Systems*, 4, pp. 415-444.

Goldberg, DE, Deb, K, Kargupta, H & Harik, G, 1993, “Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms”, IlliGAL Report No. 93004, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, Il.

Goldberg, DE, Korb, B, & Deb K, 1989, “Messy Genetic Algorithms: Motivation, Analysis and first results”, *Complex Systems*, 3, 493-530.

Goldschmidt, R, 1940, *The material basis of evolution*, New Haven, Yale University Press.

Gordon, R, 1993, “The Fractal Physics of Biological Evolution”, in *Dynamical Phenomena at Interfaces, Surfaces and Membranes*, Beysens, D, Boccara, N, & Forgacs, G, eds. Commack, N.Y.: NOVA Science Publishers, 99-111.

Gould, SJ & Vrba, E, 1982, “Exaptation—a missing term in the science of form”, *Paleobiology*, **8**: 4-15. Ithaca.

Gould, SJ, 1982a, “The meaning of punctuated equilibrium and its role in validating a hierarchical approach to macro evolution” in *Perspectives on Evolution*, Milkman, R, ed. Sinauer Associates Inc. MA. pp. 83-104.

Gould, SJ, 1982b, “The uses of heresy: An introduction to R. Goldschmidt’s: The Material Basis of Evolution”, Yale University Press, New Haven CT. pp. xiii-xlii.

Gould, SJ, 1985, *The flamingoes smile*, WW Norton Company, NY.

Grieshaber, MK, Völkel, S, 1998, “Animal Adaptations for Tolerance and Exploitation of Poisonous Sulfide”, *Ann. Rev. Physiol.*, 60: 33-53.

Gruau, F, 1994, *Neural Network Synthesis Using Cellular Encoding and the Genetic Algorithm*. PhD thesis, Ecole Normale Superieure de Lyon.

Harik, GR, & Goldberg, DE, 1996, “Learning Linkage”, *Foundations of Genetic Algorithms 4*, Belew, RK, & Vose, MD, eds., Morgan Kaufmann, San Mateo CA., 1997. pp. 247-262.

Harik, GR, 1997, *Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty Using Genetic Algorithms*, PhD dissertation, Dept. of Comp. Science and Engineering, Uni. of Michigan.

Hartwell, L, Hopfield, J, Leibler, S, & Murray, A, 1999, “From Molecular to Modular Cell Biology”, *Nature 402* (supp): p. C47-C52.

Harvey, I, 1993, “The Puzzle of the Persistent Question Marks: a Case Study of Genetic Drift,” in S. Forrest (ed.), *Genetic Algorithms: Proc. of the Fifth Intl. Conference on Genetic Algorithms*, San Mateo CA, Morgan Kaufmann, pp. 15-22.

Harvey, I, 1996, *The microbial genetic algorithm*, unpublished work.

Hillis, WD, 1992, “Coevolving parasites improve simulated evolution as an optimization procedure”, In Langton, CG, Taylor, C, Farmer JD, and Rasmussen, S, eds., *Artificial Life II*, Addison-Wesley, Reading, MA. pp. 313--324.

Hinton, GE, & Nowlan, SJ, 1987, “How Learning Can Guide Evolution,” *Complex Systems*, 1, 495-502.

Hirasawa, K, Hu J, Murata J, & Mao, J, 2001, “Genetic Symbiosis Algorithm”, *Machine Intelligence and Robotic Control*, Volume 3, No. 1 / March 2001, pp.27-34.

Höhn, C, & Reeves C, 1996, “Are long path problems hard for genetic algorithms?”, in *Proceedings of the 4th Conference on Parallel Problems Solving from Nature* (PPSN IV), volume 1141 of LNCS, Voigt, HM, Ebeling, W, Rechenberg, I, Schwefel, HP, eds., Springer Verlag, pp. 134-143.

Holland, JH, & Reitman, JS, 1978, “Cognitive systems based on adaptive algorithms”, in *Pattern Directed Inference Systems*, Waterman, DA, & Hayes-Roth, F, eds., New York: Academic Press, pp. 313-329.

