

**The evolutionary engine and the mind machine:
A design-based study of adaptive change**

by

Christopher Complin

A thesis submitted to the Faculty of Science
of The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
Faculty of Science
The University of Birmingham
September 1997

Abstract

The objectives of this thesis are to elucidate adaptive change from a design-stance, provide a detailed examination of the concept of evolvability and computationally model agents which undergo both genetic and cultural evolution. Using Sloman's (1994) design-based methodology, Darwinian evolution by natural selection is taken as a starting point. The concept of adaptive change is analysed and the situations where it is necessary for survival are described. A wide array of literature from biology and evolutionary computation is used to support the thesis that Darwinian evolution by natural selection is not a completely *random* process of trial and error, but has mechanisms which produce trial-selectivity. A number of means of creating trial-selectivity are presented, including reproductive, developmental, psychological and sociocultural mechanisms. From this discussion, a richer concept of evolvability than that originally postulated by Dawkins (1989) is expounded. Computational experiments are used to show that the evolvability producing mechanisms can be selected as they yield, on average, 'fitter' members in the next generation that inherit those same mechanisms. Thus Darwinian evolution by natural selection is shown to be an inherently adaptive algorithm that can tailor itself to searching in different areas of design space. A second set of computational experiments are used to explore a trajectory in design space made up of agents with genetic mechanisms, agents with learning mechanisms and agents with social mechanisms. On the basis of design work the consequences of combining genetic and cultural evolutionary systems were examined; the implementation work demonstrated that agents with both systems could adapt at a faster rate. The work in this thesis supports the conjecture that evolution involves a change in replicator frequency (genetic or memetic) through the process of selective-trial and error-elimination.

'I believe it was Plato who said that good judgement consists equally in seeing the differences between things that are similar and the similarities between things which are different.' (Brian Magee, *Confessions of a Philosopher*, p. 138)

Acknowledgements

I am very grateful to Aaron Sloman for giving me the opportunity to do this research and for his continual support and guidance throughout. This was no mean task as I came to the department with no knowledge of computers or computer science.

I would like to thank my Thesis Group: Glyn Humphreys and especially Peter Hancox who has provided much useful advice and encouragement.

I would also like to thank my fellow research students, Symon Cotton for his advice and assistance on a number of computational issues and Dorota Iskra for proof reading my thesis.

This research has been funded by EPSRC, studentship number: 94701624.

Table of Contents

1. Introduction	1
1.1 The objectives	1
1.2 The intended audience.....	3
1.3 An overview of this thesis.....	4
2. A design-based study of adaptive change	7
2.1 Introduction	7
2.2 Different answers: types of explanations	7
2.3 The methodology: the design-based approach	12
2.4 The ontology: designs and niches	14
2.4.1 Designs.....	14
2.4.2 Niches.....	18
2.5 What is adaptation: conceptual analysis.....	20
2.6 Why adapt: requirements analysis.....	26
2.7 Summary	29
3. Abstract evolution	31
3.1 Introduction	31
3.2 The Darwinian process: Darwin's dangerous idea.....	32
3.2.1 The algorithmic nature of the Darwinian process	33
3.2.2 Abstract formulations of the Darwinian process.....	35
3.3 The heredity mechanisms: the neo-Darwinian formulation	38
3.3.1 The genetic code.....	39
3.3.2 Genetic algorithms and the Schema Theorem.....	41
3.3.3 Abstract formulation of the mechanisms	46
3.4 The myopic nature of the Darwinian process.....	53
3.5 Chapter Summary.....	56
4. The evolution of evolvability	60
4.1 Introduction	60
4.2 How developmental programs can contribute to evolvability.....	61
4.2.1 Origins of order	68

4.2.2 Kaleidoscopic embryologies	71
4.2.3 The architecture of complexity.....	74
4.2.4 Genetic programming.....	76
4.2.5 Summary on development.....	79
4.3 Reproductive mechanisms and the selection of evolvability	81
4.3.1 Introduction	82
4.3.2 Method	84
4.3.3 Results	88
4.3.4 Discussion	90
4.3.5 Conclusions and further work	95
4.4 Chapter Summary.....	96
5. Trajectories in design space	98
5.1 Introduction	98
5.1.1 Evolutionary epistemology.....	100
5.1.2 Adaptive agents	103
5.2 Method	107
5.2.1 Darwinian creatures.....	107
5.2.2 Skinnerian creatures	108
5.2.3 Dawkinsian creatures	109
5.2.4 Vygotskian creature.....	109
5.3 Results	110
5.4 Discussion	110
5.4.1 Darwinian creatures.....	111
5.4.2 Skinnerian creatures	115
5.4.3 Dawkinsian and Vygotskian creatures	121
5.5 Summary and further work.....	126
5.5.1 Further work: Popperian and Gregorian creatures	127
5.5.2 Darwin machines.....	130
6. Summary	132
6.1 Overview	132
6.2 The main points of this thesis.....	132

6.3 Further work.....	135
6.4 Contributions.....	137
7. Appendix 1.....	139
7.1 NK Landscape program.....	139
7.1.1 NK landscape generation procedures	139
7.1.2 Fitness evaluation procedures	140
7.2 Programs used to run experiments on the selection of evolvability.....	140
7.2.1 Tournament selection	140
7.2.2 Generators of variation.....	141
7.2.3 Miscellaneous procedures	141
7.3 How to run the program	142
8. Appendix 2.....	143
8.1 Main	143
8.2 Bitprocs	144
8.3 Classes.....	145
8.4 Vacfs.....	145
8.5 Spp_ga.....	146
8.6 Remove	146
8.7 Social.....	146
8.8 How to run the program	147
9. Glossary.....	149
10. List of references	150

Table of Illustrations

Figure 4-1: Weismann's 'fundamental principle of embryology'	60
Figure 4-2: Results of Koz'a (1995) genetic programming experiments with and without automatically defined functions.	79
Figure 4-3: Fitness time graph using first adaptive change measure.	86
Figure 4-4: Fitness time graph using second adaptive change measure.....	87
Figure 4-5: Graph showing the rate of change in mean fitness v. population size.....	88
Figure 4-6: Map showing predicted generator and selected generator using 1 st measure.	89
Figure 4-7: Map showing predicted generator and selected generator using 2 nd measure.	89
Figure 4-8: Fitness-time showing the action of selection.....	91
Figure 4-9: Fitness-time graph of mutation v. mutation & recombination, $K = 3$	92
Figure 4-10: Fitness-time graph of mutation v. mutation & recombination, $K = 0$	93
Figure 4-11: Graph showing the selection of mutation rate.	94
Figure 5-1: Performance System	104
Figure 5-2: Graph showing the rate of adaptive change of 5 different kinds of agents.	110

1. Introduction

Consider this brief and very simplified description of the history of the universe. After the Big Bang the matter and energy which had been produced began to interact and become organised into different structures. Elementary particles were created and different combinations formed further particles. Atoms combined into different elements and compounds. Chemical ‘evolution’ led to the creation of different molecules. Eventually, these molecules were combined into the first life forms. These organisms changed through time by a process of biological evolution. Some life forms changed so that they developed from single cells into multi-cellular structures. Organisms evolved the ability to learn which made it possible for further changes to occur within their lifetime. Some multi-cellular organisms developed a specialisation which amplified this potential for behavioural change: a nervous system. Later an area of the nervous system became specialised - the brain - allowing an even greater degree of behavioural plasticity. Further organisms evolved the capability of observing other organisms and modifying their behaviour accordingly, and some organisms were produced that were able to instruct one another as to how to behave and how not to behave. In this description there are many references to change, to structures being modified through time. Some of these changes are called adaptive and it is these forms of change which are the focus of this thesis.

In this thesis the following questions are asked: What is adaptive change? How is it possible for an agent to adapt to its environment? What mechanisms are involved in this process? Are some of these mechanisms common to adaptive changes in different domains? In the rest of the introduction I will state the objectives of this thesis, the intended audience, and give an outline of the structure of this document.

1.1 The objectives

The objective of this thesis is to examine the concept of adaptive change, the mechanisms involved in one instance of adaptive change - biological evolution - and the applicability of these mechanisms to other domains. The idea that biological evolution could form a part of an explanation of other adaptive phenomena is not a new one. While Charles Darwin, T.H. Huxley and Herbert Spencer all alluded to the generality of evolutionary theory, it was really

William James in an essay in 1880 that first thoroughly applied the Darwinian process to explaining what went on in other domains (Plotkin, 1995). Since that point in time there have been many different writers who could be called evolutionary epistemologists, for example, Baldwin, Campbell, Dennett, Piaget, Popper, Lorenz, Skinner, and Simon. The aim in this thesis is to examine through conceptual analysis and design exploration the ideas of the evolutionary epistemologists and produce computational models which can give insight into the generality of the mechanisms involved in adaptation. Note that this research aims to test and refine theories through design work and computational modelling, not develop computer programs of the sort produced by evolutionary computation researchers.

In this research I have:

- provided an analysis of the concept of adaptation and used the concepts of design space and niche space (Sloman, 1994) to provide a domain-independent framework for asking questions about adaptation
- analysed the requirements of different niches and designs which would lead to a need for adaptive mechanisms
- addressed the question of how adaptation is possible and examined in detail one possible answer: the process of biological evolution by natural selection
- examined the mechanisms involved in biological evolution and asked what their essential properties are, independent of the implementation details
- considered which variables influence the rate of adaptation
- examined Darwinian and neo-Darwinian theories from a computational perspective
- considered the role of development and reproduction in the evolutionary process
- used computational experiments to demonstrate that a Darwinian process can select the kind of reproductive mechanism which leads to the fastest rate of adaptation of their products in a given situation
- used design work to outline a trajectory in design space of agents with different adaptive capabilities and examined how this can affect their ability to adapt
- used computational experiments to quantify the relative differences in the rates of adaptation between these agents in the scenario used
- considered the requirements, costs and benefits of different adaptive mechanisms

- speculated what additional capabilities could improve the adaptability of a design

1.2 The intended audience

In writing this thesis I have had to carefully consider the background of the audience who would be reading it. A biological mechanism is taken, described from a computational viewpoint and applied to explaining certain problems in philosophy and psychology. Therefore, the main disciplines this work contributes to are biology, computer science, AI, psychology and philosophy.

Philosophers going back to Plato and beyond have considered how things change and what differences there are between different types of change. Both these issues are central to this thesis with work being done to distinguish the different kinds of change and provide a framework for asking questions about adaptive change. Other philosophers - epistemologists - are interested in what it is we can know and how we can know it. Some of the later work in this thesis examines the views of the evolutionary epistemologists who try to answer questions about epistemology in the light of evolutionary theory. Computational models are used to formalise the evolutionary epistemologists' theories and to allow testing of some of their claims, for example, the claim that the Darwinian process of natural selection is a sufficient mechanism to produce adaptive change within the lifetime of an individual.

In recent years with the advent of the field of evolutionary computation, taking a computational approach to evolutionary theory has become very popular. This type of approach has two main advantages: first, it forces a precise formulation of the mechanisms independent of their biological base; second, it provides a means of testing the properties of these mechanisms, often in ways impossible in the real world. This work produces an algorithmic formulation of evolutionary theory which takes into account the work done by different mechanisms in contributing to the rate of adaptation. Computational experiments are used to test the power of mechanisms which would allow an agent to influence subsequent generations, for example, mechanisms allowing imitation and instruction of other agents.

Psychologists have paid much attention to the phenomena of learning and psychological development. Some researchers have postulated how different forms of adaptation may be linked (Baldwin, 1896) or are of a similar form (Skinner, 1969; Piaget,

1980). This work will be of interest to psychologists who are interested in how evolution can inform psychological systems and how learning can inform or even usurp the place of biological evolution.

Finally, computer scientists and AI researchers have been looking at many of the same questions as the psychologists, but from the information-processing perspective. As I have already mentioned, there has recently been an increasing amount of work on biologically inspired models of adaptation. These researchers will be interested in this work as it shows the contribution of different mechanisms to the adaptive process and warns against ignoring the effect of sociocultural environments on the intelligence of certain kinds of agents.

1.3 An overview of this thesis

This thesis is divided into six chapters (including this one) and two appendices, and is structured as follows:

Chapter 2 concentrates on philosophical matters to do with the study of adaptation. There are two main aims of this section: first, to lay out and justify the methodological approach taken in this thesis - the design-based approach (Sloman, 1994), and, second, to analyse the concept of adaptation. A model based on the concepts of design space and niche space (Sloman, 1994) will be described to facilitate the discussion of adaptive change in different domains and to help in the process of asking meaningful questions about what is driving adaptive change.

Chapter 3 and 4 provide an analysis of the mechanisms of biological evolution. The aim is to provide a precise formulation of an adaptive mechanism, its requirements and properties. Chapter 3 describes Darwin's original theory and the subsequent neo-Darwinian formulation. In these chapters natural selection is shown to be a domain-independent, adaptive algorithm whose properties arise from its logical structure. The Darwinian process will be described as an algorithm, its properties will be given, and the different abstract formulations of evolution will be discussed. Furthermore, to help avoid possible terminological confusions the concepts of heredity and self-replication will be examined.

The second half of chapter 3 deals with the heredity mechanisms otherwise called the genetic mechanisms. Again the focus of this section is to show that it is only the logical form of the mechanisms and not the material out of which they are made that is important. This

discussion will include, for example, an examination of whether the genetic code is an arbitrary code (Hofstadter, 1986) with certain properties, such as digital (Maynard Smith, 1997). The concept of the gene will be analysed as there are different senses in which this word is used (Dawkins, 1989a). Finally, genetic algorithms will be discussed as they have led to the production of the Schema Theorem (Holland, 1994), a mathematical theory which allows insights into the role of different genetic operators in genetic search.

Chapter 4 focuses on the contributions of development and reproduction to the evolutionary process. First, in the section on development the concept of the genotype-phenotype mapping will be considered from an information-processing perspective. Understanding what the genotype denotes and the role of self-organisation in evolution is essential to understanding the constraints acting on adaptation. Dawkins' (1989a, 1996) computational work on embryologies is summarised to illustrate how development can be described as an algorithm and what effect different types of developmental programs have on adaptability. Next Herbert Simon's (1982) work is described on how a hierarchical structure could affect the rate of evolution and have certain implications for the complexity of what could be evolved. The final part of this section is a discussion of recent work in evolutionary programming that has developed a technique called 'automatically defined programming' which provides an implementation of Simon's ideas. The overall aim of this section is to evaluate the importance of development to evolution.

The work described in the second half of chapter 4 focuses on the role of reproduction in the process of evolution. Computational experiments carried out in this research will be discussed which compare the performance of different types of reproductive mechanisms operating in a range of population sizes on a range of different correlated fitness landscapes. The aim of this section is to explore how the evolutionary process can select the reproductive mechanisms which lead to the fastest rate of adaptation in a particular situation. These experiments demonstrate that certain kinds of reproduction give rise to faster adaptation, relative to other reproductive mechanisms, in specific contexts and that the fastest mechanisms can be selected by the evolutionary process.

Chapter 5 describes the design explorations carried out in order to investigate the requirements of agents which have Darwinian processes operating between and within them. The design decisions and implementation details for these agents are discussed along with

the costs and benefits associated with the different mechanisms. In describing the different agents I have borrowed Dennett's (1986, 1995, 1996) naming convention - Darwinian and Skinnerian creatures - while adding the names Dawkinsian and Vygotskian creatures to the scheme.

The first type of agent, the Darwinian creature, is an agent whose behaviour is fixed at the point of creation and then adapted by genetic evolution. The second type of agent, the Skinnerian creature, has mechanisms allowing a certain amount of trial and error learning within the lifetime of the agent. Dawkinsian creatures are capable of imitating other agents and thus have a second heredity system: the memetic system. The final type of agent, the Vygotskian creature, is capable of instructing other agents and thereby biasing what is inherited. The performance of these agents in adapting to an abstract task is quantified and compared. These experiments demonstrate how different kinds of agents - agents with different mechanisms based on the Darwinian process - can adapt at different rates. The implementations elucidate the requirements of these mechanisms and suggest how they can function to increase the rate of adaptation. In the discussion the costs and benefits of the different mechanisms are analysed. At the end of this chapter the implications of adding other mechanisms - for example, an internal model of the world - are considered and a scheme is outlined for formally specifying the different implementations that satisfy the requirements of Darwinian evolution.

Chapter 6 concludes this thesis by summarising its main points, outlining areas for further research, and stating the main contributions of this work.

The appendices include a description of the computer programs written in the course of this work. Too large an amount of code has been developed to be included here in its original form. Therefore, it seemed more practical and appropriate to give just a functional overview of some of the main procedures. The location of the code on the University of Birmingham, School of Computer Science computer network is given to allow further inspection. Appendix 1 describes the code used in the computational experiments described in chapter 4; appendix 2 describes the code developed for the computational experiments described in chapter 5.

2. A design-based study of adaptive change

2.1 Introduction

‘In philosophy,’ Bertrand Russell once remarked, ‘what is important is not so much the answers that are given, but rather the questions that are asked’ (Russell, 1959, p. 19). The late philosopher Karl Popper suggested that when reading a piece of work the first question you should ask is ‘What problem is he trying to solve,’ and not ‘What is he trying to say’ (Magee, 1990, p. 67). These two quotes identify the aim of this chapter: to clarify the questions being asked. Thus the first objective is to examine and precisely formulate what is meant by the questions: What is adaptation? and How is adaptation possible? The second objective is to identify what would be an acceptable answer to these questions and what methodological approach will give such answers. In fact these objectives are addressed in reverse order, starting with identifying the type of explanation being sought. The novel contribution to research in this chapter is to provide a detailed examination of the concept of adaptive change and an outline of a methodology and ontology to help in studying it.

2.2 Different answers: types of explanations

Many philosophers have wrestled with the problems of what an explanation is and what types of explanation are possible. It is beyond the scope of this thesis to consider these questions here. However, a selective account of this topic is given to emphasise that different disciplines seek different kinds of explanation and to specify the type of answers sought in this thesis.

A good place to start is with Aristotle. He identified four questions he wanted answered about anything - the four causes. These causes are really four different types of explanation (Nussbaum, 1988). Consider, for example, the question: Why is a flower as it is? One possible explanation would describe the materials that make up the flower. This is an example of a material explanation - Aristotle’s material cause. Another answer would be that the flower is as it is because it has that structure, that particular form. This Aristotle called the formal cause - a formal explanation. There is a third type of explanation called the efficient cause - the efficient explanation. This answers the question of why a flower is as it is by referring to the flower’s genesis, where it came from originally. Aristotle’s last cause is

called the final cause - the final explanation. The final cause addresses the purpose or goal of the flower.

Since Aristotle philosophers have argued over and added categories to the different types of explanation. The standard classification of explanations given in text books of philosophy is (Sloman, 1996b):

- causal (both deterministic and probabilistic)
- functional (both in biology and in connection with artefacts)
- purposive (involving intelligent agents and their intentions)
- genetic (i.e. historic)
- hermeneutic (i.e. interpretative: e.g. explaining the posture of a figure in an allegorical painting)
- logico-mathematical (e.g. answers to questions like ‘Why are there infinitely many prime numbers?’)

However, philosophers do not universally agree on this list of explanations. In fact, it is often not clear which explanations fall into which categories or if certain explanations are possible in certain cases. The central point of this description is that researchers from different disciplines often seek different kinds of answers and unnecessary arguments can arise if this is not realised. Freud, for example, used historical explanations to answer questions about people’s behaviour, whereas modern medicine seeks answers to question about abnormal behaviour in terms of neuroanatomy and neurochemistry - causal (material) explanations.

Let us briefly consider how these different types of explanations have been used in two of the areas of science relevant to this thesis. In ethology, the biological study of behaviour, Tinbergen (Manning and Stamp Dawkins, 1992) identified four questions that should be asked about behaviour. These are the questions of causation, development, evolution and function.

Manning and Stamp Dawkins (1992) describe the escape behaviour in cockroaches to illustrate Tinbergen’s four questions. The question of causation refers to the mechanisms inside the cockroach and the stimuli acting on it from the outside that enable it to detect it is

about to be attacked and allow it to scuttle away successfully. The second question, the question of development, asks whether the escape behaviour is innate, learnt or some combination of the two. Evolutionary and developmental questions are similar as they both refer to the chronological record of the emergence of structure, but evolutionary questions refer to how behaviour emerged over species-time, while developmental questions refer to changes that occur within an individual's lifetime. In the case of the cockroach, for example, what was the behaviour that evolved into the escape behaviour we now see? The final question, the question of function, asks the ultimate function of the behaviour: that is, what would happen if that capability were not present? In ethology, for example, final questions refer to how a behaviour contributes to an animal's survival and reproductive success.

The types of answers ethologists seek are, according to the philosophers, causal, historical (Tinbergen's developmental and evolutionary questions) and functional explanations. In cognitive science and AI, however, the focus of explanation has been slightly different. Daniel Dennett (1986), a philosopher and cognitive scientist, proposed three stances: the intentional, the physical and the design stance. To illustrate the design stance Dennett (1986, p. 4) gives the following example, 'If one knows exactly how the computer is designed (including the impermanent parts of the design: the program), one can predict its designed response to any move by following the computational instructions of the program'. The essential feature of the design stance is that we can make predictions solely from knowledge or assumptions about the system's functional design, irrespective of the physical constitution or condition of the innards of the particular object. Thus, the different design stance predictions rely on the notion of function.

In contrast to the design stance is the physical stance where our predictions are based on the actual physical state of the particular object and are worked out by applying whatever knowledge we have of the laws of nature. Only using the physical stance, according to Dennett (1986), can we predict the malfunction of systems - unless a system is designed to malfunction after a certain time, in which case malfunctioning in one sense becomes part of the proper functioning. However, if you understand the design and the design has bugs, then you can use that knowledge to predict some malfunctions.

According to Dennett, when interacting with certain systems it is best to adopt the intentional stance, for instance, when playing chess against a computer. Adoption of the

intentional stance depends on the situation, for example, for the purpose of a designer trying to improve a chess computer, the design stance is more appropriate. Dennett argues that certain systems are too complex to be approached from either the physical or the design stances. By making two assumptions - 1) that the machine will function as designed, 2) that the design is optimal, for example, a chess computer will choose the most rational move - one can try to predict the behaviour of a machine by treating it as an intentional system. However, this assumes that we can work out what the rational move is going to be and in general this is not feasible: sometimes the rational move is not even well defined, like when goals are in conflict.

The intentional stance presupposes rationality. Dennett is not the only researcher who felt the need to introduce the concept of rationality into explaining certain systems. The computer scientist and artificial intelligence researcher Alan Newell (1982, 1992, 1994), also described three levels: the knowledge level, the symbol level and the architectural (physical) level. The knowledge level, like the intentional stance, presupposes rationality.

Dennett's (e.g. 1995, p. 230) later descriptions of the intentional stance stress treating a system *as if* it were a product of a reasoned design development, with a series of choices among alternatives, in which the decisions reached were those deemed best by the designers. By doing this Dennett (1995) has blurred the distinction between the design and intentional stances, favouring the design stance.

To summarise Dennett's position, his stances relate to the philosopher's different types of explanations as follows: the design stance relates to functional explanations, the physical stance to causal explanations and the intentional stance to purposive and hermeneutic explanations.

The preceding discussion has outlined the different kinds of explanation. It is now time to relate these to the work in the rest of this thesis. The main aim of this research was to investigate a class of adaptive architectures. An architecture is a set of functional mechanisms with causal links between them (Sloman, 1994). Architectural descriptions of a structure are defined in terms of its organisation and functioning: that is, in terms of formal and functional explanations.

An architecture is a description of a machine, the formal equivalent of which is an algorithm (the logic underlying the dynamic of a machine, regardless of the details of its

material construction (Langton, 1989)) or a set of algorithms. There has been a growing interest in algorithmic explanations of things, for example, in the fields of evolutionary computation and artificial life. While computer modelling is now common in the natural sciences, the role of computation as a framework for describing the mechanisms underlying natural systems is on the increase (Belew *et al*, 1996).

One of the obvious advantages of algorithms is that it is possible to run them on a computer. This provides a means of testing the theory embodied in a subset of architectures and exploring their properties. Furthermore, the process of science has often been described as one of building models - simplified and abstracted descriptions - of natural systems (e.g. Belew *et al*, 1996). Models can be defined more precisely as formal expressions of the relationship between defined entities in physical or mathematical terms (Jeffers, 1982). While traditionally models have been mathematical, the development of the mathematics embodied in computer programs - algorithms - has allowed the capturing of complications that are difficult to treat with traditional mathematical models. For instance, mathematical models are typically limited to describing aggregate characteristics of a system rather than individual interactions (Belew *et al*, 1996).

For scientists the essential property of models is that they give rise to predictions. For example, Williams (1996) gives the following account of the discovery of Troy. The archaeologist Heinrich Schliemann proposed a theory of the narrative history and specific geography of the city of Troy on the basis of the Homeric epics and classical scholarship. In the 1870's Schliemann carried out archaeological investigations at the western end of the Dardanelles and verified his prediction of the location of Troy.

Many philosophers of science (cf. Losse, 1993) have emphasised how models give rise to predictions and that these predictions can be compared to empirical phenomena. This provides a basis of discriminating between models: one model can be regarded as better than another if it more accurately predicts the phenomena that it models. Unfortunately, discriminating between models is not that simple, for example, problems arise when models predict different overlapping phenomena. Thus, computational models not only provide a way of formalising models, but also of examining their predictions.

An important corollary of using algorithmic descriptions of architectures is that in many cases they provide a domain-independent description of the mechanisms involved. Just

as mathematical models can be applied to a wide range of different phenomena, so can algorithmic models. If certain conditions of these simplified and abstract models are met, then the model can be used to explain different types of phenomena.

To summarise, the explanations given in this thesis are of terms of function - functional explanations - and form - formal explanations. Furthermore, algorithmic descriptions are used because they provide a domain-independent formalisation of the theories examined and a means of testing the model's behaviour.

2.3 The methodology: the design-based approach

The aim in this section is to describe the methodological approach taken in this thesis and not to give an exhaustive justification of it. The design-based approach (Sloman, 1994) combines requirements analysis from software engineering, conceptual analysis from philosophy, and introduces certain concepts - for example, designs and niches - to give a comprehensive methodology for studying actual or possible systems.

The design-based approach has five main steps according to Beaudoin (1995). These steps can be executed iteratively or in parallel. The first step involves specifying the requirements of the system in question in any order. These requirements may be explicit in an engineering project or implicit in biological evolution (Sloman, 1994).

The second step involves proposing designs which satisfy the requirements given in the first stage. A design is a description of an architecture (see page 10) - that is, components with causal links between them - and its mechanisms. Step three involves implementing these designs. Building computer simulations or hardwired machines can uncover inadequacies in the original design and the realisation of additional requirements and constraints. Thus the design-based methodology does not have to be followed in a linear manner, jumping between stages is allowed.

Step four involves an analysis of both how well the design meets the requirements and how well the implementation embodies the design. This analysis can be done both mathematically or by experimental testing of the implementation.

The final stage is an investigation of similar designs to the system in question. To fully understand a design requires an understanding of the implications of changes to that design, for example: What changes would have no effect on the design's functionality? What impact

would changes in requirements have on a design? What are the constraints and trade-offs in the design? What further capabilities could the system have if the design were slightly different? Thus, as Sloman (1995) wrote, ‘Full understanding [of a system], therefore, requires an analysis of the similarities and differences between actual and possible systems’. This process is referred to as exploring a neighbourhood in design space.

However, implementing complex systems is often a daunting task. Sloman (1994) suggests a number of useful simplifications which can assist in specifying requirements and exploring designs. For example, these three simplifications were used in this thesis:

1. simplifying the environment and the agents’ interface with it
2. focusing on complete, but very simple agents with relatively few capabilities
3. exploring designs with a wide variety of functions, but with shallow implementations of these functions

The third simplification is often referred to as ‘broad but shallow’ implementations (Bates *et al*, 1991). The idea is very simple: if progress in understanding is to be made, then, initially, the complexity of most problems requires certain simplifications. The standard problem solving approach is called ‘divide and conquer’, that is, breaking down a problem into sub-problems and solving those problems individually. However, an equally important task is to understand how components interrelate. For this purpose Bates and his colleagues suggested a broad but shallow approach. If an agent with a range of capabilities is studied, in order to make it possible to implement all the capabilities, the different mechanisms are simplified. Broad but shallow agents are obviously not the ideal, but merely a starting place with the intention of deepening the implementation of subsequent models. However, a consequence of the broad but shallow approach should be noted: that is, it necessarily leads to an eclectic literature survey where the breadth of topics becomes more important than the depth in a single topic.

To summarise, the design-based approach is an engineering methodology which uses techniques developed in computer science and philosophy. The approach aims to: 1) specify concepts in relation to designs - producing theory-based concepts; 2) explore the problem -

specifying the niche; 3) explore the possible solutions - exploring design space. The concepts of designs and niches are central to this thesis and are analysed in the next section.

2.4 The ontology: designs and niches

The term ‘ontology’ refers to the things present in a given domain, ‘the catalogue of what exists’ (Dennett, 1995). The concepts denoted by the words ‘designs’ and ‘niches’ are fundamental to the subsequent discussions in this thesis. The aim of this section is to clarify what is meant by these two interrelated concepts. However, the analysis in this section attempts to provide useful terms, it is beyond the scope of this thesis to produce an exhaustive, philosophically justifiable ontology.

2.4.1 Designs

In this thesis the term ‘design’ is used to refer to the abstract description of the functional architecture of both real or possible systems. The notion of a design is not restricted to human artefacts - for instance, evolution can be thought of as a designer. Sloman (1995) gave a general definition of ‘design’ as a specification of a working system. Designs, however, are closely linked to certain environments. An implementation of a biological design - for example, a fish - could not be described as a working system if it was found on the moon. Thus, the implementation of the fish design only works in environments with certain properties.

A better description of a design is as a possibility transducer (Sloman, 1996a). A design, in this sense, is a structure which on receiving a set of inputs gives a set of outputs, with the structure of design acting to constrain the relationship between the inputs and the outputs. Thus a design links two sets of possibilities such as, properties of objects, events or processes. Sloman (1996a) gives the following example of a possibility transducer: ‘[a] piece of wire can be seen as a transducer from one range of possibilities (possible voltages) to another range of possibilities (possible currents) with the ability to constrain the relationship between inputs (voltages) and outputs (currents). That ability is its electrical resistance.’

The relationship between resistance and sets of possibilities is just one of many examples of properties linking sets of possibilities (Sloman, 1996a). A wire will have other

properties which act as possibility transducers - a modulus of elasticity, for instance. The extent to which different temperatures change the elasticity is an example of a second-order possibility transducer. Thus a design can act as more than one possibility transducers and consequently fit a range of niches. The set of niches a design fits can be called its 'niche span'.

To summarise thus far, a design is a specification of which an input-output device linking various sets of possibilities is an instance. The rest of this section will address the following questions: How are designs implemented in the physical world? Are all design features equally important to a design's functionality? Why do designs in the natural world often appear to be hierarchically organised? In what sense can some designs be non-physical? Each of these points will be considered in turn.

Aristotle argued that the concept of form - what something is - is related to its order or structure, not its material constitution. He believed this for three reasons (Nussbaum, 1988). First, in the case of a living system, matter is always going in and out, it is always changing. Second, so long as the functional structure of something remains the same, replacing a piece of matter would not change one thing into something else. Finally, '[Aristotle] argued that matter is just not definite enough to be what a thing really is. Matter is just a lump or heap of stuff, so we could not say you are some stuff or other: it's only when we have identified the structure the stuff constitutes that we can even go on to say something intelligent about the stuff itself' (Nussbaum, 1988, p. 44). Aristotle was arguing that it is the form, the organised set of functional capabilities, which defines a thing. In describing a design it is *what* the thing does - its functional architecture - not *how* it does it - the implementation detail - that is important. More recently Chris Langton (1989) has proposed that the ability to separate the 'logical form' of an organism from its material basis of construction is the principal assumption of artificial life.

Aristotle also observed that certain changes can be made to a thing which do not change it into something else. Replacing the human heart, for example, with a mechanical heart will have a negligible effect on the functioning of the human body. To explain this Aristotle postulated two types of properties (Dennett, 1995): essential properties - the goldness of gold; and accidental properties - the weight, size, or shape of a particular nugget. Consider, for example, a ruler. The essential property of a ruler - that of affording measuring

- is the necessary property of all rulers. However, a wooden ruler, because it is made of wood, has the accidental property of being an insulator of electricity; a metal ruler has the accidental property due to its material constitution of being able to conduct electricity. Thus real world instantiations of designs have a whole set of properties, with a multitude of accidental properties arising naturally out of that particular implementation.

So how are designs implemented? In the real world designs are made up of other designs and so on and so on. Herbert Simon (1982, p. 198) illustrates this point in the following passage: 'Taking the cell as the building block, we find cells organised into tissues, tissues organised into organs, organs into systems. Within the cell are well defined subsystems - for example, nucleus, cell membrane, microsomes, and mitochondria.' Simon calls such structures 'hierarchic systems' - systems composed of interrelated subsystems. The same idea is referred to by Plotkin (1995) as a structural hierarchy.

As described earlier, each design has an assortment of properties linking sets of possibilities. But as Sloman (1996a) writes, 'they are separate sets [of possibilities]: possible currents in one piece of wire (W1) are different things from possible currents in another piece of wire (W2). Normally, the properties of W1 do not constrain the possibilities associated with W2, only the possibilities associated with W1.' Simon's example of the hierarchical arrangement of the human body illustrates how new links between sets of possibilities can be created by combining physical objects into larger structures, in this case cells into tissues, tissues into organs, etc. Thus as Sloman (1996a) writes, 'a complex machine is created from simple components by linking them in such a way that the associated possibilities are also linked, and thus we get large scale constrained behaviour that is useful to us, and emergent capabilities such as telling the time and providing shelter'.

The examples of implementations of designs given so far have involved physical properties. However, designs can also be non-physical (Sloman, 1994). With the advent of computers we have begun to explore how some physical machines can implement virtual machines - like programming languages, for instance. At the level of bit manipulation a computer can perform three actions: set a bit, zero a bit, or flip a bit (Harel, 1987). These operations are combined to form different designs in a hierarchical structure which produce the machines that we are familiar with, such as, word processors and video games.

Furthermore, virtual machines facilitate certain types of self-modifiability - for example, the rapid structural reorganisation observed in learning.

A recurring feature in this discussion is the modular property of designs. The partitioning found in hierarchical systems is explained by Simon (1982) as due to intracomponent linkages being generally stronger than intercomponent linkages. Simon (1982, p. 217) writes, 'This fact has the effect of separating the high-frequency dynamics of a hierarchy - involving the internal structure of the components - from the low-frequency dynamics of a hierarchy - involving the interaction among the components.' Simon calls such systems nearly decomposable systems. The modular thesis is developed in chapters 3 & 4 (see sections 3.3.3, 4.2.1, & 4.2.4) where further reasons for expecting modularity are examined.

At the start of this section a design was defined as a *specification* of a working system - that is, a description of a thing's functional architecture. If a design has no redundancy - if no aspect of its structure can be inferred from any other - then it is its own simplest description. Hierarchical systems, however, can often be described in more economical terms. Simon (1982) identifies three forms of redundancy: 1) hierarchical systems are usually composed of only a few different kinds of subsystems in various combinations and arrangements; 2) hierarchical systems are often nearly decomposable - only aggregate or average properties of the parts enter into the description of the interactions of those parts; 3) by appropriate 'recoding,' the redundancy which is present but not obvious in a structure can be made explicit, for example, moving from a state description - a blueprint - to a process description - a recipe or algorithm.

To summarise, the concept of a design refers to a description of the functional structure of a machine which is responsible for linking sets of possibilities. The form of a machine can be abstracted from its matter and the logic underlying its dynamics described as an algorithm. I have described how new designs can be made out of other designs by linking sets of possibilities constrained by properties of the components of the design. Finally, the description of designs can often be simplified by taking advantage of redundancies implicit in the system.

2.4.2 Niches

A natural environment has a number of dimensions, including such physical and chemical factors as temperature, humidity, salinity, and oxygen concentration, and such biological factors as prey species and resting backgrounds in which an individual may escape detection by predators. It was the ecologist G. Evelyn Hutchinson in 1957 (Ricklefs, 1990) who suggested that ideally each of these dimensions could be thought of as a dimension in space. The niche of each species occupies a part of this n -dimensional volume that represents the total resource space available to the community. Sloman (1994) generalised the biological concept of a niche to be a set of requirements.

There are two types of requirements: functional and non-functional. Flying is a functional requirement, whereas gravity is a non-functional requirement - a constraint. Non-functional requirements limit what designs are possible. Some constraints - for example, gravity - are implicit within the environment; other constraints arise out of earlier design decisions - for instance, size and shape affects what can and cannot fly.

Some requirements are necessary, such as a biological organism's need to transform energy to keep it out of equilibrium with the physical forces of gravity, heat flow, diffusion and chemical reaction. There are, however, a number of sufficient means of acquiring energy. For example, organisms can use photosynthesis or eat other organisms. Requirement satisfaction is not a simple matter of finding a single design, but a matter of finding one of a number of functionally equivalent designs which, while being equivalent in some respects, may differ in others.

To fully understand a design requires an understanding of the design's niche. As previously mentioned, the physical environment alone does not determine an ecological niche. Animals can live side by side in the same environment and have completely different sets of requirements. For example, a bee and a flower may exist in the same setting, but different requirements are relevant to their designs.

A complicating factor in specifying sets of requirements is that niches change. As an ecological niche is partly defined by the designs whose instances exist there - bees partly define a flower's niche and flowers partly define a bee's niche - if those designs change, then the requirements acting on other designs can change. So if a prey species develops better escape strategies, for example, this changes the requirements of predator's capture strategies.

As a niche is made up of a set of requirements, there are often cases of conflict. The efficiency of a design can be calculated by certain benefits being divided by certain costs (Dawkins, 1996). A design may be able to go faster - the concord, for example - but the cost of this additional speed may outweigh the benefit. Another illustration of this would be when designing a house which has to be both safe and affordable. These requirements will lead to certain compromises if the safest house is prohibitively expensive.

Exact solutions to large (and dynamic) optimisation problems are often not achievable in practice and so 'good enough' (satisficing) alternatives have to be used instead of the optimal designs (Simon, 1996). Designs do not necessarily have to be the global optimum to survive, they just have to be better than those designs around them in their niche. Note also that this means that designs do not even have to be a local optimum in design space. Newtonian mechanics was the best theory of mechanics until Einstein's theory of relativity came along, and even now it is still used. Neither theory can claim to be a complete explanation - the optimal solution.

Furthermore, because a niche is defined by both physical and ecological factors, the cost-benefit analysis of different designs is dependent on the other designs present. Maynard Smith's (cf. 1976) work on 'evolutionarily stable strategies' describes the same phenomenon: that is, in situations characterised by a conflict of interests, the best strategy to adopt depends on what others are doing.

Requirement satisfaction is not necessarily an all or nothing thing. The passenger aircraft niche, for example, is partially satisfied by a range of different designs. Different aircraft have different sizes, shapes, ranges, speeds, maximum altitudes, etc. To evaluate one design as better than another requires a basis of comparison (Sloman, 1969). For example, it is possible to build faster cars, but the cost outweighs the benefit making the slower car in some senses 'better' - safer, more economical, etc. Thus it is only possible to say that one design better satisfies a niche's requirements than another design in relation to a basis of comparison.

As with designs, some niches can be described as hierarchic. This is illustrated by the following example: as energy tends to disperse, if a dynamic structure is to remain stable, to cohere in time and space, then it requires a continuous flow of energy through the system (Atkins, 1994). There are many ways natural designs can do this - there are many ways of

making a living. For instance, a design could eat plankton in the sea, eat grass on the land or catch insects on the wing. The choice of niche leads to different sets of requirements acting on a design. The viscosity and density of water requires that fish be streamlined according to restrictive hydrodynamic design principles if they are to be both efficient and swift. Respiration in water requires a design feature to extract oxygen from the water, excretion in water is relatively easy compared with land due to the aid of diffusion of water products in the surrounding water. As some design decisions are made, other requirements arise and so on until the design is specified.

Different design features are only directly linked to certain other structures, which leads to a structural decomposition. The idea of a requirement having requirements can be illustrated by considering the eye. If a design requires a distance sensor, then the eye can fulfil that role. The eye has its own set of requirements, the lens must be transparent and its shape adjustable, the retina must be sensitive to light of certain frequencies. Each of these design features is a design which fits in a niche created by other designs. A design is not directly connected to all other designs, so a partitioning of structure can occur.

To summarise, niches consist of physical constraints - such as those embodied by the laws of physics - and ecological constraints - as defined by coexisting designs. Niches are dynamic and because they consist of *sets* of requirements, numerous trade-offs have to be made. Finally, as environments provide multiple means of satisfying requirements, designs move in niche space seeking 'better' regions.

2.5 What is adaptation: conceptual analysis

When asking a question it is important to consider the implicit assumptions being made in that question. Consider, for example, the question: How did species originate? This assumes that the concept of species refers to actual things in the real world. However, this presupposition need not be true (Emmet, 1991), for example, questions can be asked about dragons and fairies even though they do not exist. In this thesis the main question being addressed is: How is adaptation possible? This begs the question of what the term 'adaptation' refers to. Does the term refer to a specific kind of thing or to a whole lot of different things? How can the concept be related to the previous discussion of designs and niches?

One strategy in analysing concepts, suggested by Sloman (1978), is to examine the definition found in a dictionary. The Collins English dictionary (1994) gives the following definitions for the words ‘adapt’ and ‘adaptation’:

adapt vb. 1. (often foll. by to) to adjust (someone or something, esp. oneself) to different conditions, a new environment, etc. 2. (tr.) to fit, change, or modify to suit a new or different purpose.

adaptation n. 1. the act or process of adapting or the state of being adapted, adjusted. 2. something that is produced by adapting something else. 3. something that is changed or modified to suit new conditions or needs. 4. Biology. an inherited or acquired modification in organisms that make them better suited to survive and reproduce in a particular environment.

The first point to make is that the term ‘adaptation’ can be used to describe both a process and a product (Dunbar, 1982). However, it can be very confusing when the same term is used in both cases. Firstly, adaptation can refer to the processes whereby an organism comes to fit a certain environment, for example, the process of biological evolution. But also adaptation can refer to functional features of an organism, such as eyes or wings. From this point on, to avoid confusing the process with the product, the process will be referred to as adaptive change.

Plotkin (1995) analyses the concept of adaptation and relates it to the concept of knowledge. He identified two properties common to all adaptations. The first of these properties is the goal-driven or functional aspect of these structures. In the dictionary definition this is expressed by the phrase ‘to fit, change, or modify to suit a new or different purpose’. The other property Plotkin emphasises is that of ‘organismic organisation’ in relation to ‘environmental order’ - ‘to adjust (someone or something, esp. oneself) to different conditions, a new environment, etc.’ Thus an adaptation is a functional structure which is coupled with an aspect of the environment.

Plotkin argues that both adaptations and knowledge - as it is generally understood - are special cases of biological knowledge. If ‘to know’ is to incorporate the thing known - not

literally - into the knower, and if adaptations are forms of incorporation of the world into the structure and organisation of living things, then adaptations are instances of (biological) knowledge.

Dennett (1995) also makes the point that adaptations are a form of knowledge. He writes ‘any *functioning* structure carries *implicit* information about the environment in which the function “works.”’ For example, seagulls embody the principles of aerodynamic design which implies that the creature is adapted for flight in a medium having the specific density and viscosity of the atmosphere within a thousand meters of the surface of the Earth. Holland (1995) calls such knowledge, which agents use to anticipate certain aspects of their environment, an internal model. Thus another way of regarding the process of adapting is as a process of building and refining a pragmatically useful model.

Thus functional features, adaptations, are design features which satisfy certain niche requirements. The process of adaptive change involves moves in design space which produce this sort of design features. To describe a design as adaptable suggests that a design’s structure can be changed in such a way that it improves or usefully expands its capabilities which may in turn allow it to access other niches.

Plotkin’s (1995) analysis of adaptations can be supplemented by Holland’s (1994) proposed formal framework for discussing adaptive change. Holland (1994, p. 3) observes that ‘adaptation, whatever its context, involves a progressive modification of some structure or structures.’ The set of factors which are involved in this change in structure Holland calls ‘the adaptive plan’. Thus the adaptive plan determines just what structures arise in response to the environment, and the set of structures assessable by applying all the possible operators delimits the adaptive plan’s domain of action.

Holland (1994) identifies three key components associated with adaptive change: 1) the environment, E , of the system undergoing adaptive change; 2) the adaptive plan, τ , whereby the system’s structure is modified to effect improvements; 3) a measure of performance, μ , i.e. the fitness of structures in the environment. Holland’s general framework for discussing adaptive change combined with Plotkin’s analysis of the consequences of adaptive change, adaptations, provide a domain-independent way of discussing both adaptive change and adaptations.

Sloman (1978) suggests that when analysing a concept it is useful to consider a list of instances of the concept and of non-instances which are in some way similar. Here is a short list of examples:

1. water adapting to the shape of a bowl
2. the epiglottis stopping you breathing food
3. a termite nest's maintaining a certain temperature
4. a lizard moving out of the sun
5. an egg developing into a chicken
6. a child learning to walk
7. the evolution of a species
8. progress in science

The next task is to try to sort out the instances of the concept from the non-instances. Some of the items on this list are examples of moves in design space which can lead to the formation of adaptations: examples 6, 7 & 8. Evolution of a species is the paradigm case of adaptive change. The processes and mechanisms involved have been elucidated over the last hundred years (see chapters 3 and 4).

A number of parallels have been postulated between species evolution and individual learning (e.g. Skinner 1969; Piaget, 1980). However, a learning mechanism can also be seen as an adaptation which allows the individual to explore regions of design space for those designs which best meet the individual's needs. Learning is also a form of adaptive change - searching for moves in design space which improve niche satisfaction.

Example 8 is an example of conceptual evolution, another case of adaptive change (James, 1880; Campbell, 1974; Popper, 1986; Hull, 1988b). The problem forms the niche which a design (solution) has to fit. Different designs make different sets of predictions which can be tested against those facts observed in the real world. These designs are implemented in natural language or logico-mathematical terms. Different designs - theories - can via a basis of comparison (Sloman, 1969) be deemed better than other theories.

The examples 2, 3 & 4 refer to adaptations, that is, functional design features. Adaptations come in many forms - for instance, physiological, structural and behavioural

(Plotkin, 1995). The epiglottis (example 2) is an example of an internal structural adaptation. Its function is to prevent animals from breathing their food. The termite nest (example 3) is an external structural adaptation. The termite nest has a number of functions, the main one being protection from predators and the elements. The structural organisation of the termite nest causes it to maintain a certain environment - temperature, humidity, etc. Thus the termite nest creates and maintains a niche different from the surrounding niches.

The lizard moving out of the sun (example 4) is an example of a behavioural adaptation. Furthermore, this behaviour is, like the termite mound, an example of niche selection. The lizard changes its short-term niche from a sunny, hot one to a dark, cool one. By selecting a different short-term niche, the lizard has changed the requirements to fit the design. Adaptive change does not have to involve changes to the design, but can instead involve changes to the niche where the design is found.

Biologists going back to Darwin have, in studying adaptations, identified a number of origins other than adaptive change which can lead to adaptations (cf. Eldridge, 1995). Some changes are the results of allometric effects - for example, an elephant has a big brain because it is a big animal - or developmental and architectural constraints. Another class of design features has been called pre-adaptations (Dawkins, 1995) or exaptations (Gould and Vrba, 1982). These design features use functionality different from the design feature's original purpose. For example, sweat's original function was to cool an animal down, but due to sweat's smell it has come to be used as a signalling device in many animals (Dawkins, 1996). The fact scent glands are pre-adaptations can be shown by examination under a microscope where they can be seen to be modified sweat glands (or in some cases sebaceous glands) - in some animals the unreconstructed glands can still be found. Thus certain adaptations may be achieved by using design features for things other than their original purpose. This may in turn allow access to different niches. For example, one theory of the origin of flight proposes that wings were originally devices for knocking insects out of the air (cf. Dawkins, 1996); subsequent modifications allowed birds to move from ground-based to airborne niches.

Biologists studying adaptations have developed general techniques for distinguishing functional design features from accidental design features. One indication that something is an adaptation is convergence of design - homologies (Ridley, 1990; Eldridge, 1995). If a

design feature can be found which has arisen separately in many different lineages, then this adds weight to it being an adaptation. For example, estimates at the number of times the eye has evolved vary between forty to sixty (Dawkins, 1996). However, this argument is an inductive argument and does not lead to certain knowledge. Anyway, for the purpose of this thesis the origin of a functional design feature is not as important as its role in a design's functioning.

Returning to the examples, the development of an egg into a chicken is an interesting case. This example does involve moves in design space. One design changes through a series of designs and often ends up in a radically different area of design space. In the case of a human, for example, a single cell, in a specially created niche - the womb - grows into a neonate which can survive in the open, albeit with a lot of parental support, and continues to mature into an adult. But is this an instance of adaptive change or is it an adaptation? I would argue it is an adaptation, a property of a design which allows it to modify its structure to take advantage of certain niches it could not directly access. In chapter 4 development is examined in more detail.

The final example - water adapting to the shape of its container - is an example of change, but is it an example of adaptive change? Adaptations are the consequence of adaptive change. The water changes its shape under the influence of gravity, minimising its potential energy. It is hard to see what could be regarded as an adaptation. The water is changed to meet new conditions, but this change does not really suit a new or different purpose as water has no identifiable needs.

To summarise, the concept adaptation can refer to two things. Firstly, it refers to certain design features which satisfy certain niche requirements, and secondly, to the processes which produce these design features by changes to the design or niche. Adaptations are a consequence of adaptive change. Adaptations have two properties: functional - they serve a purpose - and relational - they link design features to niche requirements. There are different forms of adaptations: structural, physiological, behavioural, etc. Adaptations can have different origins: evolution, learning, science. It has been argued that adaptive changes can be created in different ways: for example, moves in design space or moves in niche space. Certain other changes are not adaptive, for example, changes which do not affect the functional organisation of a design such as the changes

brought about as side effects of many physical processes, like an engine of a car getting hot after a long drive.

The process of adaptive change can be split into two categories: firstly, the invention/discovery of new capabilities, for example, flight; and secondly, optimising the performance of design features, for example, flying faster. The previous discussion of designs and possibility transducers described how a particular implementation linked more sets of possibilities than its functionality required. The consequence of this is that existing designs can be used in new ways to perform new functions: pre-adaptations or exaptations. A side effect of pre-adaptations is that this may link two different functions to the same design feature producing an architectural constraint which affects the subsequent optimisation of those features.

2.6 Why adapt: requirements analysis

The process of requirements analysis, also called requirements definition or problem analysis, involves specifying the requirements and constraints of a system. A requirement is a specification of what a system must do; a constraint or non-functional requirement limits the possible implementations of a system (Lamb, 1988). In this section the aim is to consider the conditions which require agents to be adaptable. The requirements of an actual adaptive mechanism will be considered in chapters 3, 4 and 5. Two constraints are assumed for the purpose of realism: 1) that designs have finite capabilities; 2) that niches have finite resources.

I will argue that there are at least three cases when adaptive processes become necessary: 1) in dynamic environments; 2) where designs are incomplete; 3) in competitive environments. Let us start by considering competition. Designs do not necessarily optimally fit their niches. In all but the simplest of designs there is room for improvement, ways in which the benefits can be increased and the costs cut. Furthermore, designs do not either start with innate capabilities or start from scratch with the ability to adapt. Todd (1996) describes the similarly false dichotomy of having innate knowledge and learning from an initial complete absence of knowledge. Thus a design capable of adapting could improve on its initial capabilities.

In a resource-bound world, designs which are economically more successful relative to other designs are favoured. The measure of economic success and the cause of economic success are context-dependent, but the result in different domains is the same: a relative economic success is necessary to survive. The first argument for having adaptive processes is that they can improve your economic success. If other designs have this capability, then it can become necessary just to keep up with them.

The second requirement for adaptive processes is that a design may be incomplete. This may be due to constraints acting on the processes which produced the design. For example, biological designs are the result of inductive processes and are thus not based on certain knowledge. Also there is often a lag-time between specifying a design and implementing it. Either of these facts can lead to information being wrong, for instance, due to incorrect generalisations. Processes which correct the knowledge and keep it up to date will range from useful to essential.

Limitations on the information that can be imparted to a design can also lead to designs being incomplete. In biological evolution, for example, genetic information influences the structure, physiology and behavioural character of an organism. However, because 'greater amounts of genetic material are more susceptible to the randomising effects of mutation' (Johnston, 1982, p.339), the larger the genotype, all things being equal, the more errors are made copying it. Large behavioural repertoires transmitted by the genotype would require very complex and expensive copying mechanisms as well as mechanisms for initially coding the repertoire. A compact behavioural-repertoire acquiring mechanism may be a more cost-effective solution to this problem. Such learning mechanisms have the additional consequence of helping compensate for the 'uncertain futures problem' (Plotkin, 1995) - that is, that the initial design may be out of date by the time it is implemented.

The third class of requirements for adaptive processes arises out of environmental changes. From the point of view of designs with finite capabilities, change can be split into those that are predictable and those that are not (Plotkin, 1995). Changes which are infrequent and possibly catastrophic - for example, meteor strikes - are hard to predict. In such cases survival is largely due to chance. Other forms of change may be unpredictable, but occur frequently enough to be catered for. For example, the winter temperature is not precisely predictable, but it can be approximated. The final kind of change is predictable

change. This category includes all the sorts of change which have occurred with sufficient frequency that they can be compensated for.

Plotkin (1995) suggests that 'predictable unpredictabilities' is the core concept in understanding the evolution of intelligence as an adaptation. In cases where situations are partially predictable - for example, what can be eaten may be known, but not where to find it, or that faces can be told apart by different features, but not which faces are friends and foes - adaptive processes can compensate for incomplete information by acquiring new knowledge.

In a finite world, populations of designs can be in competition for the resources. If this competition occurs, then the designs better able to acquire the resources will partially define the niche requirements. Thus, a design may be sufficient to obtain the resources only if other designs are not present. This escalation of requirements is often termed an 'arms race'. For example, if the biggest organism wins the fight and starts to reproduce, then a requirement for reproduction will be based on the size of organisms present at that moment in time.

In summary, as both designs and niches change through time (see sections 2.4.1 & 2.4.2), then if a design is to stay fitted to a niche, it has to change too. Designs, for one reason or another, can come into the world incomplete, and adaptive processes are needed to complete such designs. Finally, competition for resources will cause designs capable of improving on their performance to be favoured and thus change the niche requirements.

There are many features in the natural world which lead to the need for an adaptive process. For example, if an organism is to reproduce, it has to survive until it can reproduce: that is, survive through development and maturation, survive locating the necessary resources to maintain its structure and produce a new structure, and survive long enough to find a mate. Survival is necessary for reproductive success and it depends on the quality of the adaptations a design can deploy. Designs with adaptive processes which refine already present adaptations and invent new adaptations will assist the design in its struggle for existence.

Holland (1994) identifies five specific obstacles to an adaptive process: 1) the space of possible structures, S , can be very large and solutions largely distributed; 2) the structures in S may be complicated so that it is difficult to determine which substructures or components are responsible for good performance; 3) the performance measure may be a complicated

function with many independent parameters; 4) the performance measure may vary over time and space so that given structures are only advantageous at certain times and places resulting in many structures having to be found and stored for use when needed; 5) the environment presents so much information to the plan that it must be filtered and sorted for relevance. These obstacles are requirements for the adaptive process which lead to a competition among adaptive processes for the one which can produce the fittest products (see chapter 4).

2.7 Summary

There are many different types or kinds of explanation. In this thesis two kinds of explanation are sought: 1) functional explanations - *what* a design must do; and 2) formal explanations - *how* a design can do it. The latter explanations will be described as algorithms, that is in terms of the logic underlying the dynamic of a design, separated from the material construction.

The design-based approach is a methodology which aims to provide the types of explanation sought. In this approach an engineering stance is taken and systems are analysed in terms of the functional requirements and possible implementations which satisfy those requirements. This is combined with the philosophical technique of conceptual analysis which helps to provide precise, theory-grounded concepts and so avoid terminological confusion.

In the design-based approach the set of functional requirements for a particular problem is called the niche. Specifications for structures which meet the requirements are called designs. The design features which meet functional requirements are called adaptations. These features can be the consequence of adaptive change and can be produced by moves in both design and niche space.

There are a number of conditions which make an adaptive process beneficial or in some cases necessary:

- designs have finite capabilities and are thus limited in what they can predict
- designs may be incomplete solutions and may need processes to finish them off
- niches have finite resources, thus there can be a competition for resources which change the requirements

- niches are not static, but they change in time which affects the fit of the designs
- designs may develop, accumulate resources and seek mates, which requires an array of design features to perform these actions
- there may be a delay between design specification and implementation in which time the requirements change
- there may be limitations to what information is available to the ‘designer’ (design process) and subsequently to the design itself
- designs can also be resource limited

The central questions in this thesis are: What is adaptation? How is adaptation possible? These questions can be refined in the light of the discussion in this chapter. The question of what adaptive change is, can be rephrased as what processes produce functional design features which improve a design’s fit to its niche. Adaptive change relates to certain moves in design space which produce functional design features or moves in niche space to areas where niches are better satisfied by the design. The question of how adaptive change is possible is partly answered by the variety of quality of fit between designs and niches which leads certain designs to be, in some sense, fitter.

Holland (1994) describes what can be regarded as the minimally sufficient universal adaptive mechanism: the enumerative plan. The order in which structures are tested in the enumerative plan is unaffected by the outcome of previous tests. Thus it generates structures, preserving the fittest ones as it goes. While the enumerative plan works in principle, in many cases the size of the space of possible structures makes it uninteresting in practice. What is needed is *selective* trial-and-error (Simon, 1982), where brute-force search is replaced with a combined system of search and ‘reason’ thus eliminating a substantial amount of trial-and-error from the search process. The next two chapters describe an actual design which is capable of producing adaptations: biological evolution. The evolutionary process and its mechanisms will be described from an algorithmic perspective and its essential properties examined. The aim of the next two chapters is to describe the mechanisms which usefully constrain the evolutionary trial-and-error process.

3. Abstract evolution

3.1 Introduction

Since Darwin proposed his ideas of evolution in 1859, there has been much controversy surrounding them. Many of the arguments against Darwin have been religious or political and some scientists remain unconvinced - John Von Neumann and Fred Hoyle to name but two. The aim of this section is to describe some of the essential evolutionary concepts in a logical form, not to give an exhaustive review of evolutionary thinking. It is hoped that by describing the evolutionary theory in a different form, some of its features that have confused people can be avoided. In the next two chapters the role of development, reproduction and behaviour in the evolutionary story will be explored.

The structure of this section follows from an observation by William James (1880). James differentiated between the causes which originally produced a peculiarity and the causes that maintained it after it had been produced. James pointed out that Darwin had separated the causes of production, under the title of ‘tendencies to spontaneous variation,’ and confined his attention to the causes of preservation: selection. In the first half of this chapter Darwin’s original idea - the causes of preservation - will be described in an algorithmic form. In the second half of the chapter the causes of production will be examined as described in the neo-Darwinian formulation.

Algorithmic formulations of evolutionary processes are not new - they have been extensively used, for instance, in the disciplines of artificial life and evolutionary computation. There are at least three advantages of algorithmic descriptions: firstly, they are a type of formal description which is more precise than natural language description; secondly, such descriptions can be run on computers, which allows predictions deduced from the model to be tested using computational experiments; finally, because algorithmic descriptions derive their power from the logical structure and not the material makeup of a thing, they allow the application of the ideas to other domains which are logically equivalent.

The novel contribution to research in this chapter is to provide a comprehensive description of evolution from a computational stance and use this to start to elucidate the concept of evolvability.

3.2 The Darwinian process: Darwin's dangerous idea

The foundations of modern evolutionary thinking were laid by Charles Darwin (1809-1882) and Alfred Russell Wallace (1823-1913) with their joint communications on the theory of evolution by natural selection presented in 1858 and published subsequently in the 'Journal of the Proceedings of the Linnean Society'. In November 1858, Darwin published his book *On the Origin of Species by Means of Natural Selection, or The Preservation of Favoured Races in the Struggle for Life* in which he set out his ideas in full. The essence of Darwinian thinking is succinctly summarised at the beginning of *The Origin of Species*:

'As many more individuals of each species are born than can possibly survive; and as, consequently, there is a frequently recurring struggle for existence, it follows that any being, if it varies however slightly in any manner which is profitable to itself, under the complex and sometimes varying conditions of life, will have a better chance of surviving, and thus be *naturally selected*. From the strong principle of inheritance, any selected variety will tend to propagate its new and modified form.' (Darwin, 1859, p. 68)

This has led to the standard textbook description of Darwin's idea as arising from a series of observations and deductions (cf. Williams, 1995). Darwin's first observation was that individuals give rise to many more offspring than is necessary to maintain a constant population size. Darwin was influenced by the economist Thomas Malthus who had suggested, many years earlier, the inevitability of population explosions due to the geometrical growth of populations, and the inevitability of famines due to populations growing too large for the available resources. However, as Darwin observed, population sizes do not grow exponentially. From this Darwin deduced that because of a competition for resources there was a *struggle for existence*.

Darwin's second purported observation was that there is significant variation within populations. From his studies of natural and artificial populations, Darwin thought it was highly probable that variation had occurred that was useful to an individual's welfare. From the competition premise and the variation premise, Darwin deduced that if any advantages were enjoyed by any of the contestants, then this could bias the sample that reproduced.

Darwin's third observation was that 'like produced like' - that organisms produced not only the same type of organisms, but organisms which were more similar to themselves than to other members of the species. The fact that 'like produced like' Darwin called 'the strong principle of inheritance'. Dennett (1995, p. 41) describes this principle in the following way: 'if offspring tended to be more like their parents than like their parents' contemporaries, then biases created by advantages, however small, would become amplified over time, creating trends that could grow indefinitely'.

Darwin's final deduction was that if variations useful to the individual do occur, then these individuals will have the best chance of being preserved in the struggle for life. These individuals will via the strong principle of inheritance tend to produce offspring of a similar character - that is, with the useful variations. This principle of preservation is called *natural selection*.

Darwin summarised his ideas as a formal argument - if the conditions are met, then a certain outcome is assured (Dennett, 1995). This point is made by Dawkins (1983b) and referred to as Universal Darwinism: if the conditions are met anywhere in the universe, then the consequence will be the same - natural selection will occur. Some researchers (cf. Dawkins, 1983b) have speculated that the Darwinian law may be as universal as the great laws of physics.

3.2.1 The algorithmic nature of the Darwinian process

Darwin had two objectives in the *Origin of Species*: the logical demonstration that a certain sort of process would necessarily have a certain sort of outcome (see section 3.2), and the empirical demonstration that the requisite conditions for that sort of process had been met in nature (Dennett, 1995).

Darwin's first objective was to give, what would be called today, an algorithmic description of the process of evolution. An algorithm is a formal process that can be counted on logically to produce a certain sort of result whenever run or instantiated. Computer scientists sometimes restrict the term algorithm to programs that will terminate (Dennett, 1995, p. 50) - that have no infinite loops in them, for instance. Many programs in the real world are not algorithms if the termination restriction is used, for example, prime number generators, but their subroutines - which terminate - would be called algorithms. The

termination restriction is ignored in this section: that is, a program does not have to terminate for us to call it an algorithm.

Dennett (1995) described three key features about algorithms which are relevant to the Darwinian process. The first feature is substrate neutrality. The power of the procedure is due to its logical structure, not the causal powers of the materials used in the instantiation, just as long as those causal powers permit the prescribed steps to be followed exactly.

Dennett's second key algorithmic feature is underlying mindlessness. Each constituent step, as well as the transition between steps, is utterly simple - a breakdown of the process into dead-simple steps, requiring no wise decisions, delicate judgements or intuitions on the part of the algorithm following machine. The process forms a number of steps which can be easily mechanised.

The third algorithmic feature Dennett identifies as important to the Darwinian algorithm is that it produces a guaranteed result. An algorithm will, if executed correctly, always produce the same output in the same circumstance.

As Dennett (1995) points out, any finite process can be described as an algorithm. The main advantage of describing the Darwinian process as an algorithm is that it explains how impressive products, like all the different forms of animals and plants, can be produced by a 'set of individually mindless steps succeeding each other without the help of any intelligent supervision' (Dennett, 1995).

The Darwinian process is a type of algorithm computer scientists call probabilistic. Gould (1997, p. 36) has argued that, 'The events of our complex natural world can be divided into two broad realms: repeatable and predictable incidents of sufficient generality to be explained as consequences of natural laws, and uniquely contingent events that occur, in a world of both chaos and genuine ontological randomness as well, because complex historical narratives tend to unfurl along the pathway actually followed, rather than along any of the myriad equally plausible alternatives.' The Darwinian algorithm can be described as probabilistic for two reasons: first, due to unpredictable environmental change - such as a meteor strike to most organisms; second, due to the genetic mechanisms - such as random mutation (see section 3.3.2). While fitness biases success, it alone does not determine it, for example, fitness now is no guarantee of fitness in the near future.

The Darwinian process is also a member of a class of algorithms that computer scientists call anytime algorithms: that is, an algorithm that produces a result whose quality is a function of the time spent running. Anytime algorithms can be classified as either contract or interruptible (Zilberstein, 1996). Contract algorithms only produce a result after a contracted time period; interruptible algorithms will produce a result at any time, the quality of the result being a function of the time before interruption. The Darwinian algorithm is an example of an interruptible algorithm. How well a species fits a niche is partly a function of the time it has spent in that niche and also a function of how friendly the niche is: the requirements of some niches are more specific than others, for example, Antarctic tundra versus rain forest .

To summarise, the evolutionary process is a probabilistic, anytime algorithm. Furthermore, note that the Darwinian process is a parallel process in two senses: first, a number of organisms of the same type are ‘tested’ at one time - for example, same or similar species competing for the same niche; second, a number of different types of organisms are ‘tested’ at one time - for instance, entirely different types competing for different niches.

3.2.2 Abstract formulations of the Darwinian process

A number of abstract formulations of the Darwinian process have been produced. In this section we will consider two of them. The American biologist R. C. Lewontin (1970) proposed a formulation of the Darwinian theory comprising three principles:

1. Different individuals in a population have different morphologies, physiologies, and behaviours (phenotypic variation).
2. Different phenotypes have different rates of survival and reproduction in different environments (differential fitness).
3. There is a correlation between the fitness of parents and the fitness of their offspring (fitness is heritable).

Lewontin’s formulation explicitly characterises the evolutionary process as a general, mechanism-independent process. This general description does not mention a number of things: 1) the mechanisms of inheritance or variation; 2) reasons for differential success;

3) the specifics of the individuals and population involved. Furthermore, as Lewontin (1970) points out, it is not necessary that resources be in short supply for there to be a struggle for existence. For example, natural selection occurs even when two bacteria strains are growing logarithmically in an excess of nutrient broth if they have different division times (Lewontin, 1970). The logical conclusion is that any population where these three principles hold will undergo transformation through evolution.