Holland, JH, 1975 *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: The University of Michigan Press.

Holland, JH, 1993 “Royal Road Functions”, Internet Genetic Algorithms Digest v7n22.

Holland, JH, 2000, “Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions”, E*volutionary Computation* 8(4): 373-391.

Horn, J, Nafpliotis, N, Goldberg, DE, 1994, “A niched Pareto genetic algorithm for multiobjective optimization”, in *Proceedings of the First IEEE Conference on Evolutionary Computation IEEE World Congress on Computational Intelligence*, (Piscataway, NJ.), Volume 1, pages 67-72.

Hornby, G, 2002, PhD Thesis, in preparation.

Husbands, P, 1993, “An ecosystems model for integrated production planning”, *International Journal of Computer Integrated Manufacturing*, 6(1-2):74-86.

Hutchinson, GE, 1965, *The ecological theatre and the evolutionary play*, Yale University Press, New Haven.

Huynen, MA, Stadler, PF & Fontana, W, 1996, “Smoothness Within Ruggedness: The Role of Neutrality in Adaptation.” *Proc. Natl. Acad. Sci. (USA),* 93: pp.397-401.

Jansen, T, & Wegener I, 2001, “Real Royal Road Functions - Where Crossover Provably is Essential” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO-2001, Spector, LE, Goodman, Wu, A, Langdon, WB, Voigt, HM, Gen, M, Sen, S, Dorigo, M, Pezeshk, S, Garzon, M & Burke, E, eds., Morgan Kaufmann. 1034--1041.

Jones, T, 1995, *Evolutionary Algorithms, Fitness Landscapes and Search*, PhD dissertation, 95-05-048, University of New Mexico, Albuquerque.

Juillé, H, & Pollack, JB, 1996, “Coevolving Intertwined Spirals”, *Procs. of fifth annual conference on Evolutionary Programming*, MIT press, pp. 461-468.

Juillé, H, & Pollack, JB, 1998, “Coevolving the "Ideal" Trainer: Application to the Discovery of Cellular Automata Rules”, *Proceedings of the Third Annual Genetic Programming Conference,* Madison, Wisconsin, July 22 - 25, 1998.

Juillé, H, 1999, *Methods for Statistical Inference: Extending the Evolutionary Computation Paradigm*. Doctoral Dissertation, Brandeis University, Department of Computer Science, May 1999.

Kargupta, H, 1997, “Gene Expression: The Missing Link In Evolutionary Computation”, *Genetic Algorithms in Engineering and Computer Science*, Quagliarella, Q, Periaux, J, & Winter, G, (eds.) John Wiley and Sons.

Karypis, G, 1999, *Multilevel Algorithms for Multi-Constraint Hypergraph Partitioning*, University of Minnesota, Department of Computer Science, Technical Report #99–034.

Karypis, G, Kumar, V, 1995, *A fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs*, University of Minnesota, Department of Computer Science, Technical Report #95-035.

Kauffman, SA, & Johnsen, S, 1989, “Coevolution to the Edge of Chaos: Coupled Fitness Landscapes, Poised States and Coevolutionary Avalanches”, *Artificial Life II*, Addison Wesley, pp 325-370.

Kauffman, SA, 1986 “Autocatalytic sets of proteins”, *Journal of theoretical biology* 119: 1-24.

Kauffman, SA, 1989, “Adaptation on Rugged Fitness Landscapes”, *Lectures in the Sciences of Complexity*, Vol 11, 527-618. Addison Wesley.

Kauffman, SA, 1993, *The Origins of Order*, Oxford University Press.

Khakhina, LN, 1992 *Concepts of Symbiogenesis: Historical and Critical Study of the Research of Russian botanists*, eds. Margulis L, McMenamin M, (translated by Merkel S, Coalston R), Yale University Press.

Kimura, M, 1983, *The Neutral Theory of Molecular Evolution*, Cambridge University Press.

Knowles, JD, & Watson, RA, 2002, “On the utility of Redundant Encodings in Mutation-based Evolutionary Search”, under submission.