However, there is a potential misinterpretation of Lewontin's third principle: that is, that self-replication is necessary for natural selection. Consider the following thought experiment described by Braitenberg (1994). In his scenario there are numerous different robot vehicles which are capable of movement and of sensing their surroundings. These vehicles are placed on a table in front of a human robot-builder and allowed to move about the table top. The robot-builder regularly picks, at random, a vehicle from the table top and makes a copy of it; however, the process of making a copy is not exact and small mistakes are made. Robots that fall off the table are deemed dead and are never copied by the robot-builder.

In Braitenberg's scenario, Lewontin's first two principles are clearly observed - that of phenotypic variation and differential fitness. Furthermore, the third principle is also observed - fitness is heritable. However, the vehicles are obviously not self-replicators. The copying mechanism has been centralised, in this case in the form of the human observer. The important point is that Lewontin's third principle does not only apply to self-replicating entities, it also applies to other forms of replication.

The procedure followed by the human robot-builder in Braitenberg's scenario Campbell (1974) calls 'the blind-variation-selective-retention process'. Campbell's formulation of the evolutionary process is similar to Lewontin's, with a slight change of emphasis. It also comprises three principles:

1. a mechanism for introducing variation
2. a consistent (explanation follows) selection process
3. a means of preserving and/or propagating successful variants

The first principle, a need for generators of variation, is the same as Lewontin's first principle. The second principle emphasises the need for a certain degree of stability in the environment: the rate of change in the environment needs to be slow enough so that the successful properties in one generation are related to the properties found useful in the next generation. The final point, preserving and/or propagating variants, emphasises that successful (according to the selection process) variants need to be maintained in the population to achieve a cumulative advantage.

The blind-variation-selective-retention process is also called the generate-test-regenerate (GTR) heuristic or procedure (Plotkin, 1995). The GTR procedure and Lewontin's three principles are domain-independent descriptions of the same adaptive algorithm. Certain features are left unspecified, for instance, the variation producing mechanisms and the fitness evaluation mechanisms, but essentially, if these conditions are met and the procedure is followed, then evolutionary change will occur.

As mentioned in the chapter introduction, James (1880) pointed out that Darwin's scheme emphasised the independence of the variation producing mechanism from the selection mechanism. This distinction can be related to the exploitation-exploration distinction in search algorithms. Natural selection provides the exploitation mechanisms and the heredity mechanisms discussed in the second half of this chapter provide the exploration mechanism. To illustrate the generality of this formulation Lewontin's principles and the GTR procedure will be applied to explaining change in learning and science.

The reinforcement theories of Skinner, and Thorndike before him, viewed reinforcement as an extension of natural selection. It is interesting to note that Thorndike did his famous work on trial-and-error learning in cats as a student of William James, in James' basement (Schull, 1996). James (1880) was the first to explicitly argue that mental life was governed by the same principles of variation and selection as biological life. If we regard behaviours as the basic unit, then phenotypic variation requires there to be differences between behaviours. Differential fitness requires a basis of comparison, a goal - for example, getting food or finding a mate. If different behaviours produce different effects and if some of these behaviours are useful by achieving certain goals in certain contexts, then the behaviours known to achieve goal satisfaction should be used.

The final principle, that fitness is heritable, is often the hardest to demonstrate in a particular process. However, as we have seen, variants do not need to be self-replicators for fitness to be heritable. If the forms of the new variants are *informed* by the fitter variants of the previous generation, then fitness can be regarded as heritable. If, for instance, when trying to solve a problem you utilise information gleaned from previous trials, then this third principle can be regarded as satisfied. All that is now required for adaptive change is that the GTR heuristic is followed. Thus a Darwinian process is a sufficient mechanism for producing at least some types of learning.

One view of science has been to describe it as a Darwinian process (e.g. Popper, 1986; Hull, 1988b). In this Darwinian process the basic units are conceptual structures with phenotypic variation referring to, for example, the different predictions of the theories. Differential fitness could be assessed in relation to the empirical validity of the predictions of different theories. Again the problem arises with heredity. Is there a sense in which theories are informed by previous theories? Both of Simon's (1996) forms of selectivity can be observed: past theories are used to guide the search for new theories and analogous cases often provide information about possible solutions. Again if the GTR heuristic is followed, then evolutionary change will follow.

As stated at the start of this section, this is not an exhaustive view of the evolutionary theory, but an overview with a specific purpose: to show that the evolutionary process is an algorithmic process with a certain logical form and certain limitations. In the *Origin of Species* Darwin described the set of conditions which he argued were sufficient and necessary for evolutionary change to occur. Since then a number of theorists have produced abstract formalisations of the process (e.g. Lewontin, 1970; Campbell, 1974). The abstract formulations of the Darwinian process allow it to be used to explain other areas of adaptive change: for example, learning and science.

3.3 The heredity mechanisms: the neo-Darwinian formulation

In the 1940's evolutionists and geneticists reconciled Gregor Mendel's original findings on heredity with the theory of evolution described by Darwin. This 'modern synthesis' accounted for the origin of genetic variation as mutations in DNA as well as for the rearrangement of structures in a process called recombination. Subsequent discoveries about

the nature of DNA and the ability to manipulate this molecule, even to the point of inserting foreign genes into animals and changing their form and behaviour, justified the position taken in the modern synthesis.

Biologists distinguish between the genotype and the phenotype of an organism. The genotype is the genetic code which, when interpreted, produces a phenotype. The phenotype refers to the agent produced by the interaction of the genotype and environment during development. Natural selection, as described in the previous section, acts on already present phenotypic variation. In this section the origin of this variation is examined.

There are two main objectives in the next two sections. The first one is to show how the genetic code is an arbitrary, digital code (arbitrary in the sense that the mapping between the genotype and the phenotype is just one of many possible mappings which could have evolved). The second objective is to examine the mechanisms which operate on the genetic code and produce variation. By showing that the genetic code is an arbitrary, digital code, it follows that the theorems produced from work done using bit strings in the evolutionary computation research community (described in section 3.3.2) are directly applicable to biology.

3.3.1 The genetic code

The genetic code is often described as arbitrary (e.g. Ridley, 1990): that is, the mapping is one of a set of possible mappings, but due to a historical accident it has the specific mapping found today. One consequence of this arbitrariness is to uphold Popper's (1986) theory of irreducibility and emergence as the fact that the code is arbitrary means it is only one of a set of codes and thus only one of the possibilities which can be derived from chemistry. It should be noted that some biologists do not agree that the code is arbitrary, thinking instead that the current mapping is logically necessary. However, I think the evidence - described in this section - supports the first position. In this section Hofstadter's (1986) demonstration of this is discussed. Note that it is not necessary to be familiar with the technical biological terms in the next two paragraphs to understand the argument of why the genetic code is arbitrary.

The genetic code forms a mapping between two mutually unrelated domains. In the biological world both domains are made up of chemical units: the nucleotides and the amino

acids (proteins). The nucleotides are passive agents, like words on a page. The proteins are the active agents capable of expressing genes. Ribosomes, present in every cell, are responsible for translating between the two languages. In a process called transcription, messenger RNA (mRNA) copies a sequence of DNA. Then in a process called translation, mRNA binds to a ribosome bringing each 'word' of mRNA together with its complementary transfer RNA (tRNA), which in turn codes for a single amino acid.

The important point is that the DNA incorporates coded versions of all the tRNA molecules, all the necessary enzymes and the details of the ribosomes. This means mutations in the DNA can alter the decoding apparatus, changing the interpretation of the code - hence the code is arbitrary even as far as its own interpretation mechanism is concerned.

The code is now known to be made up of 64 nucleotide triplets which map onto 20 amino acids and a punctuation mark. Furthermore, the code appears to have some adaptive features (Maynard Smith and Szathmary, 1997): 1) chemically similar amino acids tend to be coded for by similar codons¹; 2) amino acids are coded for by between one and six different codons, with amino acids common in proteins tending to be specified by more codons. These adaptive features of the genetic code can be explained by natural selection acting on different initial codes when the amino acid-nucleotide mapping was assigned.

One reason for mistakenly thinking the genetic code may not be arbitrary was that the same code is used throughout the natural world: from humans to bacteria. However, Darwin's theory of descent with modification predicts such homologies: structures inherited from an earlier form, for example, the pentadactyl limb. But why has the code not changed? This is probably because, while it may be useful to change the assignment of one triplet to another (changing the amino acid it codes for), it would probably be a disadvantage to change all of the assignments in a particular genotype (Maynard Smith and Szathmary, 1997). What is important about the genetic code being arbitrary is that, as other codes are possible, the code can change albeit rarely and with difficulty in nature (Maynard Smith and Szathmary, 1997).

The genetic code is not only arbitrary, it is digital like the codes of computers and compact discs (Dawkins, 1995). However, whereas computers and compact discs use a binary digital code, DNA implements a quaternary code. A digital code leads to the

observations noted by Mendel: namely, inheritance is in a sense discrete (particulate) and not continuous (blending). However, while the genetic code is discrete, the effects can be continuous due to the interaction of enzymes and environmental factors.

There are functional reasons for presupposing a digital code would be used in evolution (Dawkins, 1995; Maynard Smith and Szathmary, 1997). The alternative type of code is an analogue code, like that used in the old telephone exchanges. However, messages encoded in continuously varying symbols rapidly decay into noise as they are transmitted from an individual to an individual (Maynard Smith and Szathmary, 1997). The Darwinian process requires many copies of the code to be made and high fidelity is required if like is to produce like. A property of digital codes is that high fidelity copying mechanisms are easier to obtain and that digital codes are more susceptible to analysis and manipulation.

Having identified the genetic code as an arbitrary, digital code another question can be asked: namely, why have a code in the first place? It is presently believed that in the early stages of the origins of life, the RNA molecules involved were both the genotype and the phenotype (Maynard Smith and Szathmary, 1997). Also much of the work in evolutionary computation has used simple one-to-one mappings between the genotype and the phenotype (Ackley and Littman, 1994). This fact, however, points to a functional reason for having a code: that is, the convenience of applying the genetic operators. In other words, the code is symbolic - one domain denotes something in the other - and changes made to one domain can lead to changes in the other.

3.3.2 Genetic algorithms and the Schema Theorem

In the last section the genetic code was shown to be an arbitrary, digital code. In this section we will focus on the role of the genetic code in the production of variation. One of the first mathematical analyses of natural selection was provided by Fisher (1930). Fisher's treatment of natural selection was concerned with the rates of change of individual genes in populations over time (Bonner, 1980). However, Holland's (1995) account of his Schema Theorem will be described in this section for three reasons: first, it is an extension to

¹ Coding unit of messenger RNA comprising a triplet of nucleotides

Fisher's results as it focuses on coadapted sets of alleles with epistatic² interactions (Holland, 1994); secondly, algorithmic descriptions are sought in this chapter and Holland's work is algorithmic by nature; thirdly, because Holland's results can be neatly summarised to show the function and roles of the different genetic operators in the production of variation.

Before continuing with an exposition of Holland's work, let us briefly consider what is meant by the term 'fitness' in this context (cf. Dawkins, 1983a, chapter 10). Fitness refers to an aspect of a design's relation to a niche, such that the better the design meets the niche's requirements, the greater the design's fitness: this, I think, was Darwin's original idea: fitness-as-niche-satisfaction. If fitness-as-niche-satisfaction biases reproduction, then this leads to Williams' (1966) view of fitness-as-reproductive-success. However, while reproductive success is a consequence of fitness-as-niche-satisfaction and measuring the relative numbers of offspring allows 'fitness' to be quantified, due to contingency effects fitness-as-niche-satisfaction is not a sufficient condition for reproductive success. Note, in genetic algorithm research the term 'fitness' usually refers to fitness-as-niche-satisfaction.

Let us start by considering the biological case before abstracting and analysing it mathematically. The chromosomes which make up the genotype are made up of strings of genes. Each gene can have a number of different forms, called alleles. The standard approach to assessing the fitness contribution of a particular allele is using the techniques developed in classical mathematical genetics. In this approach it is assumed that each allele contributes something positive or negative to the overall fitness of an organism. The fitness contribution is estimated by looking at the average fitness of all the individuals carrying that allele. The overall fitness of a genotype would be a sum of the contributions of its constituent building blocks (Holland, 1995). However, as Holland indicates, there are two major difficulties with this approach: first, different alleles have different effects in different environments; secondly, alleles interact. Thus the fitness in a given environment is a non-additive function of the alleles present.

Holland (1994, 1995) suggests a way of thinking about building blocks made up of multiple alleles. If a chromosome is a string of length L , made up of a series of genes, each

² Interaction of non-allelic (allele = representatives of the same gene position - locus) genetic elements or their products

with one of two alleles, then there are 2^L possible strings. A building block - schema - is a number of alleles at different positions on the string: 1st, 2nd and 5th positions, for instance. The only data about fitness relates to the fitness of a whole string. The heuristic for calculating the fitness contribution of a schema, suggested by Holland, is to calculate the average fitness of strings with the schema present and divide this number by the mean string fitness.

Evolution implicitly carries out a similar calculation as reproduction is biased by fitness-as-niche-satisfaction. This can be shown in a simplified mathematical scenario. In this scenario the fitness of a string directly determines the number of offspring and the average string fitness is 1. If there are three instances of one schema in strings with strengths 1, 0 and 1, then the schema's fitness is $2/3$. In another case a schema is in strings with fitnesses 2, 1 and 2 and thus has a fitness of $5/3$. Above average schemata will increase in the population, below average ones will be removed. Mathematically, this can be postulated as (from Holland, 1995):

$$M(b, t + 1) = S(b,t)M(b,t)$$

Where $M(b,t)$ is the number of instances of schema b at time t and $S(b,t)$ is the average fitness of b at t . As Holland points out this is exactly the result required: that is, increase in the number of fitter schemata (those with the biggest fitness contribution).

However, there is a serious limitation in this scenario. As the length of the string becomes bigger, there is an exponential increase in the number of possible strings so that sorting through all the strings becomes an intractable task. Also this scenario is limited to the initial population of schemata as there is no mechanism for introducing variation.

Crossover can recombine schemata without greatly disturbing the desirable outcome of increasing the frequency of the fitter schemata in the population. In crossover a random point is picked. In the mathematical form, if L is the length of a string and $L(b)$ is the length of schema b , then $L(b) / (L - 1)$ is the probability that crossover will fall within the outer limits of b , and $1 - L(b) / (L - 1)$ is the chance that crossover will not fall with the outer limits of b . If we assume that every crossover falling within the outer limits actually disrupts the schema, then $1 - L(b) / (L - 1)$ is the chance that the schema will not be disrupted. For

example, in a string of length 10, a schema covering 4 genes has a probability of being disrupted of $4 / (10 - 1) = 4/9$ and a probability of not being disrupted of $5/9$. Thus for schema b of length $L(b)$ in a string of length L the chance of b entering the next generation is:

$$M(b, t + 1) = [1 - L(b) / (L - 1)]S(b,t)M(b,t)$$

Longer schemata have a higher chance of being broken up, but if they are made up of smaller, fitter schemata, this is not such a problem. If two parents have identical copies of the same schemata, then crossover cannot disrupt them. It follows that recombination rarely disrupts longer schemata that are made up of combinations of shorter, above average schemata. If some of these long schemata are above average, then they will spread through the population.

However, in this scenario when an allele no longer occurs in the population, the action of reproduction and crossover cannot replace it. Under these circumstances the allele is said to have gone to fixation. Each time this occurs in a string where each gene has 2 possible alleles the search space is reduced by $1/2$. Point mutation can be viewed as a way of injecting variety back into the system - an ‘insurance policy’ against premature fixation. By randomly flipping a bit, new schemata are created and can be subsequently tested. The chance of a schema increasing in the population where $P_{mut}(b)$ is the probability a mutation will modify schema b becomes:

$$M(b, t + 1) = [1 - L(b) / (L - 1)][1 - P_{mut}(b)]S(b,t)M(b,t)$$

Before considering the effects of combining reproduction, crossover and mutation I want to mention a consequence of point mutations. The phenomenon in question is called ‘random drift’. Random point mutations in finite populations can lead to different characteristics fluctuating by chance. The smaller the population, the greater the fluctuations are in frequency until ultimately one type can become fixed (only available type) by chance alone. For example, Maynard Smith (1994) mathematically shows the effect of random drift in an asexual population of N individuals, with separate generations. If the number of

offspring produced by an individual has a Poisson distribution, then the number of generations until all individuals are descended from the same individual is $2N$, with a standard error a little greater than N . The important point is that a certain amount of *random* change is a consequence of the Darwinian process due to point mutations.

Returning to the previous discussion, reproduction, crossover and mutation can be combined with the following results: reproduction biased by fitness-as-niche-satisfaction causes an increase in above-average, fitness contributing schemata in the population. Crossover generates offspring which are different from their parents by recombining schemata passed on in reproduction. This process can disrupt schemata, but tends not to if the schemata are either short or made up of above-average short schemata. Mutation acts as an ‘insurance policy’ to prevent alleles becoming fixed under the effects of differential reproduction and recombination. Furthermore, mutations produce hill-climbing search which is very good at local, myopic optimisation, while recombination provides ‘genetic search’ which is particularly good at obtaining evidence about what confers fitness from widely separated points in the search space (Hinton & Nowlan, 1987). Combining these operators produces a more effective search algorithm than just having the operators alone.

Combining recombination and mutation also avoids a problem that has been dubbed ‘Muller’s ratchet’ (cf. Cohen & Stewart, 1995). J. H. Muller pointed out a serious problem with asexual reproduction - that is, with heredity and mutation alone - because successive mutations tend to degrade the quality of the product. This follows from the observation that most mutations are bad, thus subsequent mutations are more likely to be bad than reverse previous bad mutations, and degradation occurs. However, recombination provides a way of both bringing together good sets of schemata and eliminating bad schemata as crossover can combine the good sets from two individuals or produce individuals with bad sets which are subsequently selected against.

Holland (1995) writes, ‘The most important feature of a genetic algorithm is its ability to carry on this sophisticated manipulation of building blocks by acting only on whole strings’. While the whole string operations (reproduction, crossover and mutation) do not directly deal with schemata and carry out no explicit computations involving them, genetic algorithms act as if such computations were being made and exploited. The ability to

manipulate large numbers of schemata implicitly through the explicit manipulation of a relatively small number of strings Holland calls ‘implicit parallelism.’

There are some simplifications implicit in this description of genetic search. One simplification is that the effects of other genetic operators have been ignored: for example, inversions, deletions and duplications. Inversions occur when ‘a piece of chromosome detaches itself at both ends, turns head over heels, and reattaches itself in the inverted position’ (Dawkins, 1989b, p. 31). Inversions may have the following function (Dawkins, 1989b; Holland, 1994): Holland’s Schema Theorem has shown that the distance apart of different alleles affects their chances of being transmitted together into the next generation, the inversion operator functions to selectively increase the linkage (decrease the length) of schemata. Holland (1994) showed mathematically that inversion can decrease the length of schemata exhibiting above-average performance, and it does this in an intrinsically parallel fashion. The genetic operators of deletion and duplication are considered in the next chapter (see section 4.2.4).

Another feature of the genotype, which has been ignored in this discussion, are the large stretches which do not appear to code for anything. These stretches could play a functional role in both recombination and inversion. First, in recombination these stretches could help separate sequences, which means that there would be a larger number of possible crossover points which will not disturb schemata. Second, these non-coding stretches could stop inversion from destroying the important sequential information needed in crossover: that is, if inversion has moved the genes about in such a way that crossover between genotypes leads to a resulting genotype with two sets of some genes but none of others, then this will probably cause problems for the offspring. The non-coding buffer areas would allow some rearrangement without disrupting the sequential information.

3.3.3 Abstract formulation of the mechanisms

An alternative to the Lewontin-Campbell formulation (see section 3.2.2) of the Darwinian process is to focus on the entities involved rather than the process of selection itself. This section starts with an examination of Dawkins’ (1982, 1983a, 1989b) replicator-vehicle formulation, followed by a discussion of Hull’s (1988a) problem with the term ‘vehicle’ and finishing with Williams’ (1992) theory of two domains.

George C. Williams (1966) influenced evolutionary biologists with his seminal account of selection occurring at the genetic level. One biologist so influenced was Richard Dawkins who subsequently postulated that the evolutionary process involved two theoretical entities: replicators and vehicles (Dawkins 1982, 1983a, 1989b). Dawkins (1989b, p. 254) writes, ‘the fundamental units of natural selection, the basic things which survive or fail to survive, that form lineages of identical copies with occasional mutations, are called replicators.’ Dawkins identifies three attributes of good replicators: 1) longevity - they must be long-lasting; 2) fidelity - capable of making good copies of themselves; 3) fecundity - fertile in making as many good and long-lasting copies of themselves as possible.

Dawkins (1982, 1983a, 1989b) regards replicators as anything in the universe of which copies are made. For example, Cairns-Smith (1991) put forward a hypothesis about the origin of life which involved an inorganic replicator: clay crystals. Crystals dropped into a solution of the same chemical will replicate their atomic structure. Also the growing crystal will from time to time break into pieces under its own weight. This process involved a kind of heredity: for example, flat crystals will produce flat crystals. Note, however, that this process does not involve a separate genetic code, the clay crystal is a replicator or as Cairns-Smith (1991) calls it a ‘naked organism.’

Dawkins (1982, 1983a) distinguishes two classes of replicating entities: passive and active. Active replicators have influence over their probability of being copied, passive replicators do not. Dawkins also suggests that something which appears at first sight to be a passive replicator - for example, a Xeroxed sheet of paper - may in fact be an active replicator - whether the piece of paper becomes Xeroxed depends on what is written on it. Braitenberg’s robot vehicles (see section 3.2.2) are a case of active replicators as they have an influence over their chance of being copied - in this case, if they can stay on the table.

Another subclassification of replicators is into germ-line and dead-end replicators (Dawkins, 1982, 1983a). A germ-line replicator is the potential ancestor of an indefinitely long line of descendant replicators; a dead-end replicator is not. For example, the DNA in a zygote is a germ-line replicator, while the DNA in a liver cell is a dead-end replicator (unless, of course, the DNA in the zygote and the liver cell is viewed as the same DNA, the same type).

Dawkins (1982, 1983a, 1989b) proposes the gene as a paradigm case of an active, germ-line replicator. However, the term 'gene' has two separate uses related to the dual role of the genotype. Firstly, the genotype is involved in reproduction, in this sense a gene refers to differences in character which can be mapped onto specific differences in DNA - like albinism or Parkinson's disease (Cohen, 1995). Secondly, the genotype is involved in development and in this context a gene refers to the DNA which codes for a single protein. Subsequent discoveries have shown some proteins are coded for by pieces of DNA distributed throughout the chromosomes - hence Dennett (1995) suggested the term 'virtual gene' would be more appropriate. To avoid terminological confusion, it is better to use the term 'replicator' instead of 'gene' in the evolutionary context, and just use 'gene' in the developmental context. Holland's schema (1994, 1995), discussed in section 3.3.2, can easily be generalised to richer structures than a linear schema (as is common place in computer science) and provide a more precise description of replicators.

To introduce Dawkins' second theoretical entity, the vehicle, let us briefly return to Cairns-Smith's (1991) clay crystals. Clay crystals have complex surfaces with many needle-like crystal projections. Those structures contain irregularities, dislocations, at which the regular lattice structure of the crystal is broken. Because crystals grow by extending their atomic lattices, the dislocations are repeated. There is even the possibility of mutations: spontaneous dislocations. But what has this to do with vehicles? Well, if the surface of these crystals provides the right conditions to catalyse other chemical reactions and if these chemical reactions influence the replication of that clay crystal - for example, increasing fidelity, fecundity, or longevity - then those clay crystals will have an advantage over clay crystals without this chemical assistance. This chemical catalyst of replication is a vehicle.

There are two points I want to make before considering vehicles in more detail. First, if a replicator arises, it enters a cyclic process which will keep on going unless the conditions change to make replication impossible. Kauffman (1993, 1995) is at pains to demonstrate that these autocatalytic sets are likely to arise quite naturally and quite often. Furthermore, there is no reason or purpose being suggested for this replication. The clay crystals, for example, are pure replicators which can have certain accidental properties that subsequently influence their rate of replication.

The second point I wish to make relates to what Cairns-Smith (1991) calls 'genetic take-over'. That is, when one process can give rise to another which would have been unlikely or impossible to have arisen by itself. Replicating cycles could arise that produce more and more copies of a molecule, even though that molecule cannot replicate unaided. For example, a replicating cycle of nucleic acids - RNA, or possibly DNA - could get started by using clay as a scaffolding (Cohen & Stewart, 1995). There is a distinction here between order-from-disorder and order-from-order. By reducing the problem that order-from-disorder (self-organisation) has to solve, it is then possible to postulate how this new found order can give rise to further order-from-order.

Returning to Dawkins' theoretical entities, the vehicle is conceived as the survival machines which replicators travel in. As Plotkin (1995, p. 96) observes, 'The vehicles are the instruments of replicator preservation. More accurately, because of selection, the vehicles are the instruments of differential replicator preservation.' There are two ways to characterise evolution (Dawkins, 1982, 1983a), both of which are correct. Firstly, evolution is the external and visible manifestation of the differential survival of alternative replicators; secondly, vehicle selection is the process by which some vehicles are more successful than other vehicles in ensuring the survival of their replicators.

After Dawkins (1989b - first published in 1976) defined 'vehicles' in *The Selfish Gene* he went on to question in *The Extended Phenotype* (Dawkins, 1983a) what the term 'vehicle' actually refers to and how discrete an entity a vehicle actually is (Maynard Smith, 1982). Dawkins takes the position that the vehicle is what selection acts on and the replicator is what is selected. There are a number of possible levels of selection - that is, a number of possible vehicles - but in each case it is the replicator - for example, the gene - that is selected. Thus Lewontin's (1970) units of selection are in fact levels of selection (Dawkins, 1982). For example, an organism is not a replicator whereas an organism's genome is because changes to a replicator are passed on to its descendants, and in the case of an organism this is not true - acquired characteristics are not inherited.

Dawkins (1989b, p. 253) states that the central thesis in *The Extended Phenotype* is, 'An animal's behaviour tends to maximise the survival of genes 'for' that behaviour, whether or not those genes happen to be in the body of the particular animal performing it. ... the theorem could apply, of course, to colour, size, shape - to anything'. Natural selection

favours those genes that manipulate the world to ensure their own propagation. For example, while many of the members in social insect colonies do not reproduce, the genetic relatedness between individuals leads to the indirect propagation of their own genes. Another similar example are the cells in an individual: while the majority of cells are dead-end replicators, their genes are identical to the germ-line replicators and so they are transmitted into the next generation. These vehicular co-operatives arise out of self-interest, as Gould (1991, p. 12) puts it, 'if the world displays any harmony and order, it arises only as an incidental result of individuals seeking (promoting) their own advantage'.

Dawkins' formulation has aroused a lot of interest in both biological and philosophical circles (Plotkin, 1995) much of which ignores the fact that there are two ways of describing the evolutionary process. For instance, Eldridge (1995) is at pains to point out that Darwin's original description saw that economic success - obtaining food, for instance - biases reproductive success, and that such an effect inevitably biases the transmission of heritable features from one generation to the next. As stated earlier, this is just one of the two ways the evolutionary process can be regarded. The point being made by Dawkins is that it is only those entities which are perpetuated along lineages - that is replicators - which are evolvable, changes directly to vehicles are not genetically propagated.

David Hull (1988a), a philosopher-biologist, has emphasised the active role of the vehicle. He renames the vehicle the interactor: an entity which interacts as a cohesive whole with the environment to produce differential replicator selection. Hull stresses that the interactors are not under the sole control of the replicators and that interactors are causal agents in their own right. Interactors are defined in relation to their effect on selection - if they do not affect selection, then they are not interactors.

Let us consider how interactors can have causal power in the evolutionary process. There are a number of different types of selection mentioned in the literature: for example, artificial, frequency-dependent, kin, natural, niche and sexual. Of these frequency-dependent and kin selection can be regarded as consequences of natural selection, although kin selection can lead to further developments relating to the advantages of kin recognition. Artificial selection was used by Darwin as a model for natural selection. However, artificial selection is more closely related to sexual selection. In artificial selection the selection of a mate is carried out by a human rather than another member of the same species. Sexual

selection is necessary to prevent individuals trying to mate with individuals of a different species. If, for example, different individuals had different mate selection strategies and those strategies were innate, then natural selection would select not only individuals that mated with like species, but those that mated with the fittest members of their species. Sexual selection is a mechanism at the level of vehicles, a mechanism which gives vehicles causal power in the evolutionary process.

A second mechanism which gives a vehicle causal power in the evolutionary process is niche selection. There are a variety of possible niches and vehicles seek out different niches. Again this may be due to innate strategies and can lead to natural selection of the individuals with the best niche selection strategies. However, the result of a vehicle selecting its niche is that it changes the selection pressures operating on it and changes the fitness of the replicators inside of it. Both niche and sexual selection provide additional mechanisms, which with natural selection, I think, provide a more comprehensive picture of how speciation could occur.

The main point which Hull's (1988a) work emphasises - a point made by Dawkins as far back as 1976 (1989b, p. 191): 'Darwinism is too big a theory to be confined to the narrow context of the gene' - is to question a theory of evolution solely based on changes in gene frequencies. This led Dawkins (1989b) to postulate a new type of replicator - a meme. Examples of memes include tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches.

Williams' (1992) theory of two domains, the material and the codical, helps to clarify the nature of a meme. Williams suggests there are two mutually exclusive domains of selection, one which deals with material entities and another which deals with information: the codical domain. The unit of selection in the codical domain he calls a codex: a message coded in a medium - however, the medium is not the message. Thus it does not matter if a message is coded in DNA, in the brain or in a book - they all have the potential to be replicators. The term meme, for example, is used for nongenetic messages

By postulating a second heredity system there is the possibility of a new kind of genetic take-over - that is, a memetic take-over. For example, in the discussion of the causal role of the vehicle in evolution, two mechanisms were given which were described as relying on genetic differences: niche selection and sexual selection. However, these mechanisms

need not rely on genetic differences - the differences could be acquired rather than inherited. Hence, if the memetic system produces a faster rate of adaptive change in a particular case, then the memetic system could 'take over' from the genetic system in that situation.

First, Lamarck (1914) argued that it was the habits - the second nature - that formed the organism not the other way around: that is, behaviour begets form. If, for instance, niche selection is acquired from the parents through imitation, then there need not be any genetic difference between individuals for them to end up following different life styles. Another effect of niche selection can be seen in what is sometimes referred to as habitat tracking (Eldridge, 1995) - that is, where rather than adapting to changing conditions animals do not change, but alter their conditions to find a niche like the original one.

The second mechanism, mate selection, can, in many animals, be acquired rather than inherited. The biologist Hugh Patterson (cf. Eldridge, 1995) postulates that animals have a Specific Mate Recognition System (SMRS) which has many component mechanisms: for example, meeting and recognising a prospective mate, actual mating, successful fertilisation, and the production of viable offspring that will eventually mate and reproduce. Thus changes to the SMRS can cause individuals to no longer recognise one another reproductively.

Todd (1996, p. 389) carried out a set of computational experiments to investigate the effect of learnt SMRS strategies and he found that 'mate preference learning evolves when learned preferences give individuals an advantage in finding appropriate mates'. Todd's simulations showed that parental-imprinting learning was a mechanism which could increase the rate of adaptive change and that imprinters were better at adapting to evolving population structures. Mate preference learning in particular is not only sexually selected, but it also helps promote sexual selection through the choices it promotes. Mate preference learning in some sense allows an individual to say, 'These are the kind of characteristics I want in my kids,' and to choose a mate who will provide the genes more likely to produce those characteristics.

To sum up, Dawkins has postulated two theoretical entities, replicators and vehicles. Replicators are messages coded in different media and replicator survival occurs in what Williams calls the codical domain. Vehicles are survival machines for genes and the sorts of vehicles we are familiar with exist in Williams' material domain or in 'virtual-reality' individuals in artificial life systems. As Hull has emphasised, vehicles can have causal

power in the evolutionary process. The discussion of niche and sexual selection has demonstrated how individuals could have phenotypic differences which are learnt and behaviourally passed on to the next generation without involving the genetic system.

Cairns-Smith argues that one heredity system can give rise to and fall foul of another heredity system, like the genetic system falling foul of Dawkins' memetic system, for example, suicidal cults. Thus Darwin's strong heredity principle - 'like produces like' - need not be grounded solely in molecular biology. In the final part of this section it was considered how such a memetic take-over could be possible, which seriously undermines the view of evolution as just changes in gene frequency. The memetic heredity system is modelled in chapter 5.

3.4 The myopic nature of the Darwinian process

The Darwinian process is often described as myopic because selection works on the basis of immediate benefits and not potential, long-term benefits. Thus the Darwinian process can select things which are good in the short term, but are not the best choice in the long run. However, a number of factors conspire to reduce the effect of this feature of natural selection and lead me to the belief that this limitation is neither as deep nor as problematic as it at first seems.

Kauffman (1995) suggests in his book *At Home in the Universe* that partitioning procedures are well known in many disciplines by a variety of names, from federalism to profit centres, to restructuring, to checks and balances, to political action committees. Kauffman calls this partitioning procedure 'patch logic' and analyses how it can be applied to hard combinatorial problems, like evolutionary search. The basic idea is very simple: take a hard, conflict-laden task in which many parts interact, and divide it into a quilt of non-overlapping patches, then try to optimise within each patch. As this occurs, the coupling between parts in two patches across patch boundaries will change the problem to be solved by the parts in the adjacent patches. The coadaptation of patches relates to coevolution in biology.

Kauffman (1995) shows that patching reduces the risk of premature convergence. The reason is that if a system is in a local maximum and it is split into X patches, then it is unlikely that all X patches are also in their own local maxima. If one of the X patches is not

in a local maximum, then that patch will start adapting and so change the conditions of the adjacent patches. The ontology described in chapter 2, where designs fitted in niches formed by other designs and so on, implicitly incorporates the notion of patches - designs form patches.

While the patches procedure demonstrates a means of reducing premature convergence, there are two types of convergence (Harvey & Thompson, 1996): genetic convergence - already mentioned in relation to gene fixation (see section 3.3.2), and fitness convergence to an asymptote. Harvey and Thompson (1996) have demonstrated that these two concepts are not the same, for example, in one experiment they found that while genetic convergence occurred within 10 generations, fitness continued to rise for 3000 generations. They have also shown that genetic convergence occurs through the action of genetic drift in the absence of selection. But, more importantly to this section, Harvey and Thompson's work demonstrated how 'neutral networks' can affect myopic selection. An example of neutral networks discovered by biologists (cf. Huynen *et al*, 1996) is that it is possible to transverse RNA space through selectively neutral mutations: that is, the selectively neutral mutations prevent populations getting stuck in local optima.

A neutral network of a fitness landscape is defined by Harvey and Thompson (1996) as 'a set of connecting points of equivalent fitness, each representing a similar genotype; here connected means that there exists a path of single (neutral) mutations which can traverse the network between any two points on it without affecting fitness'. The function of neutral networks is to provide avenues into other areas of genotype space where potentially fruitful mutations are possible: that is, neutral networks act like 'worm holes' in phenotype space.

Harvey (1997) gave a simple example of a neutral network: take a genotype, double it and add a switch so that only the first or the second half is interpreted. In this case, mutations can occur to the uninterpreted half of the genotype which allow subsequent mutations to move into areas of genotype space with potentially fitter phenotypes. This example bears similarities to diploid genotypes: that is, genotypes like our own which have two sets of genes and a dominance mechanism.

Holland (1994) investigated dominance and concluded that a given minimal rate of occurrence of alleles can be maintained with a mutation rate which is the square of the rate required in the absence of dominance, and with the dominance-change operator the

combination of alleles defining a schema can be reserved or released in a single operation. Holland's work describes, what we called previously, neutral mutations. Thus the so-called 'junk DNA' has ascribed to it two roles: 1) functions as schema memory; 2) allows neutral mutations - mutations to uninterpreted sections of the genotype which can be interpreted in latter generations.

The effect of neutral networks can be summarised as follows: redundancy can create neutral networks which can transform a fitness landscape. Harvey and Thompson (1996) give an example where a fitness landscape of any ruggedness can be transformed into a different, but related, landscape with a single global optimum (and thus there are no local optima in which to become trapped). Unfortunately, Harvey and Thompson (1996) think that this beneficial effect will not be true of all landscapes: for instance landscapes with a single global optimum only reachable by random search.

As well as patch logic and neutral networks, a number of other factors can have an impact on the myopic nature of selection. For instance, the evolutionary process occurs in space. Space has the function of separating agents and preventing every agent competing with every other agent. It takes time for changes to propagate through space which gives other agents a chance to try out other, possibly initially poorer, avenues. Thus space functions to restrict the interaction between agents.

Another factor, pointed out to me by Aaron Sloman, which can have a similar effect to spatial distribution is that an agent only interacts with a limited number of other agents: selective interaction. An obvious example is seen in sexually reproducing species where individuals can only mate with the opposite sex, which means, in most cases, with only half of the population. In science, while modern communications media essentially remove the effect of spatial distribution, partitioning still occurs because different research goes on in different disciplines and communication between the disciplines is less, on average, than communication within the disciplines. Thus selective interaction can be due to physical constraints - how many interactions are physically possible - or behavioural ones - that is, who agents 'choose' to interact with.

An important case of selective interaction in evolutionary theory is constrained reproductive interaction: that is, when an individual is constrained as to how frequently it can reproduce, such as females with a limited supply of eggs. Constrained reproduction

would limit the maximum number of offspring an individual could produce and thus the genetic influence an individual could have on the next population.

To conclude, Kauffman's work on hard combinatorial mathematical problems has shown how breaking up hard combinatorial problems into a series of interrelated sub-problems - patches - can reduce the risk of becoming caught in local optima. Certain features of the natural world provide the necessary conditions for patches to occur. Furthermore, certain conditions - for example, space, the physical landscape and interaction constraints - may slow down the rate of evolution in the short term by preventing new forms immediately taking precedent in a population which allows other trajectories in design space to be explored. However, these constraints could prevent a good trajectory in design space being followed, for example, if selective interaction prevented individuals mating and producing what could go on to become a fit lineage. Finally, work on neutral networks (see page 54) has demonstrated how fitness landscapes can be transformed by selectively neutral mutations providing avenues of escape from local optima (Harvey, 1997).

3.5 Chapter Summary

The aim in this chapter has been to give an abstract description of the current state of evolutionary thinking. The first section focused on Darwin's theory of natural selection. Darwin proposed his ideas in the form of a formal argument which can be readily transformed into an algorithmic description. The algorithm is of the following form: if reproduction is biased by the quality of a structure, then greater numbers of fitter structures are propagated.

In the second half of this chapter the nature of the heredity mechanisms was explored. These mechanisms are responsible for both heredity and variation. The genetic code is shown to be an arbitrary, digital code. The fact the code is arbitrary allows the results derived from work on arbitrary codes to be applied to genetics. Differential reproduction is shown to be a sufficient mechanism for adaptive change, but with some serious constraints: firstly, differential reproduction is constrained to the initial variation; secondly, differential reproduction alone is an exponential time algorithm - that is, impractical for large problems. The genetic operators of mutation and recombination are shown to augment differential reproduction by adding variation and manipulating schema.

While the fact that evolution involves cumulative change is obvious, one possible misunderstanding needs to be ruled out. Darwinian evolution is not a process of *random* trial and error, but of *constrained* or *selective* trial and error (Simon, 1982): that is, evolution is not a completely random process, there is a selectivity to the trails generated. Brute-force search is replaced with a combined system of search and ‘reason’ which can eliminate a substantial amount of trial and error from the search process.

Simon (1982) identifies two basic kinds of selectivity. One form of selectivity is that as various paths are tried out, the consequences of following them are noted, and this information is used to guide further search. The second source of selectivity, identified by Simon, is previous experience of a similar problem, for instance, when the problem to be solved is analogous to the ones that have already been solved. In cases where the future is like the past, by selecting the paths which led to previous solutions or their analogues, trial-and-error search can be greatly reduced or altogether eliminated. In the next chapter the thesis that development may be one way in which evolution exploits the second type of selectivity - analogous problem solving - is examined.

To conclude, this description reflects the core of current evolutionary thinking. All the mechanisms have been abstracted and shown to be logically sufficient for the task at hand: producing adaptive change. However, there are ways in which the theory can be expanded. Firstly, Darwin described the principle of heredity as ‘like begets like’. With the advent of the neo-Darwinian synthesis, heredity mechanisms have become synonymous with genetics and evolution is often characterised as changes in gene frequency. While researchers would agree that genetics is not the only heredity mechanism, the hold of molecular biology is so strong that it leads to language like: ‘all adaptations are the preservation of DNA’ (Hull, 1988a, p. 31), or ‘the origin of evolutionary novelty is a process of gene mutation’ (Maynard Smith, 1977, p. 180) or ‘evolution requires genetic change, mutation’ (Dawkins, 1989b, p. 262). To be fair, many of these comments are probably based on over-specific terminology, and there are many researchers who have considered other kinds of heredity mechanisms (e.g. Bonner, 1980; Plotkin and Odling-Smee, 1981; Dawkins, 1989b; Moravec 1989).

The main points in this chapter are:

1. The basic Darwinian process is a trial-and-error mechanism which involves variation and selection. While Darwin focused on the role of selection, the role of constrained variation is fundamental to the understanding of evolution.
2. Darwin developed his theory by a formal argument which can be readily transformed into an algorithmic description. The Darwinian algorithm is a parallel, probabilistic, anytime algorithm.
3. The Lewontin-Campbell formulation of the Darwinian process aims to show the sufficient and necessary conditions for adaptive change to occur. For example, according to Lewontin, phenotypic variation, differential fitness, heritable fitness are the sufficient and necessary conditions for adaptive change to occur (Dunbar, 1982).
4. The genetic mechanisms form the base of our current understanding of heredity and variation. The genetic code is an arbitrary, digital code and the genetic operators, recombination and mutation, combine to form a powerful information processing mechanism which searches for fit combinations of genes.
5. The formal analysis of different genetic operators has shown that while Darwin emphasised the role of selection, there is a role to be played by constrained variation: that is, variant generators which have been selected because they explore certain areas of design space.
6. Abstract formulation of the entities involved in the evolutionary process has led to a second heredity mechanism being described: the memetic system. Cairns-Smith's hypothesis about the origin of life suggests the possibility of take-over: that is, one heredity system taking over another one. Furthermore, there is a potential for conflict between the genetic and memetic heredity systems.
7. Evidence has been given to show how individual organisms can have a causal role in the evolutionary process. This factor combined with the last point gives rise to the possibility of adaptive change occurring without necessarily requiring changes in gene frequencies.
8. While selection is myopic, there are a number of features - patches, spatial distribution, constrained interactions and selective mating - which *reduce* the risk of getting stuck in local optima.

In the next chapter, the role of development in exploiting Simon's second type of selectivity is discussed, and the claims made by a number of researchers (e.g. Goodwin, 1994; Kauffman, 1993, 1995, 1997) that self-organisation plays a big role in evolution are examined. A set of computational experiments are described in the second half of chapter 4 which investigated, among other things, the selection of evolvability: that is, the selection of heredity mechanisms which produce the fittest variants.

4. The evolution of evolvability

4.1 Introduction

Central to the structure of this chapter is the so-called ‘fundamental principle of embryology’ described by the German embryologist August Weismann. In this principle, Weismann distinguished between the development from the genotype to the phenotype and reproduction between genotypes (see Figure 4-1).

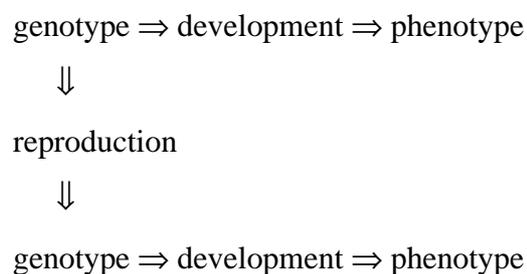


Figure 4-1: Weismann’s ‘fundamental principle of embryology’.

This scheme shows the separation of development from reproduction and the lack of a causal arrow from phenotype to genotype illustrates that the genetic transmission of acquired characters is prohibited. It is now agreed that, with a few exceptions, information only goes from DNA to RNA and then to protein and not in the other direction.

Weismann’s scheme clearly indicates the two roles of the genotype: in development and in reproduction (Dawkins, 1989a). The role of the genotype in reproduction was examined in the last chapter. In this chapter the following questions are asked: What role does the genotype play in development? How does the genotype map onto the phenotype? How is the genotype interpreted? Furthermore, Dawkins (1994, p. 129) postulated that ‘certain kinds of embryology are good not only for individual survival, but also for giving rise to further evolution’. Dawkins (1989a) called this property ‘evolvability’. In this chapter development is discussed in relation to the property of evolvability and the hypothesis that evolvability is selectable is tested using computational experiments.

As already stated this chapter is split into two main sections: development and reproduction. In the first section, the role of development in evolution is discussed. The aim

in this section is to see how certain properties of developmental programs can contribute to the evolvability of a lineage. For example, the effects of: developing from a single cell (see section 4.2); constrained embryologies (see section 4.2.2); and abstract design principles like modularity and hierarchical organisation (see sections 4.2.3 & 4.2.4). This section shows how developmental programs take advantage of Simon's second form of selectivity: reuse of solutions in similar problems. Furthermore, the role of self-organisation in evolution is examined in section 4.2.1

In the second half of this chapter the results of a set of computation experiments are described. These experiments demonstrate two things. First, that evolvability can be selected in evolution. Second, that evolvability is a context-dependent property: that is, the performance of a mechanism is partly dependant on the context it finds itself in.

The novel contribution to research in this chapter is to provide an analysis of some of the ways development can contribute to evolvability and, using computational experiments, examine in what conditions evolvability-enhancing mechanisms can be selected.

4.2 How developmental programs can contribute to evolvability

Developmental biology or embryology is concerned with the processes of change which transform a single cell through embryonic stages to maturity and beyond. Dawkins (1995) suggests calling this entire process 'total embryology'. In this section the aim is to explain how the genotype is believed to map onto the phenotype, not to provide an exhaustive literature survey on embryology. The description of development will then be used to show how different embryologies could affect both what can evolve and how fast it can evolve.

To start with, the genotype is often mistakenly regarded as a kind of blue print (Dawkins, 1988; Cohen and Stewart, 1995). A better metaphor is of the genotype as specifying a recipe (Dawkins, 1988; Cohen and Stewart, 1995). In the recipe metaphor, the genotype is regarded as coding for a construction process for a phenotype. This is the view of development as a program.

The genotype (program) requires an interpreter - that is, the DNA's derives its 'meaning' by the process of being interpreted (Cohen and Stewart, 1995). In most cases, a copy of the mother's interpreter (her mRNA) is passed on in the egg. The maternal

interpreter has the same function as the start up disk for a computer (Bonner, 1995). Thus both an egg and a genotype are necessary for development to start.

The term 'gene', as already mentioned, has two senses: the gene coding for a phenotypic character and the gene coding for a protein. In embryology, the term 'gene' is used in the second sense. However, the view that one gene codes for one protein is complicated by the fact that genes have been found to be coded within other genes and different stretches of DNA have been found which are interpreted in different ways at different times during the life cycle.

Let us return to the one-gene-one-protein picture. Some genes code for structural proteins which are incorporated into different structures within the cell. Other genes code for enzymes which catalyse specific chemical reactions, functioning like condition-action rules: if chemical A and chemical B are present, then make chemical C. Yet other genes administer the actions of other genes. These genes, called 'homeotic genes', switch on or off other genes. It has been estimated that in some plants and animals, five percent or less of the DNA specifies how to make proteins, while much of the rest is involved in control sequences which organise procedures (Cohen and Stewart, 1995).

One feature of the developmental program which is very important is what computer scientists call 'conditional branching'. This refers to a program which has control structures which test for certain conditions at different points as the program is executed. Consider the following example which illustrates conditional branching. The journey from my house in London to Birmingham can be specified in a number of ways. For example, I could say go 40 miles around the M25 and then go 100 miles up the M40. Alternatively I could say take the M25 until you get to the M40 junction (condition 1) and follow the M40 until you get to Birmingham (condition 2). In the first route, changes to the lengths of either road would cause the plan to fail; in the second route, the plan is robust to such changes. Many features in development need to be specified like the second route to prevent problems such as having bones which are too big for the muscles or there not being enough skin to cover the surface of a limb.

Two types of conditions can be discriminated in developmental programs: internal and external. First, the developmental program is self-referential (Cohen and Stewart, 1995). Such self-referential features form internal conditions. Second, the developmental program

is affected by environmental conditions - for example, temperature or chemical concentrations. The environmental factors thus form external conditions.

As already mentioned, there are certain complications with the notion of genes, more specifically, with the notion of the genotype-phenotype mapping. There is no one-to-one mapping between genes and phenotypic features. The same stretches of the genotype can code for different phenotypic features producing a form of information compression. This leads to a connection between features called pleiotropy. Pleiotropy can constrain what can evolve because gene changes which benefit one feature can be detrimental to the performance of another feature. However, because 'greater amounts of genetic material are more susceptible to the randomising effects of mutation' (Johnston, 1982, p.339), there is an advantage to shorter and more compressed genotypes. A selection pressure for shorter genotypes can produce an increased amount of pleiotropy. Further developmental constraints produced by constrained embryologies are discussed in section 4.2.2.

Now we have all the concepts needed to explain an example of a current theory of development and to illustrate differential evolvability of embryologies. However, before we do this let us briefly discuss the phenomenon of genetic assimilation (Bonner, 1995; Scull, 1996), (also called the Baldwin effect in relation to behaviour (Baldwin, 1896), or canalisation in relation to development (Waddington, 1942)).

Genetic assimilation is when genes infiltrate or become assimilated to reinforce something that is already occurring (Bonner, 1995). When, for example, nongenetically determined phenotypic characters become genetically determined by mutations which do not disturb these phenotypic characters. Mutations which do disturb these features are selected against. A hypothetical example would be if an organism grows larger if it develops in a hot environment and this additional size, relative to other organisms, provides a selective advantage. Genes which cause the same effect (increased size) as developing in a hot environment will have no effect on the beneficial phenotypic character, genes which disrupt the temperature effect remove the beneficial phenotypic character. Thus the genes which mimic the temperature effect will be genetically assimilated: that is, phenotypic characters which were originally produced by external conditions are, due to the action of the process of genetic assimilation, now produced by internal conditions.

The process of genetic assimilation can guide evolution. For example, conditionally branching developmental programs can, by organisms seeking out different environments, lead to different phenotypes. While the different phenotypes produced in different environments are from the same genetic stock the process of genetic assimilation can incorporate genes which produce the same phenotypic effects. The important point is that mutations - genetic change - comes second in this scheme. In chapter 5 I shall discuss how learning can guide evolution.

Let us return to developmental programs. There are three basic, constructive processes that take place as an egg unfolds into an adult: growth, morphogenetic movements and cellular differentiation (Bonner, 1995). But how can all the phenotypically different kinds of cells in an organism have the same genotype? As the previous discussion of conditional branching and genetic assimilation have shown the expression of the developmental program is context-dependent. First, a different subprogram may be expressed because of conditional branching leading to different subprograms being switched on; second, differences in the initial conditions can cause the same subprogram to produce different results.

Early research in embryology distinguished two types of developmental programs: mosaic development - the fate of all the cells is spelled out in a set of initial rules; and regulative development - which we have been discussing. Later research has shown this distinction to be oversimplified with development often showing both mosaic and regulatory phases. The question is: How is the developmental process orchestrated, how, for example, does a cell 'know' what it is meant to turn into?

The developmental biologist Lewis Wolpert (1969, 1995) postulated a theory of positional information to explain how development occurs. Wolpert suggests that implicit within the developing embryo are different conditions which affect the part of the DNA program which is interpreted and thus affect the ultimate form of both the cell and the organism. Initially an embryo begins as a single cell which has certain chemical gradients within it. These chemical gradients have been attributed, in some cases, to a 'maternal effect': that is, the distribution of cytoplasmic molecules is influenced by maternal genes. When the cell divides the result of the chemical gradients within a cell is that each of the daughter cells will have a slightly different chemical makeup. The different chemical

concentrations in the cells cause different genetic subprograms to be activated and produce, among other things, cellular differentiation.

However, there are numerous other factors involved - for example, as well as chemical gradients, electrical gradients have been found which can also provide information about the position of a cell (Bonner, 1995). Thus developing cells have other sources of positional information.

Genes can also produce secondary effects of enormous consequence by producing molecules, called cell adhesion molecules (CAM), that lie on the outside of the cell membrane and affect the adhesiveness of the cell. CAMs produce differential adhesion between different cell groups. As the cells divide, the whole cluster goes through carefully choreographed changes with certain regions developing at different rates and affecting the overall form.

The research on homeotic genes (see page 62) has led to the discovery of 'gene nets' (Bonner, 1995). This refers to the system of isolating genes into units that can behave relatively independently of one another. Dramatic examples of this are achieved by geneticists changing homeotic gene sequences: for example, the *antennapedia* mutation replaces an antenna with a leg, or the *cockeyed* mutation that replaces eyes with genital structures (Cohen, 1995).

The effect of gene nets is to produce a modular structure. Mutations to gene nets can affect one design independently of other designs. This allows the parallel evolution of different subsystems. Furthermore, these modules (designs) form the niches of the other modules (designs), thus producing patches (see section 3.3.3). Patching, as you will remember, has the effect of reducing the risk of structures getting caught in local maxima.

Even this very simplified description of the developmental process shows the enormous complexity involved. It is clear that a genotype plus an interpreter and a great deal of contextual information is necessary to produce a phenotype. The points to remember are: 1) the developmental process can be described as an algorithm - a parallel processing, conditionally branching program; 2) some developmental programs have a modular and hierarchical structure. Before examining the consequences of these points, two questions will be considered: First, why separate growth from reproduction? Second, is it possible to inspect the genotype to determine the phenotype's properties?

Some organisms can reproduce by the same process as they grow, literally bits of the organism break off and grow into separate organisms. However, many organisms have separate processes of development and reproduction. Dawkins (1989b) postulates three selective advantages - which he calls, 'back to the drawing board', 'orderly life-cycle' and 'cellular uniformity' - for separating the two processes.

First, Dawkins (1989b) argues that there are constraints to the process of direct transformation of one design into another, for example, while it may be possible to beat a sword into a ploughshare it would be hard to imagine how to beat a propeller engine into a jet engine. If the majority of changes bring about bad results, then the transformation suffers a problem, equivalent to Muller's ratchet (see section 3.3.2), where there is an accumulation of bad results. What is required is for the designer, informed by the old ideas, to go back to the drawing board and redesign the product. Thus while a change to a design product may require many carefully co-ordinated changes, a change to a design process may bring about all those changes in one go while leaving the original product untouched.

Secondly, Dawkins (1989b, p. 262) argues that, 'well-tempered regulations of gene activity are a prerequisite for embryologies capable of crafting complex organs and tissues'. The growth cycle provides a clock by means of which embryological events can be triggered. Development has been described as a conditionally branching process and a growth cycle is another way of providing the necessary conditional information.

The third reason Dawkins (1989b) gives for separating development from reproduction is a genetic one. If an organism reproduces by the same process it grows, then a mutation can arise in any cell in any part of the organism. However, if an organism produces, for example, spores, then a mutation to a spore will, through development, come to affect all the cells in the new organism. Thus, the second case has a strict genetic identity which the selective process can act on.

To sum up, Dawkins gives three selective advantages of separating development from growth. Current work in evolutionary computation has largely ignored development (Ackley and Littman, 1994). However, while natural selection acts on phenotypes, what is selected is the genotypes: the construction processes for phenotypes. In the second half of this chapter a consequence of this process-product distinction will be examined, that is, where selection acting on the phenotype can also select between different generators of variation.

The second question to be discussed was: Is it possible to inspect the genotype to determine the phenotype's properties? Developmental biologists think not (e.g. Wolpert, 1995). Wolpert points out that without an egg to provide the correct initial conditions and the initial interpreter, a genotype is useless. Moreover, the process of interpreting the genotype is conditional on the context. Thus both the program (genotype and egg) and the input (initial conditions of egg and environment) need to be known if the phenotype is to be computed. Without knowing the exact initial environmental conditions there is the possibility of chaos effects: that is, very small differences in initial conditions could lead to very big differences in phenotypes. Even if all this information were known - about the genotype, the interpreter and the environmental conditions - then the computation required to examine the phenotype's properties would be immense (intractable in practise).

An additional complication to this computation is that to assess the fitness of future phenotype's requires knowledge of the environment that that phenotype is to be found in. The longer the developmental process takes, the greater the number of environmental possibilities that have to be taken into account. Thus, as discussed in chapter 2, the basis of comparison which is necessary to assess the fitness of a design can change, and these changes would have to be predicted if the fitness of future phenotypes was to be assessed correctly.

While this intuitive argument suggests that the computation is not possible in practise, some researchers (e.g. Langton, 1989) believe that there are theoretical reasons preventing this calculation. Turing's famous work on the halting problem proved that a machine could not be constructed which, given any other machine and its start state, could determine in a finite time whether or not the machine would reach its halt state. Turing's work has been extended, by Rice and others (cf. Hopcroft & Ullman, 1979), to show that determination of any non-trivial property (a property of belonging to nothing or to everything) of an arbitrary machine is not possible.

However, Turing's work demonstrated that a single machine could not be constructed that could determine, in a finite time, whether any other machine would terminate. In other words, certain machines could determine whether a subset of possible machines could terminate. Thus while it may not be possible to construct a machine that can determine all of

a genotype's phenotypic properties, machines could be constructed that could determine certain properties.

To sum up, some developmental programs appear to be written in a procedural-like language in the form of a conditionally branching and modularly organised, hierarchical control system. The contribution of these features to the evolvability of a lineage are discussed in sections 4.2.3 & 4.2.4. A number of advantages have been discussed for having separate processes of development and reproduction, for example, 'back to the drawing board', 'orderly life-cycle' and 'cellular uniformity'.

Finally, it appears that when generating new designs it may not be possible - in practice, maybe in principle - to determine their properties. This implies a need for a trial and error process, guided by feedback, to evaluate the variants. The Darwinian algorithm is such a process. In fact, Langton (1989) argues that it is quite likely that the Darwinian process is the *only* efficient, *general* procedure for finding genotypes with specific phenotypic traits.

4.2.1 Origins of order

Proteins are a major class of biological macromolecules whose wide range of structural and functional roles in cells is reflected in the diversity of their chemical structures and three-dimensional shapes (Rees and Sternberg, 1987). The organisation of a protein is split into four levels: primary, secondary, tertiary and quaternary. The primary structure refers to the chemical formula of the protein and describes the linear sequence of amino acids. The other levels relate to how the linear sequence is organised in space. In the previous section it was described how the genotype codes for a protein. Specifically, the DNA only codes for the primary structure of a protein. The spatial organisation of the structure arises spontaneously as the structure seeks a lower energy state. However, in some cases other proteins can influence certain fold choices (Cohen and Stewart, 1995). The central point is that the genotype codes only for the primary organisation, the three-dimensional shape arises out of self-organisation - 'order for free'.

It has been known for many years that physical systems exhibit spontaneous order: an oil droplet in water forms a sphere and snowflakes exhibit a six fold symmetry. A number of scientists argue that self-organisation has a central role to play in development and evolution

(e.g. Kauffman, 1993, 1995, 1997; Goodwin, 1994). What is being recognised is that there are certain characteristic types of order which emerge from the interactions of many different components (Goodwin, 1994). Mathematical models of dynamical systems have led to a formal understanding of this phenomenon. For example, numerical computations on simple non-linear systems revealed unsuspected invariants that predicted, for broad classes of systems, at what point they would change from orderly to chaotic (Simon, 1996). The understanding which has been gained in many aspects of chaos, however, does not imply we can predict the behaviour of such systems.

Let us consider some of the work of Kauffman (1995) who has been investigating the phenomenon of self-organisation for over 30 years. Using computational experiments he has begun to understand the conditions in which order arises. In these experiments he used Boolean networks. These networks can be characterised by two variables: N , the number of nodes and K , the number of inputs per node. A network is generated by randomly assigning to each node K inputs and, again at random, a Boolean function. Kauffman (1997) studied the generic properties of classes of networks, or ensembles, by selecting random samples and analysing many such samples to provide an understanding of the typical behaviour of members of each ensemble.

Kauffman has found that two features of the way networks are constructed can control whether they are in an ordered regime, a chaotic regime, or a phase transition regime between these. One feature is the number of inputs a node has. Sparsely connected networks, with $K = 1$ or $K = 2$, exhibit order; highly connected networks exhibit chaos. A second feature that controls the emergence of order or chaos is bias in the control rules. For example, AND and OR rules tend to create order.

Kauffman's results can be best understood by considering a concrete example. When $K = 1$, randomly started networks quickly fall into very short state cycles, so short they often consist of one state. Such networks rapidly become frozen into a single state.

At the other end of the scale when $K = N$, the expected median state cycle length was the square root of the number of states. Kauffman found that these networks do exhibit order. However, as $K = N$ networks have state cycles whose expected lengths scales exponentially with the size of the system, in large systems these state cycles are not observable and thus state changes appear random. For example, a network with 200

variables would have a state cycle of length 2^{100} (approximately 10^{30}) which at a rate of one state cycle per microsecond would require some billions of times the history of the universe since the big bang 14 billion years ago to orbit the state cycle.

Kauffman found that the number of attractors in a $K = N$ network is N/e , where e is the basis of the natural logarithm. Thus in a network where $N = 100000$ there are about 37000 attractors. Kauffman has also been able to show that $K = N$ networks show the maximum possible sensitivity to initial conditions.

Chaos in systems with a few continuous differential variables is defined as when the systems fall onto 'strange attractors' where local flow is divergent but remains on the attractor (Kauffman, 1997). Kauffman calls this low dimensional chaos to distinguish it from high dimensional chaos: systems with a large number of variables in which the length of the cycle scale exponentially with the number of variables, and which show sensitivity to the initial conditions. Even though both behaviours are well established, the relation between low dimensional chaos in continuous systems and the high dimensional chaos described here is not clear (Kauffman, 1997).

Order arises when $K = 2$. The expected length of the state cycles in these networks is in the order of the square root of the number of nodes. Therefore, when $N = 100000$ there are about 317 states per cycle. To reiterate this point, out of a possible 2^{100000} or 10^{30000} states the network visits only about 317 of them. In this class of system, their dynamics cause them to move to a very small region of the state space and stay there (Kauffman, 1995). Also nearby states converge in state space. In other words, two similar initial patterns will likely lie in the same basin of attraction, hence driving the system to the same attractor - they are not chaotic in the sense described earlier (Kauffman, 1995). And if a system undergoes a mutation - a change to its wiring or logic - it does not become chaotic, mutations cause graceful alterations in the system's behaviour. Because small changes lead to small changes to the basins and attractors, these systems are more readily tuneable or evolvable. Finally, these systems are not too orderly, they are not frozen like the $N = 1$ networks, but are capable of complex behaviour.

To conclude, Kauffman's experiments demonstrate some of the conditions out of which order can arise: in Boolean networks order is dependent on the number of inputs and the Boolean functions used. Even in randomly generated, sparsely connected systems using a

subset of the Boolean functions there is immense order. But how does all this relate to this thesis? Kauffman's original work was on gene networks and how development arises out of them. As his results demonstrate, there is inherent order in such systems and they are relatively robust to small mutations and which lead to graceful, potentially selectable, changes in system behaviour.

The abstract mathematical properties discussed in this section are inherent in systems made up of interacting nodes regardless of their material construction. For example, Kauffman discovered that neural networks are fundamentally the same kind of system as gene networks and thus will have many of the same properties (Waldrop, 1994).

The possible roles of self-organisation in evolution need to be better understood. For example, selection does not have to sort through all logically possible states, just the causally possible ones that arise in that type of system. And even chaotic systems can be tamed by substituting control for prediction:

'Few of the adaptive systems that have been forged by evolution or shaped by man depend on prediction as their main means for coping with the future. Two complementary mechanisms for dealing with changes in the external environment are often more effective than prediction: homeostatic mechanisms which make the system relatively insensitive to the environment and retrospective feedback adjustment to the environment's variation.' (Simon, 1982, p. 149)

The research into self-organisation has also demonstrated that chemical cycles are highly likely to arise spontaneously. Cairns-Smith's hypothesis of the origin of life required chemical cycles to form naturally (see section 3.3.3). Furthermore, the evolution of life is often assumed to be highly improbable - the question is, how improbable? As we discover more of the mechanisms involved in evolution, it appears that with each new find the probability of the evolution of life seems a little more likely.

4.2.2 Kaleidoscopic embryologies

Richard Dawkins (1989a, 1996), using techniques now referred to as artificial life (Langton, 1989), has explored how different embryologies can lead to a property he calls

‘evolvability’. Dawkins (1989b, p. 219) wrote, ‘Perhaps there is a sense in which a form of natural selection favours, not just adaptively successful phenotypes, but a tendency to evolve in certain directions, or even just a tendency to evolve at all.’ For example, if the members of one lineage adapt faster than those of another, then the fittest individuals and their associated mechanisms will be naturally selected.

In the second half of this chapter, I document some of my own computational experiments which, among other things, demonstrate that evolvability can be selected for by evolution. The aim of this section, however, is to examine Dawkins’ computational experiments in embryology and relate them to the previous description of development.

Dawkins engineered two initial conditions into his world: i) the entities must be replicators (see section 3.3.3); ii) for selection to act there must be an embryology which the genes differentially influence. Dawkins wrote a computer program around two main procedures: develop and reproduce. Selection was left to the aesthetic taste of a human chooser.

Two kinds of embryology can be distinguished: unconstrained and constrained (Dawkins, 1989a). An unconstrained embryology has maximum generality, for example, a picture on a computer screen with one gene for each pixel. This approach, however, has a number of problems. Not only are huge areas of the space of possible phenotypes uninteresting, but the time it would take to navigate in such a space is impractical.

Constrained embryologies, on the other hand, specify smaller numbers of phenotypes and have a smaller set of genes. In Dawkins’ program these genes control drawing operators. For example, genes could code for families of lines: straight lines with genes controlling lengths and angles and curves whose polynomial coefficients are specified by genes. However, Dawkins chose to use recursive procedures - procedures which as part of their execution call themselves.

Dawkins’ basic embryology consisted of a tree function which received as input a length - a depth of branching (specified by one gene) - and an array of values (specified by eight genes). In the original program all the embryologies had bilateral symmetry.

What the experiments with constrained embryologies demonstrated was that different embryologies affected the accessibility relationships between phenotypes. That is, the embryology defines what phenotypes are possible and the relation between phenotypes. For

example, if the distance between phenotypes was measured in mutations, then one embryology may cause two different phenotypes to be a single mutation away from each other, whereas another embryology may place the phenotypes many mutations away from each other.