Knowles, JD, & Corne, DW, 1999, “The Pareto Archived Evolution Strategy : A New Baseline Algorithm for Pareto Multiobjective Optimisation”, in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, Volume 1, pp. 98-105.

Knowles, JD, & Corne, DW, 2000, “Approximating the nondominated front using the Pareto Archived Evolution Strategy”, *Evolutionary Computation*, 8(2), pp. 149-172.

Knowles, JD, Watson, RA, & Corne DW, 2001, “Reducing Local Optima in Single-Objective Problems by Multi-objectivization”, in *Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization (EMO'01)*, pp. 269--283, Springer-Verlag.

Koza, JR, 1992, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: MIT Press.

Koza, JR, 1994, *Genetic Programming II: Automatic Discovery of Reusable Programs,* Cambridge, MA: MIT Press.

Kozo-Polyansky, BM, 1921, “The Theory of Symbiogenesis and Pangenesis, Provisional Hypothesis” in Journal of the first All-Russian Congress of Russian Botanists in Petrograd in 1921. Peterograd (in Russian, see Khakhina 1992).

Kvasnicka, V, 2000, “An evolutionary model of symbiosis”, in *Quo Vadis Computational Intelligence?* Sincak, P, Vascak, J, (eds.), Physica-Verlag, Heidelberg, 2000, pp. 293-304.

Lawrence, JG, & Ochman, H, 1998, “Molecular archaeology of the *Escherichia coli* genome”, *Proc. Natl. Acad. Sci*., USA 95:9413-9417.

Levins, R, & Lewontin, RC, 1985, *The Dialectical Biologist,* Harvard University Press.

Lewontin, RC, 1983, “The organism as the subject and object of evolution”, *Scientia*, 118, 65-82.

Lewontin, RC, 1982, “Organism & environment”, in *Learning, development, culture,* H. Plotkin (ed.), New York: John Wiley, pp. 151-170.

Lewontin, RC, 2000 *The Triple Helix: Gene, Organism and Environment,* Harvard U. Press, Camb., MA.

Lipson, H, & Pollack, JB, 2000, “Automatic Design and Manufacture of Artificial Lifeforms”, *Nature* 406, pp. 974-978.

Louis SJ, & Rawlins GJE, 1991, “Designer Genetic Algorithms: Genetic Algorithms in Structure Design”, in *Procs. of fourth international conference on genetic algorithms*, Morgan Kauffman, pp. 53-60.

MacArthur, RH, 1968 “The theory of the niche”, in *Population Biology and Evolution,* Lewontin RC (ed.) Syracuse University Press, New York.

Mahfoud, S, 1995, “*Niching Methods for Genetic Algorithms*”, PhD thesis, University of Illinois. (also IlliGAl Report No. 95001)

Mandelbrot, 1982, *The fractal geometry of Nature*, Freeman, San Francisco.

Mao, J, Hirasawa, K, Hu, J, and Murata, J, 2001, “Genetic Symbiosis Algorithm for Multiobjective Optimization Problems”, *Trans. of the Society of Instrument and Control Engineering,* Vol.37, No.9, pp.894-901. (also GECCO 2001 late breaking papers, pp. 267-274.)

Margulis, L, 1970, *Origin of eukaryotic cell,* Yale University Press, New Haven.

Margulis, L, 1993a, *Symbiosis in cell evolution: Microbial Communities in the Archean and Ptroterozoic Eons*. 2d ed., WH Freeman and Co. New York.

Margulis, L, 1993b, “Origins of Species: Acquired Genomes and Individuality”, *BioSystems*, Vol. 31 (2-3) pp. 121-125.

Margulis, L, 1995 “Gaia Is a Tough Bitch”, in *The Third Culture*”, Brockman, J, Simon and Schuster, New York.

Margulis, L, & Fester, R, (eds.) 1991, *Symbiosis as a Source of Evolutionary Innovation,* MIT Press.

Mayley, G, 1996, “Landscapes, Learning Costs and Genetic Assimilation,” in *Evolution, Learning, and Instinct: 100 Years of the Baldwin Effect, special issue of Evolutionary Computation*, Vol 4, No. 3; P.Turney, D. Whitley and R. Anderson (eds).