Dawkins (1989) further developed his program by adding more symmetries to the embryologies and incorporating genes to control which symmetries were switched on or off. The additional symmetries changed both what phenotypes were possible and the distance between phenotypes measured in mutations. But by their very nature constrained embryologies can only produce a subset of possible phenotypes - for example, in this case only those with even numbers of symmetries.

Dawkins (1996) also added the property of segmentation to his embryologies. This allowed a new subset of phenotypes to be explored. Genes were added which controlled segment gradients allowing individual segments to vary in size. In Dawkins' (1996) second program Arthromorphs the fixed repertoire of genes was replaced by a genetic system which incorporated the genetic operators deletion and duplication. These operators allowed the length of the genotype to vary resulting in an open-ended search space. A grammar was developed in which each feature of the arthromorph's body is influenced by three genes. There is a gene specific to the segment concerned, a gene that applies to the whole animal, and a gene that applies to a sub-sequence of segments - called 'tagma', such as the thorax. The effect is quite dramatic, mutations can affect a single segment or be propagated down a number of segments or even the whole body.

Dawkins' computational experiments illustrate a number of the properties discussed in the section on development - for example, recursion and conditional branching. Constrained embryologies can limit the search for phenotypes to specific areas of the phenotype space and change the accessibility relationship between phenotypes. Thus constrained embryologies can increase the evolvability of an entity by reducing the number of trials possible at each generation and reducing the distance in genotype space between genotypes. However, there are inherent costs incurred by using constrained embryologies: developmental constraints. Finally, the property of segmentation in the arthromorph program demonstrated how subcomponents can be reused. Reuse of subcomponents is discussed in the next two sections.

4.2.3 The architecture of complexity

There have been a number of suggestions for linking hierarchies with evolution (e.g. Lewontin, 1970; Plotkin and Odling-Smee, 1981; Salthe, 1993; Eldridge 1995; Gould, 1997). Hierarchy theory was mentioned in chapter 2 (see section 2.4.1) with reference to Herbert Simon's (1982) article, 'The Architecture of Complexity'. Simon set out to show that hierarchic structures - of the 'box within box' form - have certain properties which allow them to evolve faster than non-hierarchic structures. In this section Simon's argument is considered and related to development as described in the previous sections and the Schema Theorem described in the previous chapter (see section 3.3.2). The aim in this section is to see how development could take advantage of 'stable intermediary forms'.

Simon explains the concept of stable intermediary forms using the parable of two watchmakers, Hora and Tempus. Both watchmakers make watches consisting of 1000 parts. Tempus constructs his watches so that if he puts down a partly assembled watch (which can be up to 999 units) to, for example, answer the phone it falls to pieces. Hora, on the other hand, puts his watches together in ten element subunits, which are further combined until the watch is completed. If Hora's phone rings, then he loses a maximum of nine units.

Simon analyses this scenario mathematically: if the probability of a phone call is p , then the probability of Tempus finishing a watch is $(1 - p)^{1000}$ with an average loss of time with each interruption of $1/p$ parts. Hora has to complete 111 assemblies per watch. The probability of him being interrupted per unit is $(1 - p)^{10}$ with an average loss of five units. If p is about 0.01, then it takes Tempus on average approximately 4000 times longer to assemble a watch than Hora. Thus Hora's watches built with the aid of subassemblies (stable intermediary forms) pays back 4000 fold relative to Tempus's watches which are constructed without subassemblies.

Simon (1996) defined the span of a system as the number of subsystems into which a system can be partitioned (see page 17). The mathematical analysis of the Hora-Tempus parable has been extended so that, 'if there exists a hierarchy of potential "subassemblies," with about the same span s at each level of the hierarchy, then the time required for a subassembly can be expected to be about the same at each level - that is, proportional to

$1/(1 - p)^s$. The time required for the assembly of n elements will be proportional to $\log_s n$, that is, the number of levels in the system' (Simon, 1996, p. 190).

As Simon is well aware, this scenario incorporates many simplifications. However, he lays to rest four possible misunderstandings. First, the theory specifies no teleological mechanism. Complex forms can arise from simple ones by purely random processes; while direction is provided to the process by the stability of the complex forms once they come into existence.

Second, Simon points out that not all complex systems are hierarchical - for example, polymers. However, Simon suggests regarding these as special cases - hierarchies with a span of one.

Third, Simon points out that the evolutionary process does not violate the second law of thermodynamics:

'The evolution of complex systems from simple elements implies nothing, one way or the other, about the change in entropy of the entire system. If the process absorbs free energy, the complex system will have a smaller entropy than its components; if it releases free energy, the opposite will be true... For the evolutionary process we are describing, the equilibria of the intermediate states need have only local and not global stability, and they may be stable only in the intermediate stage - that is, as long as there is a source of free energy to draw upon.' (Simon, 1982, p. 191)

Finally, because organisms are not energetically closed systems, there is no way to calculate the direction or rate of evolution purely from thermodynamic considerations. For example, the amount of entropy involved in the formation of a single cell is trivially small and the 'improbability' of evolution has nothing to do with this quantity (Simon, 1996).

The main point being made in this section is that stable intermediary forms can exert a powerful effect on the evolutionary process. Simon's (1996, p. 193) claim is that 'the potential for rapid evolution exists in any complex system that consists of a set of stable subsystems, each operating nearly independently of the detailed processes going on within the other subsystems, hence influenced mainly by the net inputs and outputs of the other

systems.’ This argument also implies that many complex systems are hierarchical because they are the ones which had time to evolve. Thus, a modular-hierarchical organisation is a property which can potentially increase the evolvability of a structure.

4.2.4 Genetic programming

Research in evolutionary computing substantiates Simon’s claims that subassemblies can affect the rate of evolution. In this section Koza’s (1992, 1994, 1995) work on genetic programming provides a concrete illustration of the power of stable intermediary forms, as well as demonstrating the advantages of non-fixed length genotypes.

Koza (1992) proposed genetic programming as a domain-independent method for evolving computer programs that solve, or approximately solve, problems. The technique of genetic programming is an extension of Holland’s work on genetic algorithms (see section 3.3.2). Koza (1992) identified what he called ‘a representation problem’ in genetic algorithms which limits their performance. Thus he substituted general, hierarchical computer programs of dynamically varying size and shape for the standard fixed length bit strings (Koza, 1992). This new representation allows genetic programming to incorporate more readily certain adaptively beneficial properties - such as iteration, recursion, and dynamic variability.

The basic premise of genetic programming is the same as that of genetic algorithms: large populations of computer programs are genetically bred and the Darwinian principle of survival and reproduction of the fittest, along with genetic recombination, are used to produce the next generation. Thus genetic programming starts with an initial population of randomly generated representations of computer programs consisting of functions and terminals appropriate to the domain. The functions can be standard mathematical operations, standard programming operations, logical operations or domain-specific functions. Each individual computer program in the population is measured in terms of how well it performs in a particular environment.

The reproduction process involves selecting, in proportion to fitness, computer programs from the current population of programs, and allowing them to survive by copying them or their offspring into the new population. The genetic process of recombination is

used to create offspring computer programs from two parents. The computer program has the form of a parse tree which allows swapping of subtrees in the process of recombination.

There have been two advances in genetic programming which are directly relevant to this thesis. First, automatically defined functions (ADFs) can be seen to be an implementation of Simon's stable intermediary forms. Second, architecture evolving operators gives a domain-independent account of the function of certain genetic operators used in Dawkins' embryologies. Each of these developments in genetic programming will be considered in turn.

The work described in Koza's (1992) first book *Genetic Programming* had the limitation that the vast majority of its evolved programs have a single part: one result-producing main part, but no subroutines. Koza (1995, p. 734) writes, 'I believe that no approach to automated programming is likely to be successful on non-trivial problems unless it provides some hierarchical mechanism to exploit, by reuse or parameterization, the regularities, symmetries, patterns, and modularities inherent in problem environments.' So Koza (1994) developed a technique for evolving multiple-part programs consisting of a main program and one or more reusable, parameterized hierarchically-called subprograms.

The subprograms are called automatically defined functions (ADFs). These are functions that are dynamically evolved during a run of genetic programming and which may be called by a calling program that is concurrently being evolved. Thus a population using ADFs consists of programs with a hierarchy of reusable function-defining branches along with a main result-producing branch.

Koza (1994) has shown that genetic programming with ADFs is capable of solving numerous problems. The main point is that using ADFs reduces the computational effort - the number of fitness evaluations - required to solve a problem (provided the difficulty of the problem is above a certain relatively low break-even point). The reason for this is that even if a compound function is to be used more than once it only has to be discovered once. Subsequently, programs can incorporate a number of instances of this function in different locations and the design can be optimised. Thus, as Koza's (1995) experiments show both learning efficiency and parsimony appear to be the properties of genetic programming with ADFs.

There is also evidence that the results obtained using genetic programming with ADFs is scaleable. Koza (1995) has shown that with several problems for which a progression of scaled-up versions were studied, the computational effort increases as a function of problem size at a slower rate with ADFs than without them. Also, the average size of solutions similarly increased as a function of problem size at a slower rate with ADFs than without them.

However, the use of ADFs leads to another problem: when ADFs are used it is necessary to determine the architecture of the yet-to-be-evolved program. The specification of the architecture consists of: a) the number of function-defining branches in the overall program; b) the number of arguments possessed by each function-defining branch; c) if there is more than one function defining branch, the nature of the hierarchical references allowed between the function-defining branches. While sometimes the architectural choices flow directly from the nature of the problem, Koza (1995, p. 734) indicates that, 'in general, there is no way of knowing *a priori* the optimal (or sufficient) number of automatically defined functions that will prove to be useful for a given problem, or the optimal (or minimal) number of arguments for each automatically defined function, or the optimal (or sufficient) arrangement of hierarchical references'.

Koza (1994) suggested a limited means of automating the process of architectural choice by starting with an initial population of diverse architectures and allowing selection to pick the fittest architectures. However, this procedure was incapable of producing new architectures on the run. To perform dynamic reconfiguration of architectures, Koza added biologically inspired procedures. Genetic operations sometimes lead to deletions and insertions of genetic material. For example, in nature this can lead to gene duplication - the same protein being coded for twice. If one of these genes is changed, then the original functionality can be maintained by the copy and the new functionality can be explored.

Koza defines six procedures based on insertion and deletion: 1) branch duplication; 2) argument duplication; 3) branch creation; 4) argument creation; 5) branch deletion; and 6) argument deletion.

Figure 4-2 shows that genetic programming with ADFs and architecture altering procedures can evolve solutions faster and produce more parsimonious solutions than genetic programming without these features. Thus ADFs and architecture altering

procedures can be viewed as an automated way to change the representation of a problem while solving the problem (Koza, 1995). These results support Simon's mathematical speculations described in the previous section.

Approach	Runs	E	W	S
1	14	5,025,000	36,950	469.1
2	14	4,263,000	66,667	180.9
3	25	1,789,500	13,594	88.8
4	25	1,705,500	14,088	130.0
5	25	1,261,500	6,481	112.2

Figure 4-2: Tests using the Boolean even-5-parity problem and five approaches: 1) without ADFs; 2) with ADFs and initial architecturally diverse population; 3) with ADFs and architecture altering procedures and crossover with point typing; 4) with ADFs and good choice user supplied architecture and crossover with point typing; 5) with ADFs and good choice user supplied architecture and crossover with branch typing. Koza used three performance criteria: computational effort, E (with 99% probability); the wallclock time, W (with 99% probability); and the average structural complexity, S (Koza,1995).

4.2.5 Summary on development

This section has focused on a number of factors that can affect both the rate of evolution and what can evolve. First, the developmental process was described as a biochemically implemented, algorithmic process - a modular, conditionally branching, hierarchical control system.

The role of self-organisation in evolution was examined with regard to both chaos and complexity theory. Chaos effects are characterised by complex behaviour from simple systems; complexity theory goes the other way with complex systems producing simple behaviour. The main point was that systems made up of interacting components can naturally have orderly behaviour which can greatly assist evolutionary search. For example, sparsely connected systems demonstrate a natural robustness to perturbation and graceful changes in behaviour when perturbed: that is, the gradual changes which natural selection favours. Furthermore, the natural robustness of sparsely connected systems creates neutral networks with the advantages discussed in section 3.3.3.

Hierarchy theory was described in sections 4.2.3. There are a number of reasons to believe that selection would favour modular and hierarchical structures. First, the ontological picture I put forward in chapter 2 (see section 2.4) suggest how new functionality is derived from either structural reorganisation or creating new structures (by, for example, adding to old structures). Second, Kauffman's patch logic (see section 3.3.3) shows how partitioned systems are less predisposed to premature convergence. Third, Simon showed how hierarchies of stable subassemblies can evolve faster than non-hierarchic structures (see section 4.2.3). As Koza's work with automatically defined functions (see section 4.2.4) shows Simon's claim has been successfully tested in one domain.

The main points so far are:

- That embryologies are algorithms with certain properties, for example, conditionally branching, modularity, and hierarchical organisation.
- A number of factors prevent evaluating the fitness of potential genotypes, for example, information about the program (genotype and egg), the input (initial conditions) and (as fitness is relative) the other phenotypes all needs to be known and the subsequent calculation would be formidable.
- Structures form spontaneously from self-organisation which can aid the process of natural selection. Self-organisation can reduce the number of possible states a system will go into and certain systems, those of the ordered regime, exhibit a stability that produces the gradual changes most easily favoured by evolution. Also self-organisation can provide the initial order required for processes to get started, like the origin of life (see page 47).
- Certain abstract design properties, such as modularity and hierarchical organisation, can increase the evolvability of a design. For example, using a modular representation in the genotype creates discreet and thus recombinable features.
- A modular design has a number of advantages, for example, it allows reuse of components and thus removes the need of having to evolve the component each time, and it forms Kauffman's patches reducing the risk of structures getting caught in local optima.
- Simon's shows how hierarchically-organised structures can evolve faster than flat structures by allowing sets of building blocks to accumulate at different levels which prevent higher level building blocks having to be constructed from scratch.

To summarise, development can affect evolution in the following ways:

- Constrained embryologies can lead to useful trial-selectivity thereby reducing the amount of trial and error necessary.
- The work on self-organisation shows how it may be possible for different types of structure to facilitate or inhibit adaptive change.
- Embryologies with a modular design and/or hierarchical structure can in certain circumstances evolve faster than embryologies without these abstract design principles.
- Phenotypic plasticity produced by the developmental program can guide genetic evolution by allowing genetic assimilation.

In the final half of this chapter some computational experiments are described which show that evolvability can be selected for.

4.3 Reproductive mechanisms and the selection of evolvability

The aim of this work is to explore how an evolutionary process can select the mechanisms that lead to the fastest rate of adaptive change. The computational experiments used to test this hypothesis were carried out in two stages: first, two different generators of variation were chosen and their rates of adaptive change measured for a range of population sizes and on different fitness landscapes. The exact type of generator was unimportant, all that mattered was that the two generators of variation performed differently in different environments: that is, produced different offspring. The second stage of the experiment was to see if in populations in which both generators were present that the generator predicted from the results of the first stage was selected. It was found that in populations containing two generators the evolutionary process was able, in most cases, to select the better generator. In some cases, however, the other generator was selected and this was found to be due to the short sighted nature of selection.

It should be noted that work has already been done in evolutionary computation comparing different generators of variation (e.g. Holland, 1994) and in the computational technique of evolutionary strategies on selecting generator parameters (e.g. Salomon, 1996).

However, the aim of this work is to demonstrate, using simple computational experiments, that evolvability can be selected and to emphasize that there are differences in kind, not just in degree, between generators. For instance, the mutation-and-recombination generator is not simply a parameterised mutation generator and thus cannot be selected using the computational technique of evolutionary strategies unless it is originally part of the specification.

4.3.1 Introduction

Let us start by considering a hypothetical example: imagine there are two equally fit organisms, O_1 and O_2 . Both organisms have the same phenotypes, but they have different genotypes. The consequence of the difference in the genotypes is that the two organisms have different generators of variation (genetic systems), G_1 and G_2 respectively which are inherited by O_1 and O_2 respectively. This leads to the offspring of O_1 being on average fitter than the offspring of O_2 . Through time selection will favour the offspring of O_1 because they have a higher fitness than the offspring of O_2 . Thus the generator of variation G_1 will also be selected. This, of course, depends on how close the original organisms are to maximum fitness.

This example illustrates two levels of selection: direct selection of the individuals and indirect selection of the generator of variation - the mechanisms implementing the Darwinian process. Dawkins (1989a) makes a similar distinction when he discusses the two types of mutations: 1) mutations that are ordinary changes within an existing genetic system, and 2) mutations that are changes to the genetic system itself. It is the different genetic systems (G_1 and G_2) in our example that are responsible for producing the different offspring.

Another way of describing our example would be using two spaces: the space of possible organisms, and the space of possible generators of variation. A generator of variation is in principle a mechanism for simultaneously searching *both* spaces. While the fitness of an individual can be measured in the individual's lifetime, the quality of a generator has to be measured in species (lineage) time. Through species time the generators of variation operate on themselves (as the generator is coded in the genotype it can be changed) and as in our example the 'better' generator(s) of variation can be selected.

However, search involves both exploration of the space of possibilities and the exploitation of certain possibilities. In Darwinian evolution the genetic system implements exploration while selection exploits certain products. This suggests a means of evolving evolvability: by constraining the variation. In biological evolution variation is constrained by, for example, the genetic mechanisms, the embryology and selective mating.

As already mentioned, Herbert Simon (1982), one of the early champions of artificial intelligence, emphasised the advantage of *selective* trial-and-error as a way of reducing the amount of actual trial-and-error required. Darwinian processes carry out a form of selective trial-and-error. As the next generation is related to the last generation, exploitation constrains exploration to certain regions of the search space. That is to say, if ‘like begets like’ (Darwin, 1859), then selection reduces the coverage of the search space and recursively searches the remaining regions on an increasingly fine scale. Thus the individuals in a population and the search mechanisms available define the search space.

In the following experiments the central hypothesis is: in a population of variants it is possible for a Darwinian process to select the generator(s) of variation which produce the fastest rate of adaptive change. Computational experiments were used to test this hypothesis.

Two different generators of variation (mutation and mutation with recombination) were used to produce the next generation. These generators were used because they are common in the literature and simple to implement. Many other generators of variation are possible: for example, logical operators which deduce new variant forms from the successes and failures of past instances.

The fitness of the variants was determined using an NK-landscape algorithm (Kauffman, 1995), the implementation details of which are explained in section 4.3.2. The NK-landscape algorithm was used for two reasons: first, it is a well-documented algorithm; second, using an arbitrary fitness landscape helps avoid the problem of coding the solution into the problem.

The emphasis in this work is on the rate of change of fitness through time not on arriving at a given state (e.g. the global optimum). The rate of adaptive change was calculated in two ways to allow the quantitative comparison of performance between the generators. The first measure was obtained by calculating the gradient of a run; the second measure was obtained by calculating the change in fitness between the start and the finish of

a run. A third way of measuring the rate of adaptive change - by counting the number of generations to reach a given fitness - was ruled out because to check that the given fitness was not greater than the global optimum on a particular landscape required an exhaustive search of the NK-landscape.

The first section outlines the experimental methodology. This is followed by sections detailing the results, a discussion of the results, and a summary.

4.3.2 Method

This section describes *what* the experimental algorithms do (the functional architecture) without going into details of *how* they do it (the implementation details). First, there is a description of the ontology (the things present in these experiments), then a description of the generate-test-regenerate algorithm, followed by an outline of the evaluator mechanism (the NK-landscape algorithm) and finishing the section with a description of the methods of calculating the rate of adaptive change. There is also a description of the programs and information about how to access them in appendix 1.

The ontology consisted of *variants*, *generators*, and *selectors*. The variants were simple data structures consisting of a bit-string genotype (100 bits long), a record of the type of generator which produced them (mutation or mutation with recombination), and a fitness value (1-100).

The two generators used were mutation and mutation with recombination. The mutation generator involved flipping a randomly selected single bit of the genotype. The recombination generator involved a randomly generated single point crossover which produced a single offspring (if A = 0000 and B = 1111, then crossover at 3rd point = 0011).

The fitness of a variant was calculated using Kauffman's (1995) NK-landscape algorithm. The NK-landscape was implemented as follows: there are two parameters, N the number of genes, and K the number of links between genes. Thus the fitness of a gene (bit) is dependent on a set of K locations randomly picked from anywhere on the genotype. A series of N lookup tables are generated, one for each gene location in the genotype. As each gene had 2 possible alleles (0 or 1) then there were 2^{K+1} possible combinations of alleles and hence 2^{K+1} random entries in each lookup table between 1 and 100.

The fitness of a genotype was determined as follows: each gene was evaluated by inspecting the fitness value of its allele and of the alleles of the genes it was linked to. This allele combination was used as a key to access the lookup table and determine the fitness of the gene. When all the genes had been inspected their individual fitness values were summed together and divided by the number of genes (N). All the genotypes had a fitness value between 1 and 100.

A tournament selection algorithm, the selector, was used to pick the parent variants:

1. pick a random sample from the old population (e.g. sample size = 5);
2. return fittest variant from sample (if there is a tie, then randomly select from those that came first).

The evolutionary algorithm, the Darwinian process, had the form of the generate-test-regenerate heuristic:

1. produce a random population (population size varies);
2. evaluate each member of the population using the NK-landscape;
3. create the next generation using the tournament selection algorithm to pick the parents and the generators of variation to produce the offspring, return to step 2.

Two procedures were used to measure the rate of adaptive change. The first procedure involved calculating the mean fitness change in the population from the start of a run to the end of a run. Thus if the mean fitness at the start of the run was F_1 and the mean fitness at the end was F_2 , then the change in fitness was $F_2 - F_1$.

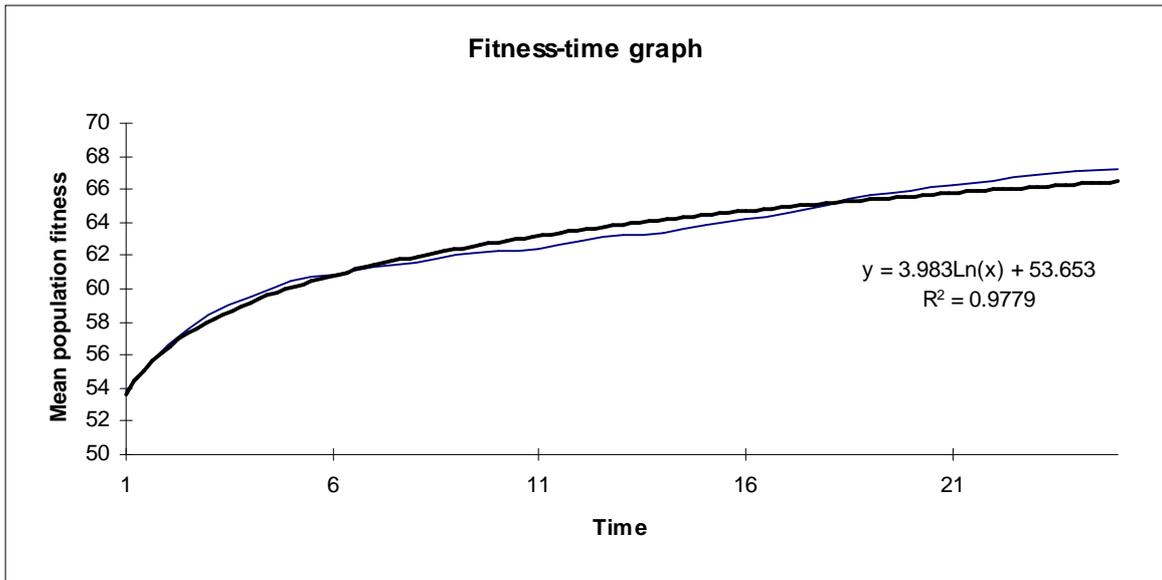


Figure 4-3: This is a fitness-time graph of a single run of a mutation generator on a $N = 100$, $K = 1$ landscape. $F1 = 53.85$ and $F2 = 67.20 \therefore F2-F1 = 13.35$. A logarithmic trend-line has been added to the graph and the fit is so close ($R^2 = 0.9779$) it almost completely obscures the actual line.

The second means of calculating the rate of adaptive change was to raise 10 to the power of the fitness value Y (actually Y divided by 10 to prevent overflow). Then a straight line was fitted using linear regression and the quality of fit tested (see Figure 4-4). The gradient of the line was obtained from the linear regression. This allowed for greater ease in comparing the gradients.

Both methods of calculating the rate of adaptive change have problems as the fitness-time graphs showed a rapid increase in fitness at the outset that tailed off towards an asymptote (see Figure 4-3). Thus the information about the rate of the early increase in fitness is lost, which would particularly be a problem if the population found the global maximum. To reduce the risk of finding the global optimum the size of the landscape and the number of time steps was adjusting making this possibility highly improbable.

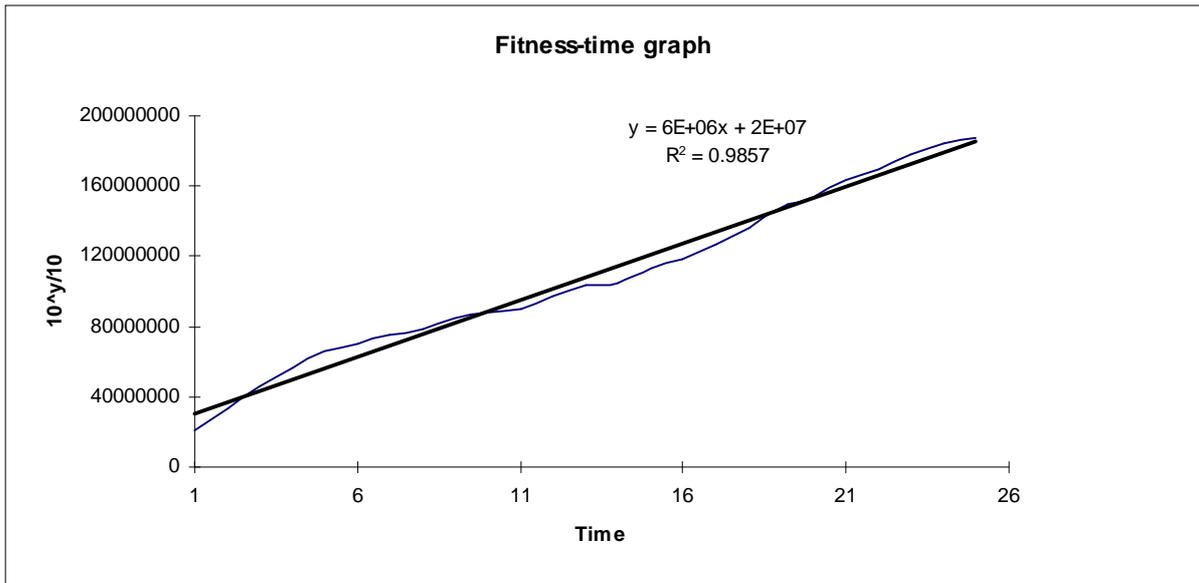


Figure 4-4: In this fitness-time graph, fitness was plotted as $10^{y/10}$ and then a straight line was fitted. The quality of the fit (R^2) is 0.9857, and the gradient (the rate of adaptive change) is 6000000.

The overall experimental design was as follows: there were three variables: the K-value of the landscape, the population size and the generator type. The number of genes in the landscape was kept constant at $N = 100$, while the K-value was varied from 0 to 9. A greater range of K-values was not used as the size of the data type representing an NK landscape increases exponentially in relation to the K-value. An approximation of the size of the data type representing an NK landscape can be calculated from the sum of the seven bit data entries in the fitness lookup tables - $7.N.2^{K+1}$ (e.g. if $N = 100$ and $K = 20$, then an NK landscape has an approximate size of 1468 MB). The population sizes were varied from 10 to 1280 variants by doubling the population size at the end of each run: 10, 20, 40... 1280. Varying the population in this way produced a population size variation over two orders of magnitude; memory constraints inhibited using larger populations. Forty runs were carried out for each population size and K value. The number 40 was reached by starting with a single run, calculating the mean fitness and then doubling the number of runs until there was negligible further change in mean fitness. Three experimental runs were carried out: mutation alone, mutation with recombination, and a combination of both generators.

4.3.3 Results

Figure 4-5 shows the effect of population size on the rate of adaptive change. Figure 4-6 and Figure 4-7 show on the left the generator that was selected and on the right the generator with the fastest rate of adaptive change: that is, these figures show whether or not the generator with the fastest rate of adaptive change was selected. Figure 4-6 shows: 1) the generator selected: either mutation or mutation with recombination and 2) the generator which performed best when tested alone. Thus the squares in the 2-6 K value range and the 640-1280 population size range show deviation from the expected results.

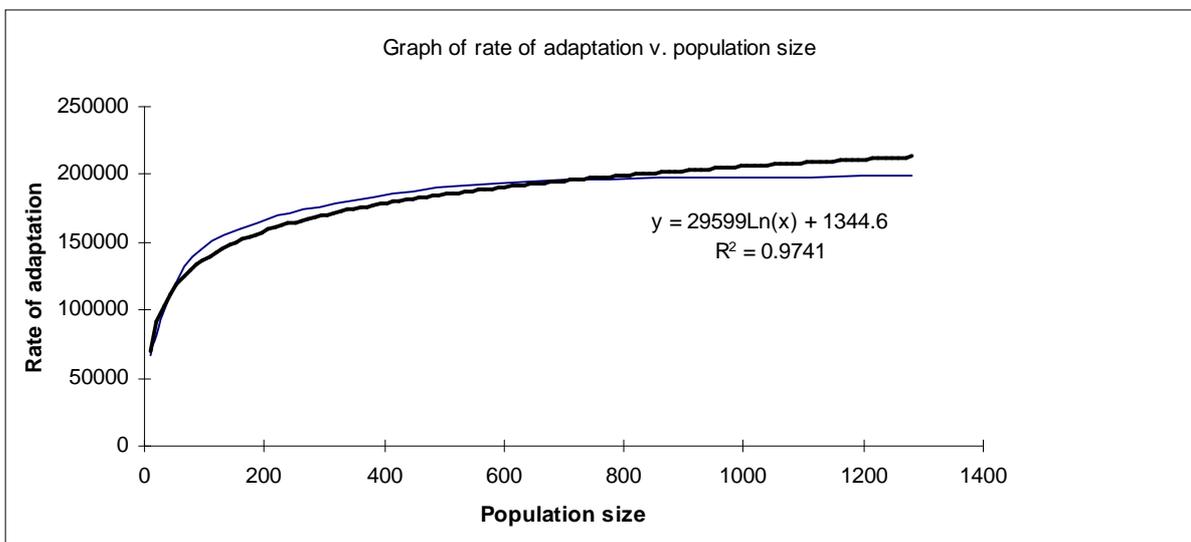


Figure 4-5: Rate of change in mean fitness against population size for the mutation generator running on $N = 100$, $K = 0$ landscape. The thick, dark line is a logarithmic trend-line that has been fitted results.

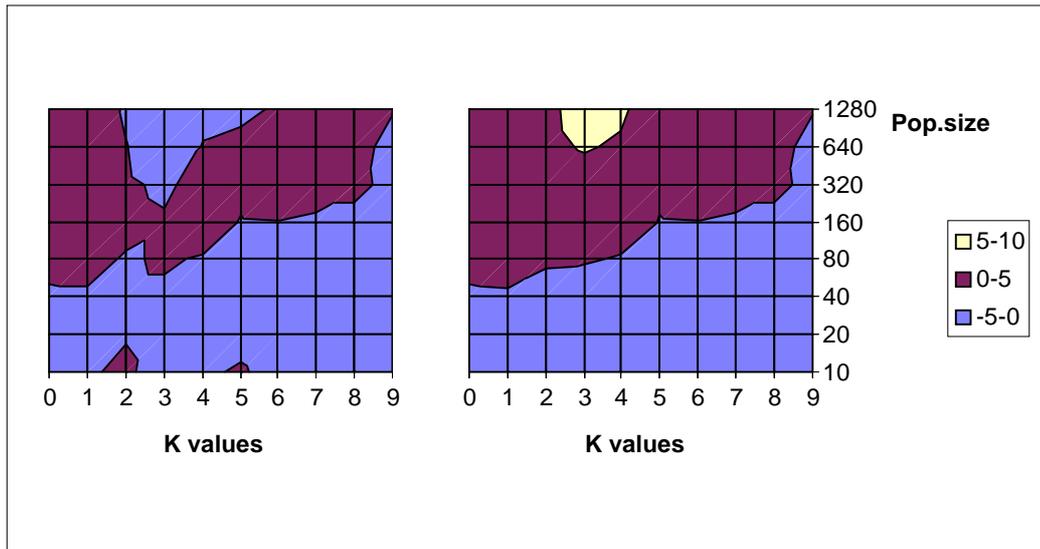


Figure 4-6: The calculation of these maps involved subtracting the rate of adaptive change of one generator from the other using the following formula: $Y = |C-M| - |C-R|$, where C = combined change in fitness over time (df/dt), M = mutation df/dt and R = recombination df/dt . Thus where Y was positive (dark and white), the recombination generator was favoured; where Y was negative (light but not white), mutation was favoured. The first map shows which generator was picked; the second map shows the generator that performs best in the same situation when not competing. The df/dt values used in these calculations were obtained by subtracting the initial mean population fitness from the final mean population fitness.

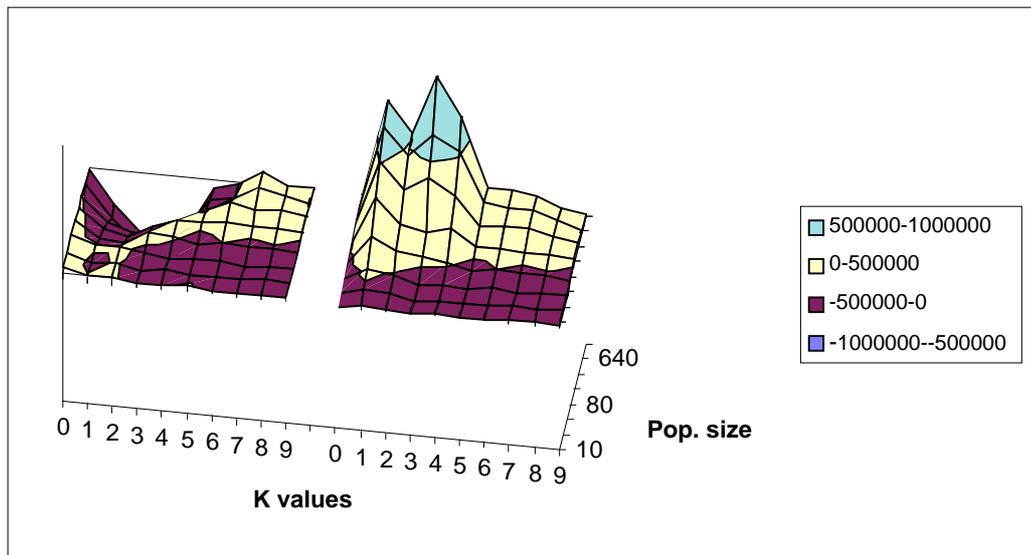


Figure 4-7: These maps were created using the same technique used in **Figure 4-6** while using the mean gradients of the fitness-time graphs. The map on the left shows the generator selected (negative numbers when mutation was picked, positive numbers when mutation with recombination was picked), the map on the right the generator that was expected to be selected.

4.3.4 Discussion

The experimental hypothesis was that a Darwinian process could indirectly select the better generator of variation by selecting the better products. However, a number of variables other than the generators were found to effect the rate of adaptive change, for example, population size and the K value of the landscape.

Figure 4-5 shows the effect of population size on the rate of adaptive change: when $K = 0$ the trend-line was approximately logarithmic. Thus, as the population size increases the rate at which the population adapts followed a law of diminishing returns. It should be noted that this trend was not constant across both generators and on all landscapes. For example, the mutation with recombination generator, especially at $K = 3$, had an increase in the rate of adaptive change which was approximately linear: that is, directly proportional to the increase in population size.

Taking into account the effects of population size and the K value neither generator outperformed the other in all cases (see Figure 4-6 and Figure 4-7). When the population sizes were between 10-80, the mutation generator performed better than the crossover generator. Also as the K value increased, the size of the population in which the mutation generator outperformed the recombination generator increased. There were two small perturbations when the population size was 10 and $K = 2$ and $K = 5$. However, these small populations are very unstable as the initial effect of selection can easily remove either generator from the population before the generators have had time to be evaluated.

An interesting result was the large perturbation that occurs in the population sizes ≥ 320 when the K value was between 2 and 5. This is clearly shown in Figure 4-7 where there is a large trough where a peak is predicted: that is, where the predicted generator is not selected. The calculation of these landscapes involved subtracting the rate of adaptive change of one generator from the other. Therefore, the height or depth of a point on the landscape is proportional to the difference between the generators' rates of adaptive change. Figure 4-7 shows that the recombination generator was much faster than the mutation generators in certain cases. For example, when $K = 3$ and population size = 1280 the mutation generator produced an average change of fitness of 17.5 while the recombination generator produced an average increase in fitness of 24. In this case it is clear that

recombination was the faster generator, but this was not the generator selected: out of 40 runs, when $K = 3$ and population size = 1280, the recombination generator was selected in only 11 cases.

Whilst trying to explain this perturbation two assumptions were called into question: 1) the effect of early selection; 2) the effect of generator competition. Trials were carried out to test the effect of selection by examining a population with no variant generator acting (see Figure 4-8). In the early stages of a trial, selection exploits the fitter variants and thus the mean population fitness increases. It was assumed that this would be the same for both generators and so could be ignored. However, the recombination generator seems to be more sensitive to changes in population size which could result from early selection.

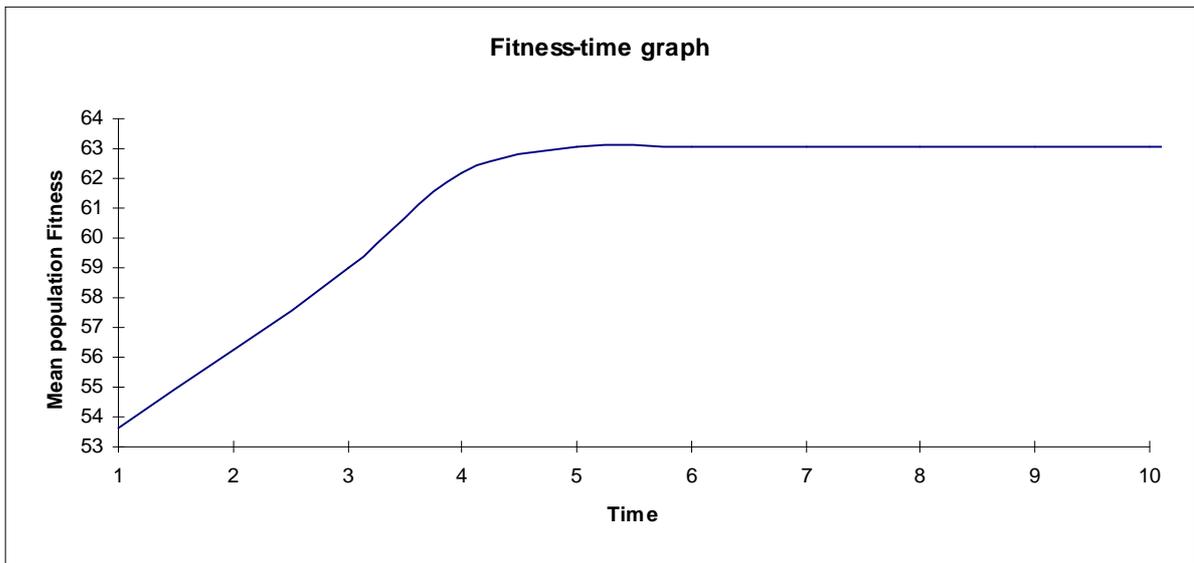


Figure 4-8: This fitness-time graph was produced in a population with *only* selection acting. This shows the initial increase in mean fitness that can be attributed solely to selection: in this case from 53.62 to 63.06.

The second false assumption was that a population of one type of generator would adapt at the same rate as a population with the two generators competing. Both generators suffer from a reduced population size: the generator alone has a population size of X while the generator in competition has a population size of $X/2$. Also, as mentioned before, the recombination generator seems to be more sensitive to changes in the population size than the mutation generator.

The cause of the perturbation, when the generator with the fastest rate of adaptive change was not selected, was found to be due to the myopic nature of search (in this case Darwinian evolution): that is, the short-sighted selection of individuals with the greatest immediate gain. Mutation combined with selection appeared to produce a faster increase in fitness than mutation with recombination and selection in the initial stages of a trial (see Figure 4-9). This was probably because it took the recombination generator time to build up a set of building blocks: that is, high fitness but radically different variants were initially being crossed unsuccessfully. The mutation generator, on the other hand, was producing subtle variations on successful variants. This, however, begs the question: why was the mutation with recombination generator successfully selected in so many cases?

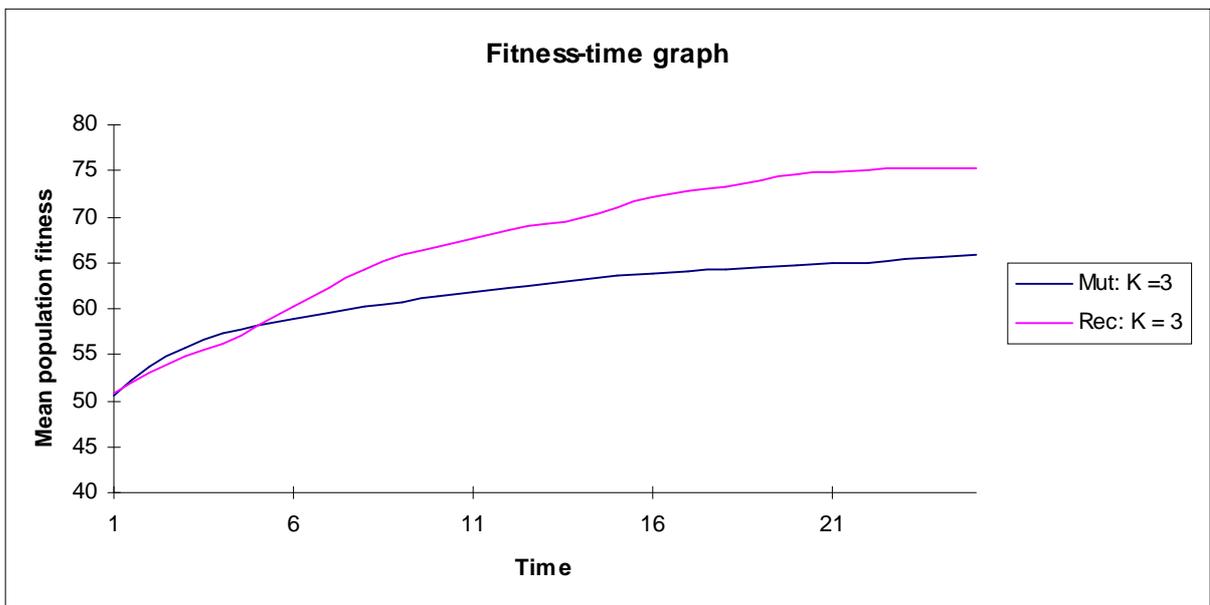


Figure 4-9: Mutation and recombination generator, on a $N = 100$, $K = 3$ landscape with a population size of 100.

To explain why the generator was picked successfully in so many cases we need to look at the shape of the fitness-time graph in a case when the predicted generator was picked. What was found was that depending on the K value of the landscapes, different trends were obtained (see Figure 4-10). In the lower K -value landscapes recombination allows larger jumps and thus produces a faster rate of adaptive change. However, when the K value was high both generators show a reduced performance.

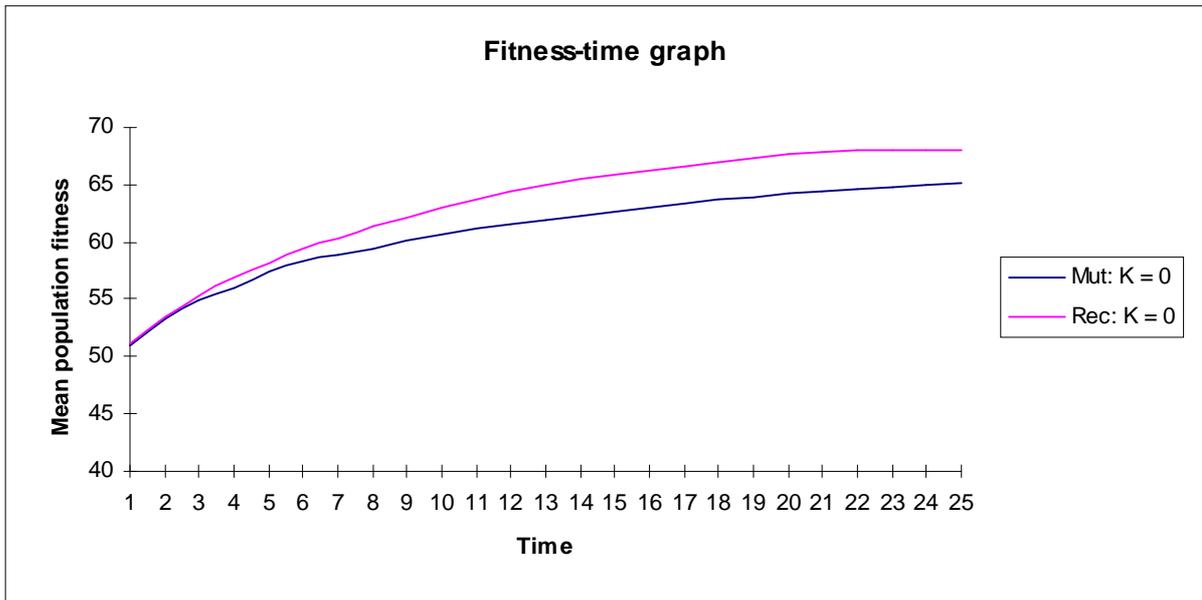


Figure 4-10: Mutation and mutation with recombination generators on a $N = 100$, $K = 0$ landscape, population size of 100.

In addition to the main experiments comparing selection of generators that differed in *kind*, a simple experiment was run comparing selection of generators that differed in *degree*. In these experiments selection optimised the generators parameters. Koza's (1995) work (see section 4.2.4) demonstrates how the genetic systems can invent new structures and possibly new generators. However, this experiment tested how the parameters of a particular kind of generator of variation could be optimised in a specific context.

A simple experiment was carried out in which the mutation generator operates with a probability P . This probability could itself be mutated. Figure 4-11 shows what happens when this experiment is run. There is an initial period when the mean probability of mutation increase exploring the space of possibilities, then the mutation rate begins to drop as selection exploits the better variants. The mutation rate stabilised with a probability of mutation of 0.0191. The gradual reduction in mutation rate through time is reminiscent of simulated annealing in which there are initially large jumps and the distance between these jumps is reduced through time.

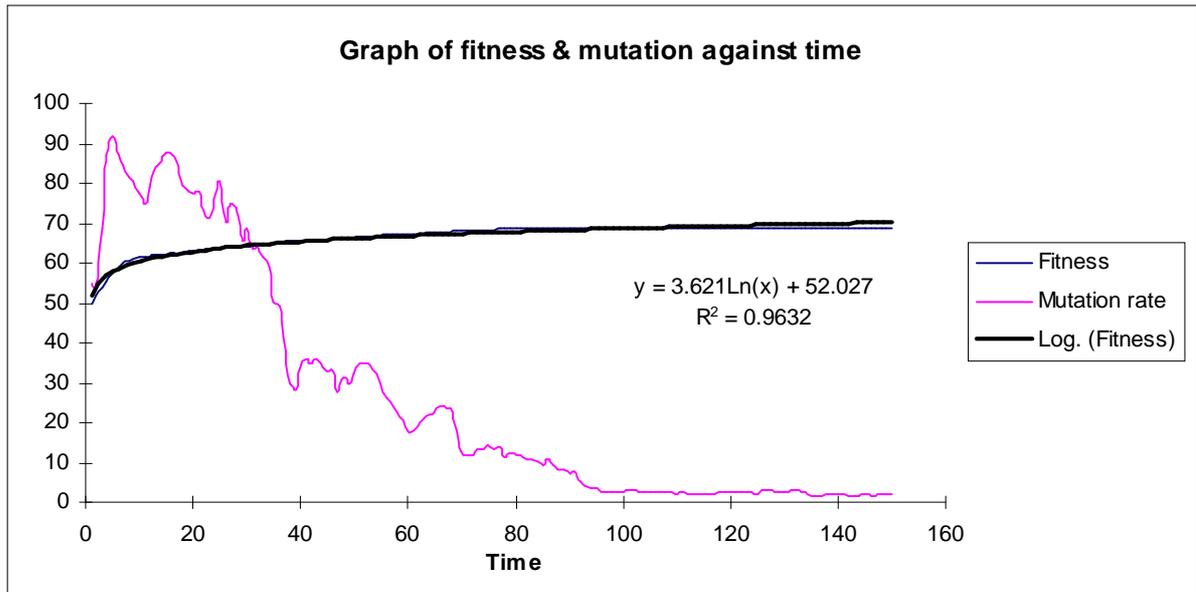


Figure 4-11: shows 1) the rate of increase in mean population fitness on a $N = 100$, $K = 0$ landscape (the dark line with a fitted trend line), and 2) the probability of mutation.

Let us consider some general points about this work. These experiments imply that selection of the fittest agents will indirectly select the generators of variation which produce the fastest rate of adaptive change. This effect of selection works on the basis of the fitness of the agent produced and not on the kind of generator: that is, any factor which increases the fitness and is heritable will be favoured. In other words, what is important is the product and not the specific process; a different process which produces the same product will be equally favoured.

However, these results have shown that the mechanisms' performance is conditional on certain variables. Thus if these variables change as a result of the mechanisms' actions, then a different type of generator may become 'better'. For example, while asexual reproduction is more robust when the population size is small, as the population size increases sexual reproduction may produce a faster rate of adaptive change. Note the similarity of this result to that of the 'no free lunch theorems' (Wolpert and Macready, 1995) which proves that algorithms that search for an extremum of a cost function perform exactly the same, according to any performance measure, when averaged over all possible cost functions.

These experiments also demonstrate how a group property, the population size, can affect individual fitness and hence what is selected. However, selection is clearly acting at

the level of the individual and indirectly selecting the group because its individuals are the fittest (or evolving the fastest). Note in these experiments the reproductive species concept is used: that is, reproductive compatibility (Ridley, 1990). The morphological species concept - morphological similarity, not strict identity - is ignored in these experiments.

Some preliminary attempts were made to implement sexual selection on the basis of genotypic similarity. This was done by mate selection being on the basis of fitness multiplied by genotypic similarity (the number of bits in common). These experiments were unsuccessful. It appeared that more sophisticated strategies were needed to select mates. However, Todd (1996) found that the selection of mating strategies relying on phenotypic features led to the formation of diverging sub-populations: that is, sexual selection produced speciation.

Sexual reproduction produces a parallel search where groups of individuals can share their findings in the next generation. For example, if one agent has good hearing and another agent has good sight, these traits could be combined in a single generation (Maynard Smith, 1994). Thus even though an agent can only pass on half of its genetic material, if the resulting offspring are fitter than its rivals' offspring, then there will be an indirect selection of sexual reproduction. The other advantages of sexual reproduction - for example, avoiding the problem of Muller's ratchet (see section 3.3.2) and in the natural world where agents pass on pairs of chromosomes that form neutral networks (see section 3.3.3) - all add to the strength of the argument that natural selection will favour sexual reproduction.

It should be noted that there are a number of simplifications in these experiments: for example, there is a direct genotype-phenotype mapping and the agents have no constraints on who they can mate with or how many times they can mate. Thus the advantages of patches, neutral networks and constrained embryologies are all lost.

4.3.5 Conclusions and further work

The hypothesis being tested was that a Darwinian process could select the mechanism that produces the most rapid adaptive change. While this proved to be true in most trials there was a deviation from the expected results. The reason for this perturbation was that selection is a short sighted process: that is, variants are selected on the basis of their current

performance not their potential future performance. As the asexual individuals produced the faster initial increase in fitness, the asexual generator was selected.

The implications of these experiments are not limited to biological evolution. The ability of the Darwinian algorithm to select its generators is derived from its logical structure and not its physical form. For example, it is possible to imagine cases in science where the processes which produced the best results have been adopted. However, as these experiments show, there are some important cases when this approach does not work.

There are many possibilities for further work. The experiments could be run again with more runs so that the results could be analysed statistically. A broader range of population sizes and K values could be explored. The third measure discussed in section 4.3.1 could be investigated. Other generators of variation could be tested and the work aimed at parameterising these generators continued.

4.4 Chapter Summary

In this chapter the potential roles of development in evolution have been examined. This has included a discussion of the role of self-organisation in adaptive change and the evolvability of different embryologies. Finally, a set of computational experiments was described which demonstrate in the simple situation tested that agents with mechanisms which did not directly change their phenotypes, but which changed those of their offspring are selected by evolution if they produce faster adaptive change than other mechanisms. However, these experiments also demonstrated that there are a number of variables which affect the rate of adaptive change and different mechanisms are sensitive to different variables.

The main points of this chapter are:

1. The genotype has two roles: one in development and the other in reproduction.
2. The developmental program is a parallel, conditionally branching, hierarchical control system. Parallel in two senses: first, within a cell there is concurrent transcription of DNA; second, between cells there is concurrent differentiation. The conditions can be self-referential (internal) or environmental (external) and are hierarchically organised as there are genes called homeotic genes which switch on or off parts of the program.

3. Self-organisation has gained support as a factor in evolution. This can be seen in a number of ways: the DNA does not code for all the properties of a structure, for example, the 3D structure of proteins; structure arises spontaneously within networks, for example, autocatalytic chemical cycles will naturally occur. Thus there is inherent structure for selection to work with, and some of that structure (for example, the structure produced by sparsely connected networks) is highly susceptible to Darwinian evolution by natural selection.
4. Dawkins' (1989a, 1996) embryologies show how Simon's (1982) second form of selectivity (reuse of previous solutions) can be implemented. Simon's mathematical analysis of stable subassemblies show how embryologies that use subassemblies could evolve faster than those without them. This claim has been strengthened by the work of Koza (1994, 1995) that shows that genetic programs with ADFs (stable subassemblies) evolve faster than those without them.
5. To produce open-ended search spaces Koza (1995) introduced the genetic operators of deletion and insertion into genetic programming. Genetic programs evolved using these genetic operators adapted faster than programs evolved without them.
6. The thesis in the second half of the chapter was that a selective system could indirectly select the generator of variation which resulted in the faster rates of adaptive change. This hypothesis was corroborated, but not in all cases.

5. Trajectories in design space

5.1 Introduction

Bonner (1980) distinguished between genetic and cultural evolution by stressing the way information is transmitted: in the former-genetically and in the latter-behaviourally. The discussion so far has focused on genetic evolution and its mechanisms, particularly those that can speed up the process of adaptive change, for example, heredity, recombination, and hierarchical organisation. There is, however, another class of adaptations which can increase evolvability: the mechanisms that produce cultural evolution. The aim in this chapter is to extend the previous discussion of Darwinian evolution by incorporating the mechanisms that can bring about change within the lifetime of an individual and allow individuals to share this knowledge.

However, examining genetic and cultural evolution in the real world or the laboratory is very difficult. An animal would have to be chosen that had a short lifetime so that it could be observed over successive generations, and this animal would be required to demonstrate both genetic and cultural evolution. Furthermore, there is the notorious difficulty of discriminating between innate and acquired characters: in other words, between genetic and cultural contributions to a feature. The result is that comparing and contrasting genetic and cultural evolution in the real world is, putting it mildly, problematic.

There have, however, been a number of mathematical models of cultural evolution (e.g. Cavilli-Sforza and Feldman, 1981; Lumsden and Wilson, 1981; Boyd and Richerson, 1985). But as May (1977) points out, 'formidable mathematical difficulties stand in the way of a fuller understanding of the interplay between cultural and biological evolutionary processes'. For example, the general equations relating gene frequencies in successive generations are not only non-linear, but also involve the frequencies from earlier generations (May, 1977). Also the mathematical theories of cultural evolution tend to model the cultural dynamics and pay less attention to the mechanisms which make it happen (Gabora, 1997).

While there are a number of difficulties with the usual experimental and mathematical approaches, there is an alternative approach: the computational approach. By using a computational approach the dynamics of a genetic-cultural evolutionary system can be modelled, and to implement the model requires functionally specifying the mechanisms

involved. The use of computational models has a number of other advantages: it avoids the lifetime problem encountered while observing real world systems - that is, simulations of evolution need not take billions of years; it allows the experimental subject to be custom-made, for example, agents with the required genetic and social mechanisms; it removes the nature-nurture problem as implementations can be designed that make it possible to discriminate the relative contributions of each factor; it allows the direct observation of the agents' internal states, both genetic and psychological.

For the reasons given in the previous paragraphs (also see chapter 2) a computational approach was taken to investigate the relationship between genetic and cultural evolution. With the growth of work in artificial life and the advent of journals like the *Journal of Memetics* there has been increasing interest in cultural evolution (for a recent review see Gabora, 1997) and a number of computer models have been produced (e.g. Ackley and Littman, 1994; Gabora, 1995). However, these researchers have focused on modelling cultural evolution independently of genetic evolution: in the case of Ackley and Littman (1994) this meant separately testing Darwinian and Lamarckian evolution and comparing their rates of adaptive change; in the case of Gabora (1995) this meant ignoring genetic evolution altogether. In neither case were genetic and cultural evolution implemented within the same agent.

One of the aims of the work described in this chapter was to investigate the consequences of having both genetic and cultural evolution running concurrently. The work in this thesis focuses on how different mechanisms interact and the computational experiments described here implement agents with a broad range of capabilities (albeit implemented in a 'broad but shallow' manner - see section 2.3).

The set of agents used in the computational experiments described in this chapter also define a trajectory in design space. The trajectory is formed by the following sequence of agents: agents in the first class were adapted through genetic evolution. The agents with hardwired phenotypes are called 'Darwinian creatures' after Dennett's (1986, 1995, 1996) classification of agents. This kind of agent forms the starting point of the trajectory in design space.

Dennett called agents 'Skinnerian creatures' if they had a certain degree of phenotypic plasticity. These agents are not wholly designed at birth, but have elements of their design

that can be adjusted by events occurring during field testing. Skinnerian agents are divided into two classes: first, agents with a fixed set of behaviours, and with the capability of reinforcing the most useful ones; second, agents capable of reinforcement learning and of exploring the space of possible behaviours.

‘Dawkinsian creatures’ are so-called because of the nongenetic heredity mechanism postulated by Dawkins (1989b). Dawkins described a new replicator, a unit of cultural transmission or a unit of imitation, which he calls a meme. Dawkinsian creatures have the capability to imitate other agents and thus behaviourally transmit information.

The final class of agents are called ‘Vygotskian creatures’ after the Russian psychologist, L. S. Vygotsky. Vygotsky, like Piaget, saw children as active constructors of knowledge and understanding, but he differed from Piaget in his emphasis on the role of direct intervention by more knowledgeable others in this learning process (Smith and Cowie, 1991). Vygotskian creatures are capable of instructing others, that is of behaviourally transmitting knowledge to other agents.

To sum up, while the idea of linking genetic and cultural evolution is not a new one, there are a number of difficulties standing in the way of testing theories about the relationship between them. The theorising of the evolutionary epistemologists is, I think, the clearest formulation of the link between biology, psychology and socioculture and will be described in the next section. In section 5.1.2, the basis for the adaptive agent (Holland, 1986, 1994, 1995) used in the computational experiments is described: the classifier system. The subsequent section describes the design-work and the computational model produced when implementing some of the evolutionary epistemologists’ ideas. This chapter is structured like a standard experimental write up with the following sections: introduction, method, results, discussion and conclusion.

The novel contribution to research in this chapter is, by implementing the different kinds of agents, to provide an analysis of the requirements and properties of agents capable of both genetic and cultural evolution.

5.1.1 Evolutionary epistemology

While Darwin, Huxley and Spencer all alluded to the generality of evolutionary theory, William James (1880) was the first to use the analogy in its Darwinian form as the basis for

an extensive analysis of thought and problem solving (Plotkin, 1988). Since then scores of writers have employed the analogy in the same context: for example, Baldwin, Campbell, Dawkins, Dennett, Edelman, Piaget, Popper, Lorenz, Skinner and Simon. Recently Dawkins (1989b) when discussing the role of culture in evolution postulated a second heredity system - the memetic system; other researchers (e.g. Moravec, 1989, p. 167) talk of another 'genetic take-over' where genes have 'finally outsmarted themselves'.

Campbell (1976) produced an extensive review of evolutionary epistemology. In this article he identifies ten levels in the evolutionary process extending from the genetic level to the level of learning, thought and science. However, for the purpose of this section a simpler multiple-level model developed by Plotkin and Odling-Smee (1981; Plotkin, 1988, 1995) will be described.

A basic tenet in evolutionary epistemology is that the genetic-level accounts of evolution are insufficient to explain all the observed phenomena such as complex, learned and social behaviours. Thus a multiple-level, multiple-process account is substituted which incorporates at least three domains: the biological (genetic and developmental), the psychological and the sociocultural.

Plotkin and Odling-Smee's (1981) evolutionary model originally employed four levels: genetic, developmental, psychological and sociocultural. Subsequent criticism of the developmental level has led to it being incorporated within the genetic level. For example, Campbell (1981) pointed out that development, via the processes involved in it, expresses the already stored information, but is not involved in the generation of the information.

However, development is epigenetic, that is, a conditional, self-referential, dynamic and probabilistic process. Due to epigenesis, adaptations are not invariant structures, but vary widely depending on the environments in which development occurs (Plotkin, 1995). This developmental plasticity can be seen as an adaptive device - a knowledge-gaining mechanism. For example, 'If the coat thickness of a mouse varies with temperature during development, then not only is the adaptation of the thickness of the coat a form of knowledge of the temperature of the world of that mouse, but the relatively short-term process of development itself, together with the long-term process by which the genes that control such flexible development are selected, can be seen as an integrated process by which that knowledge is gained' (Plotkin, 1995, p. 124).

Plotkin (1995) calls the first level of the model the 'primary heuristic'. As the primary heuristic has been examined in detail in the last two chapters, we will quickly move on to the next level of Plotkin and Odling-Smee's model. The next level, the psychological level, gains information by the processes of learning and stores information in the central nervous system. Plotkin (1995) calls the psychological level 'the secondary heuristic'.

Plotkin and Odling-Smee's model forms a nested hierarchy, that is, the higher levels are embedded in the lower levels. For example, it may not be possible for evolution to specify goal-satisfying behaviour, but a set of behaviours and a reinforcement mechanism selected by the primary heuristic could allow an agent to select its own behaviour through interacting with the environment. Thus the psychological level is nested in the biological level because the central nervous system is part of the phenotype which is, in turn, partially specified by the genotype (Bonner, 1980). However, Lewontin (1981) has made the point that as well as saying that one thing constrains another, it is important to say to what extent. A celibacy gene, for example, would immediately be selected against in Darwinian creatures, whereas a celibacy meme can be propagated through culture. Another example is given by Bonner (1980): male sopranos were castrated in the west to keep their singing voices, but castration is a very bad idea if there is any genetic basis for singing ability.

Plotkin (1995) emphasises how some changes in the world occur at a rate too fast for the primary heuristic to accommodate, for instance, the location of prey or the identity of friends. The secondary heuristic has a similar function to that postulated for developmental plasticity: that is, to accommodate those features of the world which are only partially predictable. Thus if there are 'tracking devices whose own states can be altered and held in that altered condition for the same period as the features they are tracking' (Plotkin, 1995, p. 149), then the secondary heuristic can accommodate some of the environmental changes which the primary heuristic cannot track. For example, psychological mechanisms allow knowledge to be acquired on short-term stabilities in the environment, such as where food can be found or who is a friend: that is, features that the primary heuristic cannot, to use Plotkin's (1995) terminology, track because they occur too quickly for genetic evolution to take place.

However, any information acquired by the secondary heuristic is confined to an individual. The final level, the sociocultural level, incorporates a nongenetic channel of

communication. This level Plotkin (1995) calls the tertiary heuristic. In Plotkin and Odling-Smee's model, learning occupies a dual role for the tertiary heuristic - both the sender and the receiver gain information through the crucial and pivotal secondary heuristic. The secondary and tertiary heuristics share a storage site for information - the central nervous system. But Plotkin and Odling-Smee argue that while the tertiary heuristic requires the secondary heuristic, the secondary heuristic does not require the tertiary heuristic.

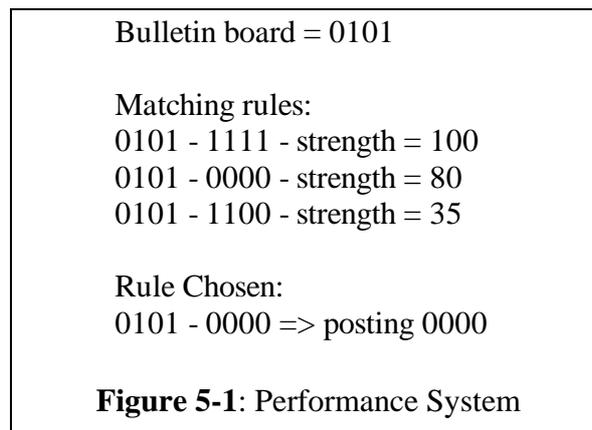
To summarise, one of the main aims of evolutionary theory is to explain adaptations. Adaptations come in a variety of forms: physiological, structural and behavioural. The neo-Darwinian genetical theory of natural selection and its mechanisms explain how *certain* adaptations can be formed. However, theorists going back to Darwin have postulated additional mechanisms which can produce adaptations and have causal power in the evolutionary process. For example, learning refers to a class of mechanisms whereby an individual can adapt within its own lifetime: that is, produce adaptations. Thus adaptations can arise from learning and then be behaviourally transmitted to other individuals (an example of the formation of adaptations in the absence of changes in gene frequencies). Plotkin and Odling-Smee's hierarchical evolutionary model postulates how this could happen. However, as Plotkin (1988) describes, a major problem with the evolutionary epistemological models is testing them. The computational experiments described in this chapter investigate the three levels described in this section: the biological, psychological and sociocultural.

5.1.2 Adaptive agents

The agents in the computational experiments described in this chapter were based on Holland's (1986, 1994, 1995) classifier system. Holland (1995) proposed the classifier system as a general representation for an adaptive agent where the capabilities of individual agents are described by a collection of stimulus-response rules. The classifier system was originally conceived as a production system in the spirit of a neural network (Waldrop, 1994). In developing the classifier system Holland was inspired by the genetical work of Fisher (1930) and the neurological work of Hebb (1949). He also made extensive use of economic metaphors such as free-market economies. In this section the basic classifier system is considered (Holland, 1986, 1994, 1995) along with Wilson's (1994) simplified

zeroth-level classifier system which is similar to the one used in the computational experiments described later in this chapter.