Maynard Smith, JM, 1982, *Evolution and the Theory of Games.* Cambridge University Press, Cambridge.

Maynard Smith, JM, 1991, “A Darwinian view of symbiosis”, in *Symbiosis as a Source of Evolutionary Innovation*, Margulis, L, & Fester, R, (eds.) pp. 26-39. MIT Press, Cambridge, MA, 1991.

Maynard Smith, JM & Szathmary, E, 1995, *The Major Transitions in Evolution*, WH Freeman.

Maynard Smith, JM, & Szathmary, E, 1993, “The origin of chromosomes I. Selection of linkage”, *Journal of Theoretical Biology* **164**: 437-66.

Mazodier, P, & Davies, J, 1991, “Gene transfer between distantly related bacteria”, *Annual Review of Genetics* 25:147-171.

Merezhkovsky, KS, 1909, “The Theory of Two Plasms as the Basis of Symbiogenesis, a New Study or the Origins of Organisms,” *Proceedings of the Studies of the Imperial Kazan University*, Publishing Office of the Imperial University. (In Russian, see Khakhina 1992).

Michod, RE, 1999, *Darwinian Dynamics, Evolutionary Transitions in Fitness and Individuality*. Princeton Univ. Press.

Mitchell, M, 1996, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA.

Mitchell, M, Forrest, S, and Holland, JH, 1992, “The royal road for genetic algorithms: Fitness landscapes and GA performance”, in *Toward a practice of autonomous systems, Proceedings of the First European Conference on Artificial Life*. Cambridge, MA: MIT Press/Bradford Books, pp. 245-54.

Mitchell, M, Holland, JH, & Forrest, S, 1995 “When will a Genetic Algorithm Outperform Hill-climbing?” *Advances in Neural Information Processing Systems*, 6:51--58, Morgan Kaufmann, San Mateo, CA.

Moriarty, DE, 1997, *Symbiotic Evolution of Neural Networks in Sequential Decision Tasks*, PhD thesis, University of Texas at Austin, USA.

Muhlenbein, H, 1992, “How genetic algorithms really work: I. mutation and hill-climbing.” In *Parallel Problem Solving from Nature 2*, Manner, R & Manderick, B, (eds), pp. 15--25, Amsterdam. Elsevier.

Nix, AE, & Vose MD, 1992, “Modeling genetic algorithms with markov chains”, *Annals of Mathematics and AI*, 5:79--88.

Noble, J, & Watson RA, 2001, “Pareto coevolution: Using performance against coevolved opponents in a game as dimensions for Pareto selection”, in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO-2001, Spector, LE, Goodman, Wu, A, Langdon, WB, Voigt, HM, Gen, M, Sen, S, Dorigo, M, Pezeshk, S, Garzon, M & Burke, E, eds., Morgan Kaufmann. pp. 493--500.

Nowak, M, & Schuster, P, 1989, “Error Thresholds of Replication in Finite Populations, Mutation Frequencies and the Onset of Muller's Ratchet”, *J. Theor. Biol.*, 137:375-395.

Numaoka, C, 1995, “Symbiosis and Coevolution in Animats”, in *Proceedings of the Third European Conference on Artificial Life* (ECAL'95). Springer-Verlag.

O'Reilly, U, 1997, “The impact of external dependency in Genetic Programming Primitives”, in *Emerging Technologies 1997*, pp. 45-58. University College London.

Oates M, & Corne, D, 2001, “Overcoming Fitness Barriers in Multi-modal Search Spaces”, *Foundations of Genetic Algorithms VI,* (2000) Martin, WN, & Spears, WM, (eds.), Morgan Kaufmann, pp. 37-50.

Ochman, H, Lawrence, JG, & Groisman, EA, 1998, “Lateral gene transfer and the nature of bacterial innovation”, *Nature* 405:299-304.

Ohno, S, 1970, *Evolution by Gene Duplication,* New York: Springer-Verlag.