There are three components to a classifier system: the performance system, the credit assignment system and the rule discovery system. The performance system is a production system whose if-then rules are made up of ternary strings representing conditions and actions. These strings consist of 1's, 0's and #'s which stand for wild cards. The performance system has a 'bulletin board' where binary strings are posted. If the condition of



a rule string matches a message posted on the bulletin board, it gets the chance to post its action in that round. Conflict resolution is performed by the matching rules competing to post their messages. Each rule makes a bid on the basis of its properties - in the simplest case, this is proportional to the rule's strength. Then the system collects all the bids and chooses a set of winners by lottery, the highest probability of winning going to the highest bidder. Figure 5-1 gives a simple example where the condition 0000 is posted to the bulletin board, three rules match this condition and the second rule wins the lottery and has the chance to post its action 0000.

Rules can be activated concurrently. This parallelism has two advantages (Holland, 1994). First, using sets of rules together allows the system to make combinatorics work for it rather than against it. Imagine the old-style police sketch pads which divided the face into a number of parts and provided a number of instances of each part. If the face is divided into ten parts - for example, hair, eyes, nose - and there are ten variants per part, then these hundred building blocks can describe ten billion faces. In the same way the rules which make up the classifier system can be reused in a number of different combinations dependent

on the situation. For example, a classifier system with 100 different rules which is able to activate 5 rules concurrently can describe over 75 million situations.

The second advantage of concurrently activated rules is that using a number of rules to describe a particular situation allows experience to be transferred to novel situations. Seeing a red car at the side of the road with a flat tyre may be a novel situation, but if it activates the rules for car, red and tyre, then the appropriate (goal-satisfy) behaviour may be elicited.

Another feature of classifier systems is that the messages on the bulletin board can come from either external detectors or from other rules. This allows these systems to produce internal feedback concurrently with processing sensory input.

As the classifier system is described, it is incapable of credit assignment - assigning worth to the individual classifiers. Holland implemented a form of Hebbian reinforcement in which the agent adjusts the strength of its rules on the basis of their behaviour. Holland augmented the basic reinforcement mechanism so that as a rule wins the chance to post a message, it pays its bid to the rule that posted the message it matched. Thus, stage setting rules which produce useful behaviour are rewarded as the strength is propagated back through the system. This algorithm is called the bucket brigade algorithm. However, as the bucket brigade algorithm was not used I will not go into any more detail about it. The credit assignment algorithm used is described later in this section when discussing Wilson's zeroth-level classifier system.

The final component of the classifier system expands its learning capacity from just exploitation to include exploration. Rule discovery in classifier systems is produced by the action of a genetic algorithm. A genetic algorithm can be used in two ways (Wilcox, 1995): first, the Michigan-style where individual classifiers are coded as individual strings evolve. Rule selection is probabilistically based on strength. New rules are created and replace other low-strength rules. The second way is the Pittsburg-style where entire rule sets are coded as a single string and evolved. Note that the Michigan-style genetic algorithm allows adaptive change within the lifetime of the system, but this approach adds additional requirements to the system - for example, those requirements produced by need for rule replacement - that are still being investigated (Goldberg, 1989, Wilson, 1995). In this work both approaches were used simultaneously, the Michigan-style simulating learning in the lifetime of the agent and the Pittsburg-style providing adaptive change over generations.

While there have been a number of modifications to the basic classifier system (Goldberg, 1989; Wilson, 1995), as they are not incorporated in this work, they will not be discussed. However, Wilson's (1994) formulation of a zeroth-level classifier system (ZCS) will be described as it formed the basis of the implementations. The ZCS keeps much of Holland's original framework, but simplifies it to increase understandability and performance (Wilson, 1994).

The ZCS, like Holland's classifier system, has detectors, effectors and condition-action rules representing behaviours. The ZCS's rules, like Holland's rules, consisting of conditions and actions made out of ternary strings based on the binary alphabet, plus the 'don't care' symbol #. Associated with each rule is a scalar strength. A ZCS has no internal message list and is thus incapable of temporary memory. In the performance cycle of the ZCS, the condition of each rule is compared to the binary string representing the detector input. If the bit at every non-# position in the condition matches the corresponding bit in the detector string, the condition is satisfied and the classifier becomes a member of the current match set, M . Next, an action is selected from among those proposed by members of M . A ZCS employs a stochastic, rule-selection method: a roulette wheel with sectors sized according to the strengths of the members of M . Thus a particular action a is selected with a probability equal to the sum of the strengths of the rules in M which advocate that action, divided by the total strength of classifiers in M . Next, an action set is formed, consisting of all the members of M which advocate a . Finally a is sent to the effector interface, and the corresponding action is carried out in the environment.

The other details of a ZCS, for example, the reinforcement algorithm, differ from the algorithm used in this experiment and thus will not be described. An important point to note about this work and the experiments described in chapter 4, is that it uses previously developed algorithms because they provide the required functionality. This work is not aimed at investigating or developing these algorithms. The classifier system was used because it provided a means of having adaptive change over individual and generation time, but if, for example, the genetic algorithm operating within an individual's lifetime was replaced by another algorithm that provided the same functionality, this would have little bearing on the design-work aspect of these experiments. But the same sort of functionality with detailed differences could make a big difference to the quantitative results.

5.2 Method

This section is split into a number of subsections, starting with a discussion of the scenario and then describing each class of agents in turn. An overview of the programs used in these experiments is given in appendix 2 along with the location of the code on the University of Birmingham, School of Computer Science's computer network.

The abstract scenario used in these experiments consisted of ten stimuli which were presented twenty times to a population of agents. Each stimulus had a desired response. The stimulus-response rules consisted of 20-bit, randomly generated, binary strings. Ten bits for the stimulus and ten bits for the response. However, this may also be a disadvantage as organisms may be designed to learn specific types of patterns and rules, not arbitrary ones.

The quality of the agent's response was proportional to the number of bits which matched the desired response - this figure was added to the agent's fitness. The advantage of using such an abstract scenario was that it prevented the experimenter from programming anything into the agent which assisted its learning.

In these experiments a constant population size of 100 agents was used. Each agent had a genotype and a phenotype: initially, the same list of 20 condition-action rules. Each generation lasted 200 rounds, thus providing 20 presentations of each stimulus.

5.2.1 Darwinian creatures

Darwinian creatures consisted of just the performance system of a classifier system. In the performance system only a single rule could post its message; internal messages were not deemed necessary for the scenario and were prevented. If multiple rules matched a stimulus, then one rule was picked at random. Initially, the contents of the agent's genotype and phenotype rule sets were identical.

The agents started with 20 randomly generated rules. A rule was made up of a ten bit stimulus string and a ten bit response string. These strings consisted of 1's, 0's and #'s - which stood for wild cards. Thus there were 3^{20} possible rules.

After 200 rounds, a new generation was created from the last generation using the following algorithms. A tournament selection algorithm was used to pick the parents (fitter agents):

- 1 pick 10 members at random and with replacement from the current population
- 2 return the strongest member (if there is a tie, then randomly select from those that came first)

There was a second form of this algorithm which returned the weakest agent. This was used to select a rule to be replaced in a fixed population of rules within a Skinnerian creature. The following algorithm was used to create a new population:

repeat 100 times

- 1 pick parent 1 using tournament selection
- 2 pick parent 2 using tournament selection
- 3 recombine the parents' rule sets and mutate the resulting rule set creating a new agent

The species-level crossover was carried out by taking the parents' rule sets and picking a random point between one and the length of a rule set. The new rule set was made up of the rules of one agent up to the random point, combined with the rules of the other agent after that point. Mutation was carried out by randomly flipping a bit within a rule. Each rule had a 0.1 probability of a single bit being mutated.

5.2.2 Skinnerian creatures

The Mark I Skinnerian creature was implemented using the basic Darwinian creature and adding a credit assignment system. At the beginning of each generation every rule started with a strength of one and after a fitness evaluation a number between zero and one was assigned to the strength - this figure was proportional to the number of bits matching the desired response. As each rule had an initial start strength of one, the agents were forced to try each rule at least once, assessing its actual value, and subsequently using the highest strength rule.

The Mark II Skinnerian creature was implemented using a Mark I Skinnerian creature and incorporating a rule discovery system. Every 100 rounds a genetic algorithm was run replacing two 'weak' rules with two new rules. This internal genetic algorithm worked

identically to the external, population-level, genetic algorithm with rules being substituted for agents: that is, tournament selection picked rules within an agent, not agents from the population, and then rule-specific crossover and mutation operators were applied.

With the operation of the internal genetic algorithm, however, there is the additional complication of adding the new rules to the rule set without disturbing its performance. This was done using the version of the tournament selection algorithm which returned the weakest rule. The rule returned by the tournament selection algorithm was replaced by the new rule. Note that this replacement only occurred in the phenotype rule set, not the genotype rule set.

5.2.3 Dawkinsian creatures

The Dawkinsian creature was implemented using the Mark II Skinnerian creature and adding the capability of imitating other agents. The old generation of agents was run concurrently with the new generation. An individual in the new generation could, with a probability P , imitate a member of the old generation. As the old generation had already tried out all its behaviours it was using its best rules, and it was no longer able to improve its performance.

Imitation was carried out by copying a rule producing the observed behaviour with a possible mutation into the imitator's set of behaviours. The obvious simplification here is that copying rules is not the same thing as imitating behaviours. This simplification, that was adopted due to time restriction on this work, avoided the problem of rule inference. An agent from the old generation was picked using the tournament selection algorithm, then the observed rule (behaviour) was copied with a possible mutation over a weak rule using the second tournament selection algorithm (the tournament selection algorithm that returns the weakest member).

5.2.4 Vygotskian creature

The Vygotskian creature was the same as the Dawkinsian creature except that it had the additional capability of instructing other agents. Instruction was implemented by the agents of the old generation instructing, with a probability of P , a randomly chosen agent of the new generation. The transmitted behaviour was chosen using the tournament selection algorithm, there was the possibility of mutation, and the to-be-replaced rule was picked using the weak-

rule returning tournament selection algorithm. Using the tournament selection algorithm to choose the to-be-transmitted rule simulated an agent demonstrating a useful rule that was not elicited by context.

5.3 Results

These experiments produced two types of results: first, qualitative results relating to observations made while carrying out the necessary design work to implement the trajectory of agents; second, quantitative results obtained by measuring the rate of adaptive change of each kind of agents. In the discussion I will describe the results of the design work, here I will give a graphical representation of the rates of adaptive change in the different agents.

The points on the graphs represent the mean population fitnesses at the end of each generation. Thus at the end of the first generation the Dawkinsian and Vygotskian agents are on average significantly fitter than the other kinds of agents.

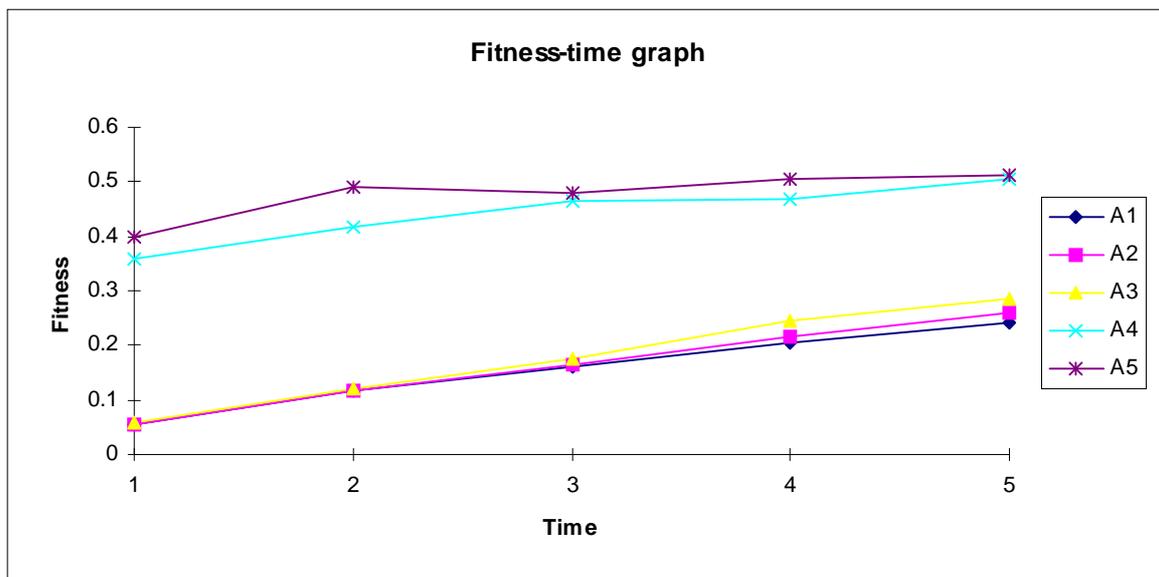


Figure 5-2: The five different kinds of agent are: A1 = Darwinian agent; A2 = Mark I Skinnerian agent; A3 = Mark II Skinnerian agent; A4 = Dawkinsian agent; A5 = Vygotskian agent.

5.4 Discussion

There are a number of deliberate simplifications in these experiments, for example, the agent's architecture is purely reactive, the agent is behaving as if it had what ethologists call

‘fixed action patterns’. However, for the purpose of modelling creatures with a greater degree of non-instinctive reasoning capacity, an architecture would have to incorporate additional sorts of control. Sloman (1997) postulates that the human information-processing control system has three rather different sorts of control: a reactive subsystem, which can itself include many different types of control with different degrees of sophistication, a deliberative subsystem and a meta-management subsystem. The agents described here have only a reactive subsystem and are incapable of deliberation, that is assembling new combinations of actions to cope with novel contexts, or of meta-management - self-monitoring and self-assessment of deliberation. Using agents with only a reactive subsystem was an intentional simplification in order to make the experiments feasible, not a specific constraint of the classifier-based implementation: Donnart and Meyer (1994), for example, implemented an agent architecture with both reactive and deliberative subsystems using hierarchically-organised classifier systems.

Another simplification in these experiments is the one-to-one, genotype-phenotype mapping. By using such a mapping, the mechanisms for increasing the rate of adaptive change discussed in chapter 3 and in the first half of chapter 4 were ignored. However, this mapping allowed a direct comparison of genetic and cultural evolution.

Other experimental simplifications include: the issue of evolving the Skinnerian agent’s reinforcement strategies (see section 5.4.2); the issue of how the agents transmit information about rules - the medium; and the constraints to information transmission imposed by the different media.

In the rest of this section I will discuss each class of agents in turn, considering their requirements, their costs and benefits, and their role in the process of adaptive change.

5.4.1 Darwinian creatures

The requirements of Darwinian creatures and their role in the process of adaptive change have already been discussed in chapters 3 and 4 and thus will not be discussed again here. In those two chapters a number of mechanisms were discussed which augment the process of adaptive change by reducing or removing the risk of evolving systems getting caught in local fitness optima. However, in the light of such augmentations to the Darwinian system we can ask why adaptive change is not more prevalent in the real world and why the fossil record

shows long periods of stasis? In this section I give an overview of stasis based on Williams' (1992, chapters 6 & 9) account before considering some additional explanations that emphasise the central theme of this chapter: the role of behaviour in evolution. In other words, this section helps build up the argument, supported by the results of the work on Skinnerian, Dawkinsian and Vygotskian creatures, for a broader theory of Darwinian evolution than the genetical theory of natural selection. Note, however, that stasis is only relevant to this thesis in so far as it highlights gaps in the explanatory power of the genetical of natural selection which are explained by a theory of evolution which incorporates cultural evolution.

Before considering the different explanations I want to emphasise that I do not believe that there is a single cause of stasis, nor should the apparent simplicity of the phenomenon require a simple explanation (Cohen and Stewart, 1995). In other words, Occam's razor, parsimony, or economy of explanation should not automatically lead us to seek a single mechanism to explain stasis or avoid complex explanations.

The first explanation for stasis, that is imperfections in the geological record, is not really an explanation of stasis at all, but an attempt to remove the need for an explanation by denying the phenomenon. Darwin (1859, chapter 9) went into great depth describing the problems that abound in the fossil record like the intermittence of geological formations and the sudden appearance of groups of species. However, enough evidence has now been collected to suggest that stasis is a real phenomenon and does need an explanation (Williams, 1992).

The next most common explanation of stasis is on the basis of historicity: phylogenetic, developmental and genetic constraints. Phylogenetic constraints refer to the fact that natural selection never designs new machinery to cope with new problems, but alters already existing machinery by slow alteration of the machinery's parameters (Williams, 1992, p. 76). Developmental constraints are merely a special kind of phylogenetic constraint (Williams, 1992). The role of developmental constraints, constrained embryologies, in facilitating adaptive change was described in section 4.2.2, however, as the names suggests, a constrained embryology can only produce a subset of possible phenotypes.

Another factor which can play a role in stasis is pleiotropy. The beneficial role of pleiotropy of forming gene nets was discussed in section 4.2. However, pleiotropy can also

lead to constraints. If different phenotypic features are coded for by the same piece of the genotype, then a beneficial change to one feature may be detrimental to another.

As Kauffman's (1993, 1995, 1997) work demonstrates, certain types of structure are quite stable, with small changes having little or no effect (see section 4.2.1). This robustness of structure can result, for example, from genetic assimilation (see section 4.2) where developmental programs have been selected that produce the same phenotypes in different environments. Thus systems with phenotypic plasticity - for example, via development or learning - can adjust to changes in the environment without any *further* genotypic changes. In other words, as the environment changes the structure stays the same.

One way a structure can stay the same in a changing environment is, if the environment permits, by changing its niche. In other words, if the environment changes, agents can either change to a different environment which is suitable to the old design or adapt to the new environment. This phenomenon was previously referred to as 'niche selection' (see section 3.3.3); Eldridge (1995) calls the same phenomenon 'habitat tracking'. If a suitable niche is present, then changing the niche will probably be quicker and more economical than anatomically adapting to the new niche.

Sexual selection, like niche selection, could also play a role in producing stasis. We discussed Todd's (1996) computational experiments on evolving populations of sexually reproducing individuals in section 3.3.3. Those experiments demonstrate that constraints on mate recognition can affect the rate of adaptive change. Large changes in those features which are used in mate recognition can make it difficult for the individual to find a mate. In other words, a beneficial adaptation may not be incorporated into a population if it produces an individual unrecognisable to other members of its own species. Todd (1996) found that learnt mate-selection strategies can lead to more rapid adaptive change (and speciation) because they allow a population to track changes in its members.

Another feature of natural selection which could create stasis is that superfluous ability is not rewarded (Humphreys, 1976). For example, while it may be possible to design a cheetah that could run twice as fast, a cost-benefit analysis compared to a normal cheetah would probably lead the latter to be evaluated as better as the cost of the increased speed would probably not be paid for by the increase in prey caught. However, if the cheetah's food starts running faster, then the requirements change and a double speed cheetah may be

selected. The point is that natural selection can create or prevent change depending on the circumstances. This has been mathematically demonstrated by Maynard Smith (1976) and led to the formulation of the concept of 'evolutionarily stable strategies' (ESS): 'a strategy which, if most members of a population adopted it, cannot be bettered by an alternative strategy' (Dawkins, 1989b, p. 69). In the previous example of the cheetah, the single speed cheetah was stable in that context, if the context changed, then the strategy may no longer be stable.

Thus far we have considered a number of factors which can cause or facilitate stasis. I now want to introduce a theory by Popper (1986) to supplement these factors and because the theory emphasises the role of behaviour in evolution. Popper's theory distinguishes between external or environmental selection pressures and internal selection pressures resulting from the preferences of the organism. He described a simple model with two kinds of genes: anatomy or a-genes and behaviour or b-genes. He further subdivided b-genes into p-genes (controlling preferences) and s-genes (controlling skills).

Popper argues that it is changes to b-genes, specifically to the p-genes, which affect the course of evolution. For example, a change to the preferences could lead to different skills and potentially to different anatomies that are compatible with the new preferences and skills. Obviously, this sequence can be cyclical: for example, new anatomies may favour changes in preference, and so on.

However, I do not think that genetics is essential to Popper's theory because adaptations can be transmitted behaviourally. As suggested before, to avoid confusion the word 'gene' in the evolutionary sense should be replaced by the word 'replicator'. This terminological change also has the effect of generalising Popper's theory to nongenetic systems. Moreover, questions can then be asked about the relative difficulties of transmitting a-replicators and b-replicators in different heredity systems. For example, is it easier to transmit a-replicators genetically and b-replicators behaviourally? Behaviour requires an appropriate anatomy, by allowing the anatomy to interpret a stored information structure controlling behaviours, you separate the transmission of anatomy and the transmission of behaviour giving b-replicators far more flexibility.

The evolution of b-replicators would be hard to identify in the fossil record and may result in the mistaken view that there is stasis. However, such changes could create new

niches and stimulate the evolution of new forms independent of changes in the external environment. (The spontaneous eruption of new forms independent of changes in the external environment could also be a consequence of genetic drift operating within neutral networks.) What is important is that in the face of changing niches, an agent could adapt *behaviourally* rather than anatomically, which could result in the appearance of stasis.

To sum up, the main point of discussing stasis was to consider the constraints acting on genetic evolution and to introduce the possibility of cultural evolution. In this section, I have discussed a number of factors which can contribute to stasis. The final part of this discussion outlined how natural selection of behaviours could adapt an agent even if anatomical adaptive change was inhibited by, for example, phylogenetic, developmental or genetic constraints. The behaviour could be genetically determined, but this is not necessary as the behaviour could be psychologically generated and transmitted behaviourally. The creatures discussed next demonstrate ways of generating and transmitting behaviours through nongenetic mechanisms. Cultural evolution can be regarded as a way around some of the constraints described in this section, as well as providing the means to increase the rate of adaptive change. Thus it may be more cost-effective for evolution to add a new information channel than to modify the old one. Even if the initial cost of moving to a behaviour interpreter for information structures is high, the additional flexibility could produce a significant benefit

5.4.2 Skinnerian creatures

A Mark I Skinnerian creature requires a set of different behaviours and a mechanism for exploiting goal-satisfying behaviours. What is important is the quality of the reinforcers: that is, whether they reinforce the correct, niche-appropriate behaviours. The standard biological answer to the question of how to assess the reinforcers' quality is through reproductive success - those creatures whose behaviours favour their survival and reproduction pass on their genes, including genetically coded reinforcers, to future generations. The selection of Popper's preferences (p-replicators) demonstrates how reinforcers could be selected (see section 5.4.1).

In the computational experiments described in this chapter the agents had the correct reinforcers coded into them. The reinforcers were not evolved because these experiments

were aimed at testing the rates of adaptive change of different kinds of agents and not their ease of evolution.

Another requirement of the Mark I Skinnerian creature was a lifetime long enough to test and evaluate their behaviours. There is a relationship between the number of trials needed to test a set of behaviours and the number of behaviours an agent possessed: that is, the more behaviours an agent had, the more trials were required to accurately assess them. This relationship is between the number of different behaviours and a particular reinforcer; however, there is also a similar relationship between the number of separate reinforcers and the length of an agent's lifetime. Each reinforced behaviour requires enough trials to determine the strong behaviours and let their subsequent use influence the agent's fitness.

The Mark II Skinnerian agent requires mechanisms for generating new behaviours and for replacing old behaviours. In these experiments a genetic algorithm was used for rule discovery. A benefit of using a genetic algorithm was that it allowed the advantages of selective trial and error discussed in chapters 3 and 4. Thus on the basis of previous trials, if certain behaviours have been favoured and the fitness landscape is correlated (i.e. similar behaviours have similar fitness), then the better behaviours can be used to guide the search for new behaviours.

A second requirement of a behaviour-replacement mechanism is created by the finite nature of the agents, in this case, the finite set of behaviours. The problem is identifying the behaviours which can be replaced without reducing the overall performance of the agent. There are two means of addressing this issue: first, to have a mechanism that identifies a weak behaviour that can be replaced and replaces it; second, to have a large set of behaviours with redundancy so that random replacement will have little or no effect. The first mechanism, selective replacement, was used in these experiments.

A number of problems occurred with relation to the internal genetic algorithm. First, there was a problem with the genetic algorithm picking parent rules of two totally different kinds (relating to different reinforcers) and producing low fitness offspring. This problem is analogous to an elephant trying to mate with a mouse. Two approaches to deal with this problem were explored: 1) an asexual genetic algorithm was used and 2) sexual selection was implemented.

First, the internal genetic algorithm was modified to implement asexual reproduction. In this case a fit rule was selected, copied, mutated with a probability P , and used to replace a weak rule. This solution to the mouse-elephant problem ignored the fact that there were different kinds of behaviour relating to different reinforcers and thus certain behaviours proliferated at a cost to the overall performance of the agent.

A second approach to this problem that was tried was to introduce sexual (mate) selection. This was implemented by the genetic algorithm picking a mate for a particular rule on the basis of similarity (number of bits in common) multiplied by strength. Sexual selection, however, did not improve the efficiency of the genetic algorithm and this problem was flagged for further work.

Skinnerian creatures have a number of benefits. For example, in these computational experiments, reinforcement learning meant that agents used their 'best' behaviours, that is, the ones that made the biggest contribution to their fitness. Thus the fitness estimates of the Skinnerian creatures were more sensitive than those of the Darwinian creatures. For example, if a Darwinian creature had two matching behaviours - one worth nine (hypothetical) fitness points and the other worth one fitness point - then it would use them equally, averaging five fitness points. Another Darwinian creature with one behaviour worth six fitness points would end up having a higher probability of transmitting its behaviour to the next generation. Skinnerian creatures reduce the dilution effect of bad behaviours by testing all their behaviours and then using only the strongest one.

Skinnerian creatures can also benefit by the process of genetic assimilation often called the Baldwin effect (Baldwin, 1896; Lloyd Morgan, 1896; Simpson, 1953; Bateson, 1963): a mechanism whereby learning can guide and augment genetic evolution (see section 4.2). Hinton and Nowlan (1987) illustrated genetic assimilation by simulating a simple interaction between learning and evolution in which a 20-connection neural network is at least partially specified by a 20-gene chromosome. The unspecified connections are left for the neural network to learn. It is assumed the neural network can implicitly recognise the desired state when it reaches it. A population of 1000 neural networks was left for 1000 trials to randomly try out different connections until the desired network was found. The next generation was created using a genetic algorithm. The parents were picked probabilistically, with those neural networks that learnt the problem the quickest being the most likely to be picked - their

probability of becoming a parent was proportional to $1 + 19n/1000$, where n was the number of trials after the correct setting had been learnt.

Because different genotypes specified different initial connections, recombination led to changes in the initial neural network configuration. As the fitter agents were the most likely to produce the next generation, their initial configurations were most often recombined leading to an increase in the number of initially correct connections and a reduction in the learning space. It took Hinton and Nowlan's model about 20 generations, with 1000 individuals per generation and 1000 trials per generation, to produce the correct response.

Maynard Smith (1987) considers to what extent learning had accelerated evolution in this model. In the absence of learning there are 10^6 genotypes, of which one has a fitness of 20 and all the rest a fitness of 1. 'An asexual population, without learning, would try out more than 10^6 individuals before solving the problem, compared to 20,000 for the simulated sexual population with learning' (Maynard Smith, 1987, p. 457). Maynard Smith (1987) argues that the effect of learning is to alter the search space in which evolution operates, surrounding the optimum by a slope that natural selection can climb.

As well as the benefits of genetic assimilation and the advantage of being able to use the best behaviours, there are the advantages of adapting to changes that occur too quickly for genetic evolution to track and the possibility that learning a function may be easier than evolving it (Johnston, 1982). First, Skinnerian creatures can adapt to changes that occur *within* their lifetimes. Darwinian creatures require previous knowledge of such changes or fortuitous accidental properties to cope with such changes. Second, because genetic evolution influences development, it may be difficult to influence the relevant phenotypic features through this indirect means. However, learning allows the acquisition of phenotypic features, and provides additional phenotypic variability for natural selection to operate on, even in cases where genotypic variability is minimal.

The mechanisms of Skinnerian creatures incur a number of costs. The Mark I Skinnerian creatures face an initial, potentially dangerous, testing phase. For example, an agent with ten behaviours, only one of which is good, will have a relatively low fitness after ten trials. However, the agents as implemented in these experiments will in a 100 trials use the strongest behaviour 91 times and the fitness calculation will be less obscured by the nine other poorer behaviours.

A problem that was initially encountered in these experiments was that the agents were implemented in such a way that the rule set was both the genotype and the phenotype. The position of a rule in the rule list was important to the genetic algorithm's action. Thus, initially, Skinnerian creatures were capable of transmitting acquired characteristics - Lamarckian evolution - because new rules were created in the lifetime of the agent and written over other rules and this modified rule set was used in the generation of offspring. However, the action of writing new rules over weak rules affected the crossover operation by disrupting building blocks (schemata). In other words, the operation of the Pittsburgh-style genetic algorithm is dependant on the order of the rules in the rule list, and the rule-replacement algorithm disrupted this order. This led to a catastrophic loss of performance, in fact, no adaptive change was observed.

Hinton and Nowlan (1987, pp. 450-451) describe a severe computation difficulty with Lamarckian evolution: 'To know how to change the genotype in order to create the required characteristics in the phenotype it is necessary to invert the forward function, that maps from genotypes, via the processes of development and learning, to adapted phenotypes. This is generally a very complicated, non-linear, stochastic function, and so it is very hard to compute how to change the genes to achieve desired changes in the phenotypes even if those desired changes are known.' However, as these experiments show, this problem does not only arise in agents with complex genotype-phenotype mappings. Simple one-to-one mappings suffer schema disruption without computational mechanisms for assigning appropriate insertion points for replacement rules. Thus Lamarckian evolution proved incompatible with the agents implemented here without the addition of extra (computationally expensive) mechanisms.

Johnston (1982) identifies six selective costs of learning: 1) delayed reproductive effort; 2) increased juvenile vulnerability; 3) increased parental investment; 4) greater complexity of the central nervous system; 5) greater complexity of genotype; 6) developmental fallibility. The cost of a delayed reproductive effort, mentioned earlier in this section, is related to the fact that there is an initial testing phase. Juvenile vulnerability can be reduced by parental care at a cost to the parents. Agents with a set of behaviours and the capability to learn require a more complex central nervous system than those without the capability. To code for a more complex central nervous system requires a more complex

genotype which in turn requires higher fidelity copying mechanisms. Finally, the very fact that behaviours are unspecified may result in agents learning maladaptive responses and being selected against.

Johnston (1982) stresses the point that while there are benefits for having the capability to learn, there are also a number of costs and if learning ability is to be selected, the benefits must outweigh the costs. However, the costs of different mechanisms need not be additive. It is unlikely, for instance, that delayed reproduction, juvenile vulnerability and parental care have additive costs (Johnston, 1982). Furthermore, if the benefits of one mechanism outweigh the costs and this mechanism is selected, then this may reduce the cost for other mechanisms, for example, the selection of better perceptuomotor skills that led to an increased central nervous system complexity would pay some of the cost incurred by learning (Johnston, 1982).

The cost-benefit analysis for different adaptations is relative to the other adaptations present. This means that learning may be beneficial at one point, and then at a later stage genetic assimilation will remove the need to pay its costs. For example, Mayley's (1996) computational experiments showed that while a learning ability is initially favoured, genetic assimilation will mean that subsequent agents that do not have to pay the cost of learning will be selected.

Another selection pressure for genetic assimilation is implicated by the false dichotomy of an agent having either instinctive abilities or a clean slate and a set of adaptive abilities (Todd, 1996). Agents with a better initial starting point (instinctive abilities) and the ability to adapt will have a selective advantage over purely instinctual agents. Hinton and Nowlan's (1987) experiments showed how genetic assimilation via the action of recombination can improve the quality of an agent's start in life.

The issue of genetic assimilation discussed in this section relates to the nature-nurture debate, that is, the difference between innate and acquired features. Mayr (1974) described another way of discussing this dichotomy using the concept of a genetic program. A genetic program which does not allow appreciable modifications during the process of translation into the phenotype is called a closed program - closed because nothing can be inserted through experience. A genetic program which allows for additional input during the lifetime of its owner is called an open program.

Darwinian creatures have fixed phenotypes - closed programs. Skinnerian creatures have a certain degree of phenotypic plasticity - open programs. If the features of the environment which are applicable to the genetic program are stable enough and the open programs are subjected to the costs described in this section, and the cost of storage and transmission of genetic information are not too high, then there will be a selective advantage for agents with closed programs. However, as the Hinton and Nowlan (1987) showed, open programs can guide evolution and result in a faster rate of adaptive change.

The phenotypic plasticity of the Mark I Skinnerian creature was limited by the size of the initial rule set. In the Mark II Skinnerian creature a genetic algorithm allowed the agent to search the space of possible behaviours. However, because of the time needed to initially evaluate the behaviours, the genetic algorithm was activated after 100 rounds: halfway through the agent's lifetime. The longer the agent lived, the more opportunity it would have to benefit from the rule discovery system. The genetic algorithm had only a minor effect in these experiments because of the short lifetime of the agents (see Figure 5-2).

Finally, the following problem with Mark II Skinnerian creatures was observed in the computational experiments: new rules that were discovered in the agent's lifetime could affect the overall fitness calculation of the agent, but they could not be transmitted to the next generation. Learning could produce phenotypically strong agents that were genotypically weak. Thus while learning could guide evolution, it could also fail to direct the evolution of agents not capable of passing on their fitness-enhancing, learnt features.