Ohta, T, 1968, “Effect of Initial Linkage Disequalibrium and Epistasis on Fixation Probability in a Small Population, with two Segregating Loci”, *Theoretical and Applied Genetics* 38, pp 234-248.

Paley, W, 1802, *Natural theology, or evidences of the existence and attributes of the deity collected from the appearances of nature*. London

Pelikan, M, & Goldberg DE, 2001, “Escaping Hierarchical Traps with Competent Genetic Algorithms” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO-2001, Spector, LE, Goodman, Wu, A, Langdon, WB, Voigt, HM, Gen, M, Sen, S, Dorigo, M, Pezeshk, S, Garzon, M & Burke, E, eds., Morgan Kaufmann. pp. 511-518.

Pelikan, M, Goldberg DE, & Cantu-Paz, E, 1999, “BOA: The Bayesian Optimization Algorithm”, in *Procs of GECCO 99*, Vol 1, Banzhaf W et al (eds.), Orland FL: Morgan Kaufmann, San Fransisco, CA pp.525-532.

Pelikan, M, & Goldberg, DE, 2000, “Hierarchical Problem Solving by the Bayesian Optimization Algorithm”, IlliGAL Report No. 2000002, *Proceedings of GECCO-2000*, Whitley, D., et al (eds.), Morgan Kaufmann. pp. 267-274.

Pollack, JB & Blair, AD, 1998, “Co-Evolution in the Successful Learning of Backgammon Strategy”, *Machine Learning* 32(3): 225-240, September 1998.

Potter, MA, & De Jong KA, 1994, “A Cooperative Coevolutionary Approach to Function Optimization”, in *Proceedings of the Third Conference on Parallel Problem Solving from Nature*, pp. 249-257. Springer-Verlag.

Potter, MA, & De Jong KA, 1995, “Evolving Neural Networks with Collaborative Species”, in *Proceedings of the 1995 Summer Computer Simulation Conference*, pp. 340-345. The Society for Computer Simulation.

Potter, MA, & De Jong KA, 2000, “Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents”, *Evolutionary Computation*, 8(1), pp. 1-29. MIT Press.

Potter, MA, 1997, *The Design and Analysis of a Computational Model of Cooperative Coevolution*, PhD thesis, George Mason University, Fairfax, Virginia.

Powell, MA, & Somero, GN, 1986, “Hydrogen-Sulfide Oxidation is Coupled to Oxidative-Phosphorilation in Mitochondria of Solemya reidi”, *Science*, 233: 563-566.

Radcliffe, NJ, 1991, “Forma Analysis and Random respectful Recombination”, in *Proc. Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, 1991.

Reidys, C, Stadler, P, & Schuster, P, “Generic properties of combinatory maps - Neutral networks of RNA secondary structures”, *Bull. Math. Biol.* 59:339-397, 1997.

Reynolds, CW, 1994, “Competition, Coevolution and the Game of Tag”, in *Proceedings of Artificial Life IV*, Brooks, R, & Maes, P, (eds.), MIT Press, Cambridge, Massachusetts, pp 59-69.

Ridley, M, 1985, *The Problems of Evolution*, Oxford UK: Oxford University Press

Rosca, JP, 1997, *Hierarchical Learning with Procedural Abstraction Mechanisms*, Ph.D. dissertation, University of Rochester.

Sapp, J, 1994, *Evolution By Association, A History Of Symbiosis by Jan*, Oxford University Press, 1994, New York.

Shackleton, MA, Shipman R. and Ebner, M, 2000, “An investigation of redundant genotype-phenotype mappings and their role in evolutionary search”, in *Proceedings of the international Congress on Evolutionary Computation, CEC2000,* San Diego, July 2000.

Shipman, R, Shackleton, M, Ebner, M, & Watson, RA, 2000, “Neutral Search Spaces for Artificial Evolution: A Lesson from Life”, in *Proceedings of Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*, Bedau, MA, Rasmussen, S, McCaskill, JS, & Packard, NH, (eds.), pp. 162-169, MIT Press, 2000.

Sigmund, K, 1998, “The Population Dynamics of Conflict and Cooperation”, Proceedings of the International Congress of Mathematicans 1998, *Documenta Mathematica* **1** (1998), 487-506.

Simon, HA, 1969, *The Sciences of the Artificial*, Cambridge, MA. MIT Press. (1996 edition)

Sims, K, 1994, “Evolving 3d morphology and behaviour by competition”, in *Artificial Life IV*, Brooks & Maes, (eds.), pp28-39, MIT press.

Slanina, F, and Kotrola M, 2000, “Random Networks Created by Biological Evolution”, Phys. Rev. E, 62(5):6170-6177.

Slatkin, M, 1981, “Fixation Probabilities and Fixation Times in a Subdivided Population”, *Evolution* 35(3) pp 477-488.

Smith, MW, Feng, DF, & Doolittle, RF, 1992, “Evolution by Acquisition: the Case for Horizontal Gene Transfers”, *Trends in Biochemical Sciences* 17(12):489-493.

Smith, RE, Forrest S, & Perelson, AS, 1993, “Searching for diverse, co-operative populations with genetic algorithms”, *Evolutionary Computation*, 1(2):127-149.

Smuts, JC, 1926, *Holism and Evolution*, The Gestalt Journal Press.

Spears, WM, 1992, “Crossover or Mutation?”, in *Foundations of Genetic Algorithms-2*, Whitley, D, (ed.). 221-237.

Spears, WM, De Jong, KA, 1991, “On the Virtues of Parameterized Uniform Crossover”, in *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 230-236.

Starkweather, T, Whitley, D, & Mathias, K, 1991, “Optimization Using Distributed Genetic Algorithms”, in *Parallel Problem Solving from Nature*, Schwefel H. and Maenner R. eds. Springer Verlag. 176-186.

Syswerda, G, 1989, “Uniform Crossover in Genetic Algorithms”, in *Proc. Third International Conference on Genetic Algorithms*, Schaffer, J. (ed.), Morgan Kaufmann Publishers, Los Altos, CA, pp. 2-9, 1989.

Syswerda, G, 1990, “A study of reproduction in generational and steady-state genetic algorithms”, in *Foundations of Genetic Algorithms*, Rawlins, G, (ed.), Morgan Kaufmann, 1991.

Syvanen, M, Kado, CI, 1998, *Horizontal Gene Transfer*, Kluwer Academic Publishers, Boston.

Tomlinson, A, & Bull, L, 2001, “Symbiogenesis in Learning Classifier Systems”, *Artificial Life* 7(1):33-62.

Thompson A, 1998, *Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution*, Springer-Verlag.

Toulouse, M, Thulasiraman, K, & Glover, F, 1999, “Multi-Level Cooperative Search: A New Paradigm for Combinatorial Optimization and an Application to Graph Partitioning”, *Lecture Notes in Computer Science*, #1685, Amestoy, P. et al. (eds.), pp. 533-542.

Tsujimura, Y, Mafune Y, & Gen M, 2001, “Effects of Symbiotic Evolution in Genetic Algorithms for Job-Shop Scheduling”, *34th Annual Hawaii International Conference on System Sciences* (HICSS-34), January 3-6, 2001, Maui, Hawaii - Track 3. IEEE Computer Society, 2001.

Valenzuela, CL, & Jones, AJ, 1994, “Evolutionary divide and conquer (I): A novel genetic approach to the TSP”, *Evolutionary Computation*, 1(4):313-334.

Van Howeghen, C, & Naudts, B, 2001, *H-iff versus Ising: Analysis of a recombinative algorithm* (Tech. Report) University of Antwerp.

Van Hoyweghen, C, Goldberg DE, & Naudts, B, 2001a, “Building Block Superiority, Multimodality and Synchronization Problems”, in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO-2001, Spector, LE, Goodman, Wu, A, Langdon, WB, Voigt, HM, Gen, M, Sen, S, Dorigo, M, Pezeshk, S, Garzon, M & Burke, E, eds., Morgan Kaufmann. pp. 694-701; (also IlliGAL report 2001020).

Van Hoyweghen, C, Goldberg DE, & Naudts, B, 2001b, "From TwoMax to the Ising Model: Easy and hard symmetrical problems", IlliGAL report 2001030.

Van Veldhuizen, DA, & Lamont, GB, 2000, "Multiobjective Optimization with Messy Genetic Algorithms", in *Proceedings of the 2000 Symposium on Applied Computing*, pp. 470-476. ACM.

Vekaria, K, & Clack, C, 1998, "Selective Crossover in Genetic Algorithms", Procs. of PPSN V, Eiben, Back, Schoenauer, Schweffel (eds.), Springer. pp. 438-447.

Vose, MD, *The Simple Genetic Algorithm: Foundations and Theory*, 1999, Bradford Books.

Waddington, CH, 1942, "Canalization of development and the inheritance of acquired characters", *Nature* **150**: 563-565.

Wagner, GP, 1996, "Homologues, Natural Kinds, and the Evolution of Modularity", *American Zoologist* 36, pp. 36-43.

Wagner, GP, & Altenberg, L, 1996, "Complex adaptations and the evolution of evolvability", *Evolution*, Vol. 50, No. 3, pp. 967-976.

Wagner, GP, 1995, "Adaptation and the modular design of organisms", in *Advances in Artificial Life*, Morán, F, Morán, A, Merelo, JJ, & Chacón P, (eds.), Springer Verlag, Berlin, pp 317-328.

Wakeley, J, 2000, "The effects of subdivision on the genetic divergence of populations and species", *Evolution* 54:1092-1101.

Wallin, IE, 1927, *Symbionticism and the origin of species*, Baltimore, Williams & Wilkins company.

Watson, RA, 1996, *A study of the Effect of Group Formation on Evolutionary Search*, MSc dissertation, Sussex University, July 1996.

Watson, RA, 2001, "Analysis of Recombinative Algorithms on a Non-Separable Building block Problem", *Foundations of Genetic Algorithms VI*, (2000), (eds.), Martin WN and Spears WM, Morgan Kaufmann, pp. 69-89.

Watson, RA, Ficici, SG, & Pollack, JB, 2000a, "Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots", *Journal of Robotics and Autonomous Systems*. to appear (also Technical Report CS-00-208).

Watson, RA, Hornby, GS, & Pollack, JB, 1998, "Modeling Building block Interdependency", *Procs. of Parallel Problem Solving from Nature V*, Eiben, Back, Schoenauer, Schweffel (eds.), Springer. pp. 97-106.

Watson, RA, & Pollack, JB, 1999a, "Hierarchically Consistent Test Problems for Genetic Algorithms", in *Procs. of 1999 Congress on Evolutionary Computation, CEC-99*, Angeline, Michalewicz, Schoenauer, Yao & Zalzala, (eds.), IEEE Press, pp. 1406-1413.

Watson, RA, & Pollack JB, 1999b, "Hierarchically-Consistent GA Test Problems: Summary and Additional Results", *GECCO-99 Late Breaking Papers*, pp. 292-297.

Watson, RA, & Pollack, JB, 1999c, "Incremental Commitment in Genetic Algorithms," in *Proceedings of 1999 Genetic and Evolutionary Computation Conference (GECCO 99),* Banzhaf, W, Daida, J, Eiben, AE, Garzon, MH, Honavar, V, Jakiela, M, & Smith, RE (eds.) San Francisco, CA: Morgan Kaufmann. pp.710-717.

Watson, RA, & Pollack, JB, 1999d, "How Symbiosis Can Guide Evolution", *Procs. of European Conf. on Artificial Life (ECAL) V*, Floreano, D, Nicoud, J-D, Mondada, F, eds. Springer. pp. 29-38.

Watson, RA, & Pollack, JB, 2000a, "Symbiotic Combination as an Alternative to Sexual Recombination in Genetic Algorithms", *Procs. of Parallel Problem Solving from Nature (PPSN) VI*, Schoenauer et al. (eds.) Springer pp.425-434.

Watson, RA, & Pollack JB, 2000b, "Recombination Without Respect: Schema Combination and Disruption in Genetic Algorithm Crossover", *Proceedings of the 2000 Genetic and Evolutionary Computation Conference,* Whitley, D., et al (eds.), Morgan Kaufmann. pp. 112-119.

Watson, RA, & Pollack, JB, 2001a, “Symbiotic Composition and Evolvability”, *Procs. of , European Conf. on Artificial Life (ECAL) VI*, Kelemen,, J, & Sosík, P, (eds.), Springer (LNAI 2159), pp. 480 -490.

Watson, RA, & Pollack, JB, 2001b, “Coevolutionary Dynamics in a Minimal Substrate”, *Procs. of Genetic and Evolutionary Computation Conference (GECCO) 2001*, eds. Spector, L., et al., Morgan Kaufmann. pp. 702-709.

Watson, RA, & Pollack, JB, 2002, “A Computational Model of Symbiotic Composition in Evolutionary Transitions”, *BioSystems*, to appear.

Watson, RA, Reil, T, & Pollack JB 2000b, “Mutualism, Parasitism, and Evolutionary Adaptation”, *Procs. of Artificial Life VII*, Bedau, M, McCaskill, J, Packard, N, Rasmussen, S (eds.), 2000. MIT Press, pp. 170-178.

Watts, DJ, 1999, *Small Worlds: The Dynamics of Networks between Order and Randomness,* Princeton University Press, Princeton, New Jersey.

Wiegand, RP, Liles, WC, & de Jong, KA, 2001, “An Empirical Analysis of Collaboration Methods in Cooperative Coevolutionary Algorithms”, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, p. 1235-1242.

Werth, CR, Guttman, SI, & Eshbaugh, WH, 1985, “Recurring origins of allopolyploid species in Asplenium”, *Science* 228, pp. 731-733.

Whitley, D, Beveridge, R, Graves, C, & Mathias, K, 1995a, “Test Driving Three 1995 Genetic Algorithms: New Test Functions and Geometric Matching”, *Heuristics*, 1:77-104.

Whitley, D, Mathias, K, Rana, S & Dzubera, J, 1995b, “Building Better Test Functions”, ICGA-6, Eshelman, (ed.), Morgan Kauffmann, San Francisco, pp. 239-246.

Wiles, J, Tonkes, B. & Watson, JR, 2001, “How learning can guide evolution in hierarchical modular tasks”, in *Proceedings of the 23rd Annual Conference of the Cognitive Science Society (CogSci 2001),* Moore, JD, & Stenning, K, (eds), Lawrence Erlbaum Associates, pp. 1130-1135.

Wilson, KG, 1979, “Problems in Physics with Many Scales of Length”, *Scientific American* 241 (2):158-179.

Wilson, SW, & Goldberg, DE, 1989, “A critical review of classifier systems”, In *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, Schaffer, JD, (ed.) San Mateo, California. Morgan Kaufmann Publishers. pp. 244-255.

Wolpert, D, & Macready, W, 1997, “No Free Lunch Theorems for Optimization”, *IEEE Transactions on Evolutionary Computation*, 1(1):67-82, 1997.

Wright, AH, & Rowe, JE, 2001, “Continuous Dynamical System Models of Steady-State Genetic Algorithms”, Foundations of Genetic Algorithms. Vol. 6. Spears, W, Fogarty, T, & Martin, W, (eds.), Morgan Kaufmann Publishers.

Wright, AH, & Zhao, Y, 1999, “Markov Chain Models of Genetic Algorithms”, *Procs. of GECCO’99*, Banzhaf, et al. eds., Morgan Kauffman, pp. 734-741.

Wright, S, 1931, “Evolution in Mendelian populations”, *Genetics* **16**: pp. 97-159.

Wright, S, 1935, “Evolution in populations in approximate equilibrium”, *Journal of Genetics* 30, 257-66.

Wright, S, 1967, “Surfaces of selective value”, *Proc. Nat. Acad. Sci*. **58**: pp. 165-179.

Wright, S, 1977, *Evolution and the Genetics of Populations*, Volume 3, U. of Chicago Press.