5.4.3 Dawkinsian and Vygotskian creatures

The first type of agent, the Dawkinsian creature, has the capability of imitation: that is, the ability to copy another agent's behaviour. A central requirement of imitation is something to imitate. Thus if an agent is capable of imitating, then the question is who or what should the agent imitate? Consider the following evolutionary sequence. The point of describing this sequence is to consider the opportunities available to agents in different niches, *not* to provide a just-so story for the evolution of intelligence.

First, consider niches in which parental care is available. In the last section parental care was discussed in relation to reducing infant vulnerability in agents capable of learning. Parental care incurs a cost to the parents while benefiting the offspring. However, parental

care satisfies a necessary condition of imitation: something to imitate, which is not to say parental care is logically sufficient for Dawkinsian creatures to evolve. Where parental care is provided, there are two reasons why it might be a good idea to imitate your parents: first, the parents' behaviour has been successful enough to lead to reproduction; second, the observed behaviour will be species appropriate as the parents are of the same species. The main point is that there are reasons to think that imitation would be a selective advantage in niches where parental care is available.

Furthermore, parental care can create family groups. Groups form a necessary condition for what Bonner (1980) calls the 'wolf pack principle': that is, where groups allow actions not possible to the individuals alone (in the case of the wolf this refers to hunting large prey). Note that the wolf pack principle is task-dependent, for many tasks an individual may be better off alone, for example, an individual will not hunt in a group unless it finds more food than hunting alone. While family groups satisfy a necessary condition (a group) under which the wolf pack principle can be exploited, as social insects demonstrate, social living alone is not sufficient for the evolution of intelligence. There are, in fact, a number of advantages of group living (cf. Krebs and Davies, 1987, chapter 6), but for the purposes of this discussion we will focus just on the wolf pack principle.

Humphreys (1976) postulates that it is social interactions between certain kinds of agents that created a requirement for increased intelligence. Thus while the cost of increased intelligence may have already been partly paid for by other functions (e.g. learning), Humphreys postulates that it is social living, particularly social negotiations, which produces the requirement for increased intelligence.

However, there is a second possibility: that is, social living does not just require increased intelligence, but it creates it. By this I mean that imitating agents in a social group can satisfy the logical conditions which produce Darwinian evolution by natural selection. In such cases imitation forms part of a second heredity system which brings about cumulative adaptive change across groups of individuals separated in space and time.

The mechanisms of imitation also compensate for the learnt adaptation problem found in Skinnerian creatures: that is, that adaptations learnt over an agent's lifetime cannot be genetically inherited by their offspring. With imitation, learnt adaptations can be transmitted to the offspring by a nongenetic heredity system. As well as leading to adaptive change

which is cumulative over species time, imitation also produces a social equivalent of recombination which is not constrained to a single instance of mating, but can operate between any individuals at any time (Bonner, 1980; Williams 1992). Thus unlike sexual reproduction where two individuals can share genetic information at one time (conception), imitation, in theory, allows any number of individuals to share information as often as possible.

Another feature of cultural evolution is that it may suffer less than genetic evolution from the problems of features being associated in such a way that changes to one feature create changes in another. In genetic evolution this occurs partly because there is a selection pressure for shorter genotypes. Obviously this selection pressure is not directly applicable to cultural evolution. It may be the case that new behaviours can be added in cultural evolution without disrupting the old behaviours and that behaviour can be more readily changed without changing other behaviours.

Imitation can also provide an advantage when an agent learns sequences of behaviours. As a behavioural sequence increases in length, the number of possible sequences rises exponentially. There are two approaches agents can take to reducing this problem: first, they can acquire parts of sequences (modules) which can be subsequently used as building blocks; second, they can acquire information about the ordering of sequences. The imitative mechanism allows agents to take advantage of both approaches.

Thus far we have considered some of the requirements and benefits of imitation, but what about its costs? Plotkin and Odling-Smee's (1981) hierarchical evolutionary model (see section 5.1.1) emphasised that the different levels of the hierarchy were nested in the levels below, for example, as the brain is coded in the genotype, the psychological level is nested in the biological level. Furthermore, the sociocultural level is nested in the psychological level as they share some of the same mechanisms. As Johnston (1982, pp. 342-3) pointed out, 'interaction among the various costs and benefits of learning is unlikely to be linearly additive, since there will most likely be reciprocal interactions among the various selection pressures that act on an evolving population... It should be noted that the evolution of one type of learning ability may make the evolution of others less expensive... then the evolution of others will not incur this selective cost and there may be a 'snow ball' effect, permitting the relatively rapid evolution of several learning abilities'. In other words, if the costs have

already been paid (or partly paid), then the additional mechanisms incur no (or negligible) costs while providing additional benefits. In the case of the sociocultural level, if the costs of learning have been paid by the psychological level, then the sociocultural level incurs negligible additional costs - in fact its benefits may help pay the costs of additional brain power.

However, imitation has a big constraint: you can only imitate what you can observe. Infrequent but essential behaviours, for example, can be missed. Hence there is an advantage of having another knowledge transmission mechanism - instruction. Instruction, like imitation, refers to a class of mechanisms, not a single mechanism. While agents with imitation can only imitate what they observe, agents with instruction can demonstrate behaviours which are not naturally elicited. For example, play stalking in many predators gives the imitator both something to observe and hunting practice. Thus instruction can be regarded as a mechanism which adds selectivity to the process of adaptive change.

Vygotskian creatures are named after the Russian psychologist Vygotsky who postulated the concept of the zone of proximal development (ZPD). The ZPD is the distance between a child's actual developmental level and his or her potential level of development under the guidance of adults or in collaboration with more competent peers (Smith and Cowie, 1991). This relates to the notion of designs developing through a sequence of niches (see section 2.5), for example, an embryo developing in a womb until it is capable of surviving in the external world with the help of the parents and so on. Thus instruction allows an agent to provide a suitable behavioural niche which facilitates another agent's behavioural development.

To conclude, imitation and instruction provide a nongenetic information transmission channel. This has a number of advantages: first, adaptations learnt within the lifetime of an agent can be passed on to the next generation; second, information can be transmitted to any individual (not just to offspring); third, information can be imparted many times (not only at the point of conception); fourth, some adaptations can be transmitted independently of other features.

Cultural evolution can be much faster than genetic evolution (Bonner, 1980). Genetic evolution is constrained by generation turnover; cultural evolution, on the other hand, allows rapid change within the lifetime of the individual. Because 'evolutionarily stable strategies'

are dependent on the other strategies present and because agents have limited capacities to predict what other agents are going to do, a dynamic environment is expected. The effect of rapid change in the cultural domain will prevent or reduce the amount of genetic assimilation. Genetic evolution cannot track changes that occur above a certain frequency related to generation time (Plotkin, 1995) and hence the necessary adaptations have to be produced by cultural evolution.

Whereas sexually reproducing Darwinian creatures are limited to receiving information from two individuals once in their lifetime, Dawkinsian creatures can receive information from any individual on many occasions. This allows a kind of social recombination the results of which are clearly seen in Figure 5-2. As the graph shows, populations of Dawkinsian and Vygotskian agents have an initial advantage as they can adapt to a new situation more quickly than the other kinds of agents. Social recombination indicates a benefit which could help pay the cost of the expensive central nervous system as it allows the ability to adapt within the lifetime of an agent, and can also produce cultural evolution.

In one theory of human cultural learning (Tomasello *et al.*, 1993) three forms of learning are distinguished: imitative, instructive and collaborative. However, these experiments show that the first two forms of learning automatically produce a form of collaborative learning. The consequence of imitation and instruction is a kind of social recombination between groups of individuals distributed in space and time. As the Skinnerian creatures were implemented so that they always used their strongest behaviours, it was these behaviours that were on show to be imitated. In other words, ignoring deception, an agent can usually be expected to use its 'best' behaviours and so these behaviours are what is imitated. However, the imitation process is not exact: 'mutations' can arise. Thus the second heredity system satisfies Lewontin's three principles of phenotypic variation, differential fitness and heritable fitness. There is a phenotypic variation in behaviours, a differential fitness between different behaviours in relation to different goals, and fitness is heritable - there is a correlation between what is observed and what is tried.

Finally, other adaptations can augment the heredity process (Maynard Smith & Szathmari, 1995), for example, language, written communication and science. Language is seen here as a communication tool, written communication allows agents from different

places and different times to communicate with other agents (including themselves), and scientific papers are a formal written structure incorporating mathematical and logical formulations which creates a high fidelity cumulative knowledge base.

5.5 Summary and further work

These experiments describe a trajectory in design space made up of five different kinds of agents. These implementations allowed the comparison of agents with and without particular mechanisms: for example, with a reinforcement mechanism (Skinnerian creature) and without one (Darwinian creature). The results of these experiments can be split into two categories: qualitative design-work observations and quantitative measurements of the rates of adaptive change. The design-work involved identifying the requirements, exploring the mechanisms which satisfied those requirements, and examining the costs and benefits of the mechanisms. The quantitative side of the study aimed to measure and compare the rates of adaptive change between the different kinds of very simple agents. These measurements showed that there was a significant increase in the ability of Dawkinsian and Vygotskian creatures to adapt compared to the other creatures in this very simple world.

The main design-based observations were as follows: Darwinian creatures are a sufficient mechanism to produce adaptive change if time permits. However, time is often in limited supply. The mechanisms of Skinnerian creatures are capable of guiding evolution (the Baldwin effect) and allowing adaptive change within the lifetime of the agents. However, Skinnerian creatures have not got the mechanisms to pass on acquired adaptations to the next generation and the costs of the mechanisms in Skinnerian creatures produce a selection pressure for genetic assimilation. Dawkinsian creatures have mechanisms which allow the transmission of learnt characters. However, social agents are faced with the formidable task of interacting with each other and this can lead to rapid change which cannot be tracked by genetic evolution. Thus cultural evolution, while influenced by genetic evolution, becomes partly independent of it. The final kind of agent, the Vygotskian creature, demonstrates how selectivity can be incorporated into the process of cultural evolution by agents being able to select which of their behaviours are on show to be imitated.

Central to this thesis is the idea that Darwinian evolution by natural selection is an algorithmic process. If the logical conditions are satisfied, then adaptive change follows. I argue in this chapter that agents capable of imitation can satisfy the logical conditions required for Darwinian evolution. There is a way in which adaptive change in the cultural domain occurs independently of the agent's intelligence. In fact the agent's intelligence can be seen as a way of adding selectivity to the trials explored and hence brings us back to the concept of evolvability. To conclude this chapter, two additional kinds of agents are described that address issues ignored so far and a scheme is suggested for representing the different kinds of agents discussed in this thesis.

5.5.1 Further work: Popperian and Gregorian creatures

Dennett (1995, 1996) describes two kinds of agents which were not implemented: Popperian and Gregorian creatures. For the purpose of this discussion these two kinds of agent will be described as if they are independent of the previously described agents, however, it may be the case that these mechanisms may be required for some of the previous mechanisms to work. For example, learning by imitation may use Popperian mechanisms. Popperian creatures have an additional mechanism, an internal model, which allows *simulated* trial and error; Gregorian creatures can build and utilise artefacts. Further work could investigate these agents and compare them to the other agents described in this chapter.

Skinnerian creatures' capacity for trial and error is fine so long as the creature is not killed by one of its early errors (Dennett, 1995). A better system involves preselection among all the possible behaviours and actions before field testing them in the real world. Dennett (1995, 1996) names such creatures 'Popperian creatures' after Karl Popper who said that this type of design enhancement permitted our hypothesis to die in our stead.

A Popperian creature requires some sort of internal environment which is structured in such a way that the actions favoured there are the same or similar to the actions favoured in the external environment. The inner environment, in other words, must contain information about the outer environment and its regularities (Dennett, 1995).

Internal models as a mechanism for guiding behaviour have also been discussed by Holland (1995). Holland emphasises the role of models in both anticipation and prediction. Holland suggests that the basic move for constructing models involves eliminating details so

that selected patterns remain. In his analysis Holland distinguishes between two types of internal models: tacit and overt. The first type, the tacit internal model, prescribes an action under an implicit prediction of some desired future state, for example, a bacterium swimming up a glucose gradient implicitly predicting the presence of food. An overt internal model is used for the basis of explicit, but internal, exploration of alternatives, a process called lookahead.

Popperian creatures can have both tacit and overt internal models. Humans, for example, are Popperian creatures that with responses like vertigo, fleeing in terror and sexual attraction demonstrate the use of tacit models, whereas chess playing, for instance, demonstrates the use of overt models.

Experiments are used in the modelling process to both test and gather data (Magee, 1988). From observations a number of possible models can be used to describe a situation. Experiments can then be used to differentiate between models as well as gather more data. The abstract models generated by this process ignore unknown details when they are not relevant.

Science is a formalised model generating and testing process. However, the use of models in science has led to a realisation of their limitations. To simulate a real world situation, like a flapping wing, requires a whole artificial physics of the relevant phenomena and information of all the properties of the wing - hardness, brittleness, elasticity, bending and compression - and any emergent properties of the model that may arise - patterns of turbulence and the interaction of the multitude of different factors (Dawkins, 1996).

Thus simulations require a lot of information which may not be known and the process itself can be computationally very expensive. However, engineers also use models such as wind tunnels to try out their ideas, often after an extensive period of computational modelling. Popperian creatures can reasonably be expected to do the same, that is, to use the internal models of the world to pre-select possibilities before field testing. Thus a necessary requirement of Popperian creatures is a delay between sensing and acting. Without this capability it is not possible for an agent to test out different possibilities against an internal model, or a simplified external one.

Dennett's (1995, 1996) final kind of agent, named after Richard Gregory, is called a 'Gregorian creature'. Gregory described the concept of 'potential intelligence' which

identifies well-designed artefacts - tools - as endowers of intelligence (external potential intelligence). Tools require intelligence to use and confer intelligence on their user. However, using a tool does not necessarily require the intelligence needed to design it. For example, adaptations can be regarded as tools, but to use a brain obviously does not require the ability to be able to build one or even understand how one works. A related point is that improving a tool does not necessarily require the intelligence of being able to build the improved tool from scratch, for example, improving a computer program.

Tools need not be physical artefacts: in the sense used here a word is a tool. Words and other mind tools give a Gregorian creature an inner environment that permits it to construct even more subtle move generators and move testers. Thus Gregorian creatures can benefit from the experience of others by exploiting the wisdom in tools that others have 'invented', improved or transmitted.

Both Gregorian and Dawkinsian creatures show how nongenetic adaptive change can occur via the same process as genetic adaptive change: natural selection. However, while Dawkinsian creatures have mechanisms of imitation, Gregorian creatures do not need such mechanisms. A physical tool created by one Gregorian creature could subsequently be found by another agent who learns how to use it. These tools form part of the phenotype of the agent and produce what Dawkins (1982) calls an extended phenotype.

It is possible to imagine Skinnerian creatures using tools. As reinforcers specify the desired states and not the means to achieve them, using a tool as part of a behavioural sequence leading to a reward would reinforce the tool usage. However, in nature, tool use appears rare. It is interesting to ask why, if many animals are capable of using tools, they do not do so more often. Tool use forms another possible benefit of a complex nervous system which could help to pay its costs.

To conclude, Popperian creatures have a mechanism for exploring the consequences of different behaviours before they are used in the external world. Gregorian creatures can incorporate structures into their phenotypes. Both tool use and lookahead can increase the evolvability of an agent: that is, these different mechanisms can increase the fitness of an agent and lead to the selection of the underlying mechanisms (see chapter 4).

5.5.2 Darwin machines

The term ‘Darwin machine’ was coined by Calvin (1987) to refer to machines whose operation is based on variation and selection. Dennett (1986) argued that variation and selection, which he calls generate-and-test, are fundamental to all processes of invention. He writes that such claims ‘owe their plausibility to the fact that they are implications of an abstract principle whose “necessity” (such that it is) consists in this: we can know independently of empirical research in psychology that any adequate and complete psychological theory must exploit some version or other of the principle’ (Dennett, 1986, p. 71). However, Simon (1982) points out that generate-and-test is not an efficient or powerful process unless the generator is endowed with a high degree of selectivity which, he goes on to say, can be equated with some kind of feedback of information from the environment. In this thesis I have described a set of Darwin machines which differ in kind. The different mechanisms add selectivity to the variation produced and this can create the property of evolvability. In this section I will outline some of the Darwin machines discussed in this thesis.

In Holland’s (1994) analysis of adaptive systems he describes the ‘enumerative plan’ - a plan for searching for an optimal structure in a set of structures. The enumerative plan exhaustively tests a set of structures and is characterised by the fact that the order the structures are tested in, is unaffected by the outcome of previous tests. The plan generates a set of structures, tests them, keeps the best, generates some more, tests them and checks the best against the best ones it has stored. While this plan is effective, it is not efficient. The enumerative plan could be termed the zeroth-level Darwin machine and is a true example of completely *random* trial and error.

However, the evolutionary process is better characterised as *selective* or *constrained* trial and error. If niche satisfaction biases reproduction (differential reproduction) and reproduction produces forms that are correlated to the parents (heredity), then the next generation is constrained to a subset of the possible variants. With reproductive mechanisms that duplicate the parents this scheme is limited to the initial variation. However, when in the process of reproduction errors are made, a constrained search space can be explored by the next generation. Mutation functions to inject new variation into the population and acts as an

‘insurance policy’ in case selection removes variation (Holland, 1994). Reproduction with and without mutation form two different *kinds* of Darwin machines.

Holland’s (1995) Schema Theorem shows that recombination is a powerful mechanism for finding the above average building blocks that make up a genotype (see section 3.3.2). Thus the recombination mechanism can be incorporated into a design to produce another kind of Darwin machine.

Todd (1996) showed that agents with mate-preference learning can adapt faster than agents without the mechanism because they can track evolving population structures (see section 3.3.3). Thus mate preference learning is another mechanism that can be incorporated into a design to create another kind of Darwin machine.

One step of the design-based methodology was referred to as exploring a neighbourhood in design space (see section 2.3). There are a number of possible designs which can satisfy the requirements for adaptive change. In particular there is a class of designs called here the Darwin machines. Darwin machines can be implemented using the mechanisms discussed in this section as well as other mechanisms such as open developmental programs, learning, modular design, hierarchical organisation, imitation, instruction, and internal models. By specifying the different Darwin machines it is possible to compare them in the following manner: one machine with mechanism X against another machine without mechanism X. It is suggested that by classifying and describing the different Darwin machines in this manner it is possible to explore their requirements, costs and benefits in a systematic and precise way.

To conclude, at the start of this chapter the problems inherent in investigating cultural evolution were identified. The computational approach was offered as a solution as it did not suffer from a number of problems that affected the other approaches. A hierarchical model of the evolutionary process was described and a design-based study of it executed. The insights into the model gained by the design work and quantitative results obtained from comparing different kinds of agents was documented. Finally, two additional kinds of agents were outlined for future work and in this section a suggestion was made for formally specifying the set of architectures which satisfy the design called here the Darwin machine.

6. Summary

6.1 Overview

This chapter takes as its starting point a summary of the major issues addressed in this thesis (note that these points are not all meant to be novel). This is followed by a consideration of possible directions for future work and a summary of the main contributions made in this thesis.

6.2 The main points of this thesis

1. There are many different types of explanation. This thesis focuses on formal and functional explanations. Aristotle argued that the essential nature of a thing is its functional organisation. Thus as long as the relevant functional properties of a design can be abstracted, the functional organisation can be described independently of the implementation details. This leads to the conjecture that an explanation of evolution and the mind can be given independently of detailed knowledge of the biology in which these machines are implemented.
2. The phenomenon examined in this thesis is adaptive change. Analysing the concept of adaptation reveals a number of things: firstly, the concept is used to apply to both the process of adaptive change and to the products of that process; secondly, adaptations have relational (to the niche) and functional properties - in other words, adaptations are functional design features which satisfy niche requirements; and thirdly, adaptive change is either a process of searching through design space looking for designs which 'better' fit a particular niche or of changing to a more suitable niche. The conclusion of the conceptual analysis was that adaptation refers to the class of design features with relational and functional properties and adaptive change refers to the class of processes which can produce adaptations.
3. The analysis of design and niche properties demonstrated that adaptive processes are required in dynamic environments, where designs are incomplete, and in competitive environments. If the niche changes, then for the design to persist it requires either to be able to tolerate the change or change accordingly. The interaction of adaptive agents results in dynamic systems producing a Hegelian world of perpetual change where

complex situations contain conflicting elements which by their very nature are destabilising and produce change (Magee, 1988). Secondly, in nature, for example, many designs are unfinished: that is, they require certain areas of the design to be modified in relation to the niche. Furthermore, in a world with finite resources, there will be a competition between designs with the better designs being more successful. In all these cases adaptive processes can provide a selective advantage, and as these conditions appear prevalent in the real world, adaptive processes are necessary to the success and persistence of designs.

4. In chapters 3 and 4 the aim is to give a logical description of biological evolution: describing the functional organisation of the Darwinian process independent of its biological base. It is argued that the Darwinian process is a probabilistic, anytime algorithm. The logical conditions sufficient to bring about Darwinian evolution were examined (e.g. Lewontin, 1970).
5. A conjecture of this thesis is that characterisations of evolution like, ‘random (or blind) variation and non-random selection’ are seriously misleading. In a Darwinian process, variation is ‘blind’ in a rather uninteresting sense (e.g. random point mutations), what is more interesting and important is how the variation is non-random. The Darwinian process is not an example of Holland’s (1994) enumerative plan where subsequent trials are generated independently of the results of previous trials, but it is an example of Simon’s selective trial and error where previous trials inform (in the biological case via heredity) subsequent trials. Different mechanisms which can lead to selectivity in trials are examined in this thesis.
6. The genetic system is logically described and the insights of the computational studies into genetic algorithms are considered. However, a number of properties of natural populations are largely ignored in these computational models. This led to a consideration of the myopic property often ascribed to evolution and concluding that this limitation is neither as deep nor as problematic as it at first seems. For example, natural partitioning of populations in space and time, patch logic and neutral nets help stop populations getting caught in local optima.
7. In the first half of chapter 4, the role of development in evolution is considered, particularly how developmental programs can affect adaptive change. Development is

described as a conditionally-branching, hierarchical, control system. The program execution is conditional on factors internal and external to the agent. The developmental program is hierarchically organised both in Simon's (1982) modular 'box-within-box' sense (e.g. nested subroutines), and in Plotkin's (1995) control hierarchy sense of programs controlling and co-ordinating other programs which are being executed concurrently and asynchronously.

8. Simon (1982) mathematically demonstrated that hierarchical organisation can affect the rate of adaptive change and Koza's (1992, 1994, 1995) work on genetic programming seems to support this claim. These mathematical and computational results demonstrate how the representation of the developmental program can influence both what can evolve and how fast it can evolve.
9. A set of computational experiments are described in the second half of chapter 4. The experiments demonstrate two things: that evolvability can be selected and that evolvability-enhancing mechanisms are context-dependent (e.g. relying on population size and landscape ruggedness). Thus some of the mechanisms described in chapters 3 and 4 indicate how agents can vary in 'evolvability' and the computational experiments demonstrate how Darwinian evolution can select the adaptive mechanisms that produce the fastest rate of adaptive change (this is not to suggest that the mechanisms selected lead to the optimal rate of evolution).
10. Design work is described in chapter 5 which investigates the role of additional mechanisms (such as, reinforcement learning, imitation, and instruction) in the process of adaptive change. Another conjecture argued in this chapter is that computational experiments allow the testing of evolutionary models in a way which is unfeasible in the real world and have been acknowledged to be mathematically formidable (May, 1977).
11. The first class of agents examined were Darwinian creatures that had fixed phenotypes. A number of factors are discussed that may constrain adaptive change and produce stasis. Rather than a single mechanism being postulated to explain stasis, a collection of mechanisms relating to the previous discussion are described.
12. The second class of agents were Skinnerian creatures that are Darwinian creatures with additional mechanisms such as reinforcement learning. The phenotypic plasticity of Skinnerian creatures helps avoid some of the factors that produce stasis, for example,

allowing the generation and testing of phenotypic variations independently of genetic variation. However, learning is not without its costs a number of which are discussed. One problem with Skinnerian creatures is that fitness producing adaptations learnt within the lifetime of the agent are lost at each new generation because they cannot be genetically inherited.

13. The final two classes of agents are Dawkinsian creatures capable of imitation and Vygotskian creatures capable of instruction. These mechanisms allowed behavioural transmission of the adaptations learnt within the lifetime of an agent. Thus imitation and instruction create behavioural adaptive change which is cumulative over species time. Furthermore, cultural evolution differs from genetic evolution by: 1) allowing agents to pass information more than once per generation and between any number of individuals; 2) allowing agents to produce phenotypic variation throughout their lifetime thus adaptive change becomes independent of generation turnover. It is argued that these kinds of agents in a cultural setting satisfy the logical conditions for Darwinian evolution by natural selection.

14. Two additional types of agents were outlined: Popperian and Gregorian creatures (Dennett, 1995, 1996). Popperian agents possess an internal model of the world which allows them to pre-test possibilities before trying them out in the real world. Gregorian agents utilise tools and the intelligence implicit within the tool's design, for example Newton's model of the world can be used to test a hypothesis without the agent having to be able to have invented the model in the first place.

15. A final conjecture is that human intelligence is to a large degree a sociohistoric phenomenon. To build a single agent capable of reaching human performance unaided by the accumulation of cultural knowledge is an unnecessarily difficult task. This is not to say that building a robot that can take advantage of culture is itself a simple task.

6.3 Further work

The broad scope of this thesis means there is a plethora of further work that could be done. The design-based methodology used in this thesis has been borrowed from artificial intelligence and applied to artificial life. In describing this approach I outline an ontological

position. Further work could refine this metaphysical model and elucidate the roles of abstract machines and causation.

In the conceptual analysis of adaptation a relationship between epistemological questions and the process of adaptive change is suggested. Further work could be done to analysis the hypothesis that adaptations are knowledge, and adaptive change is a process of knowledge acquisition.

The requirements analysis could be further refined and tested. Biological cases could be investigated to see if, for example, there are situations where the requirements are not satisfied but organisms still manage to live there. From this study of actual niches the requirements could be refined and further requirements could be specified.

There is still a lot of work that could be done to test the hypothesis that evolvability can be selected. In this work only two possible mechanisms were investigated and only on a limited range of scenarios. Further work could investigate in what conditions different mechanisms are selected and how different combinations of mechanisms interact.

The design work done in this thesis incorporates the ‘broad but shallow’ simplification. The individual mechanisms had to be simplified to be able to implement broad designs. Further work could deepen the implementations of the mechanisms, such as learning, imitation and instruction. The process of deepening the implementations is itself further design work which will create a greater understanding of the implemented mechanisms.

In the course of the design work inadequacies in the specification and hence the theory were uncovered, for example, the rule (behaviour) replacement procedure within agents. The design-based approach could be reapplied, the requirements of the mechanism analysed and possible designs which satisfy these requirements explored.

Additional work could be done to broaden and precisely formulate the class of Darwin machines outlined in this thesis. Additional Darwin machines are outlined like the Popperian creatures and design work could be done to produce functional specifications of the mechanisms and to test that they have the capabilities attributed to them.

A number of claims are made about different mechanisms and how they could influence evolution. For example, it has been argued that different developmental programs can influence the rate of evolution (e.g. constrained embryologies). The developmental

process has an inherently algorithmic nature which makes it particularly susceptible to a computational approach. However, the majority of current work ignores this distinction and uses simple (often one-to-one) genotype-to-phenotype mappings (Ackley and Littman, 1994). Thus further work could examine the effect of different types of embryologies on the rate of evolution such as embryologies with modular and hierarchical structures and embryologies with redundancies in their coding that produce neutral networks and embryologies that are inherently social.

There were a number of other mechanisms which could benefit from design work and subsequent testing through computational experiments such as Dawkins' conjectures about the relationship between reproduction and embryonic development, the role of patches, neutral networks, and factors such as space, time, constrained interaction between agents, in the evolutionary process.

Finally, a number of conjectures were postulated in this thesis and could form the starting place for further work. For example, investigating a theory of speciation caused by the joint action of natural, sexual and niche selection. Popper's theory of a-genes and b-genes would make a very good starting point for a research program investigating the role of behaviour in evolution, such as the selection of reinforcers (p-genes) in Skinnerian agents.

To reiterate a point made in the introduction, the design-based approach combined with the broad-but-shallow simplification produces an eclectic literature survey where breadth is substituted for depth. The obvious direction for further work to take is to deepen the implementations and provide mathematical analyses of them. However, the argument that has been developed in this thesis is for examining the interaction between different mechanisms, so another possible direction for further work is to continue exploring design space via analysing the requirements and examining costs and benefits of different broad but shallow architectures. While neither approach is sufficient in itself, a combination of the two approaches is needed in research to produce a happy medium between exploitation and exploration (a distinction central to the contents of this thesis).

6.4 Contributions

- **Methodological contributions:** the problem of a lack of a formal methodology in the field of artificial life is identified. The design-based methodology, developed in artificial

intelligence, is offered as a solution. Support is added to the argument that taking a computational approach removes some of the problems inherent in experimental and mathematical approaches.

- **Terminological contributions:** problems with certain evolutionary terms are identified, for example, adaptation, heredity, gene and species. The concept of adaptation is analysed and potential confusions in the terms heredity and gene are examined. Darwins' original terms have become bound to genetical theory of natural selection and suggestions have been made for a general terminology. For example, Darwin (1859) emphasised the role of heredity in evolution and while this has become synonymous with the genetic system, Darwin only stated that 'like produced like'. A more carefully worked out concept of heredity is suggested where one structure informs another without mention of a particular implementation like genetics.
- **Conceptual contributions:** the problem of Darwinian evolution being conceptualised as random trial (blind variation or random variation) and error (selection) is identified. As a solution it has been suggested that Darwinian evolution is characterised as selective or constrained trial and error. This distinction is explored with reference to the concept of evolvability and has resulted in the suggestion of specifying a set of Darwin machines. A second conceptual problem that is identified is that there is, in general, a bias to attribute causal power to only physical things (see [~axs/public-html/misc/supervenience](#)). In an attempt to address this problem the possible roles of behaviour in evolution are elucidated.
- **Theoretical contributions:** the main problems examined are that of evolvability and combining genetic and cultural evolution. Evolution is examined from an algorithmic perspective and it is argued that genetic and cultural evolution are both instantiations of Darwinian evolution by natural selection. However, as well as the similarities between the two instantiations there are a number of differences which effect the functionality of the two systems such as the rate at which adaptive change can occur: that is, its evolvability. To support the claim that there is such thing as differential evolvability, a range of mechanisms are described which can in different circumstances increase the rate of adaptive change of their owners.

7. Appendix 1

A brief functional description of the programs used in the experiments described in the second half of chapter 4 is given in this section. The actual programs can be found on the University of Birmingham, School of Computer Science computer network in the directory ‘~cxc/Public/Code’. The programs in this section took approximately six months to develop and due to their computationally intensive nature, data collection took a further month with ten Dec Alphas being used in parallel.

The following functional specification will be split into two parts: in the first section the NK landscape programs are described, in the second section the programs which run the experiments are described. It should be noted that due to the number of agents used in some of the experiments and the intensive computational nature of this work that considerable attention was paid to both size and speed, and this dictated the choices of the data structures used etc. At the end of this section is a brief overview of how to compile and run the code.

7.1 NK Landscape program

This file contains 11 procedures that can be split into procedure for generating an NK landscape and procedures for evaluating a bit vector on an NK landscape.

7.1.1 NK landscape generation procedures

The main procedure is called *newlandscape*. This procedure takes as input two integers (N & K) where the second value has to be less than or equal to the first value and generates an NK landscape of the specified size (N genes, K links) with the associated fitness lookup tables (see section 4.3.2 for a detailed specification). Three data types, *nklandscape*, *gene* and *sevenbit*, are defined. The *nklandscape* data type consists of two integer values (N & K) and a list of *gene* data types. The *gene* data type consists of a list of integers (referring to other genes which effect that genes fitness) and a *sevenbit* data type (the fitness lookup table). The *sevenbit* data type is a vector with seven bit slots that hold integer values from 1 to 100. Each procedure has an accompanying initialisation procedure called *newnklandscape*, *newgene*, and *newsevenbit*. The last procedure, *newlink*, generate a list of integers representing the genes to which a particular gene is linked.

7.1.2 Fitness evaluation procedures

The main procedure is called *evaluate*. This procedure takes as input a bit vector representing a genotype and a *nklandscape* data type representing an NK landscape and returns the fitness of the genotype according to that NK landscape. A bit vector data type is defined called *onebit* with an accompanying initialisation procedure called *newonebit*. There are two additional procedures used in *evaluate* called *gen_to_int* and *int_to_gen*. These procedures convert a binary number to a decimal and a decimal to an integer respectively.

7.2 Programs used to run experiments on the selection of evolvability

The main procedure is called *run*. This procedure takes as input an *nklandscape* data type representing an NK landscape, a list of *agents* representing an agent population, an integer representing the number of generations to run, another integer representing a sample size for tournament selection, and a generator of variation procedure for generating the next generation (see section 7.2.2). The data type representing the NK landscape is produced by the programs described in section 7.1.1. The agent population consists of a list of *agent* data types produced by a procedure called *newagent*. The *agent* data type consists of a *onebit* data type representing a genotype, an integer representing a generator type, an integer representing a probability of mutation and an integer representing fitness value (the nature of these variables will become clear in the subsequent discussion). The number of generations is an integer specifying how many subsequent generations are to be produced. The sample size for the tournament selection specifies the number of agents that are randomly picked by this algorithm (explanation follows). The final variable is the procedure which is to be used to produce the next generation. In the following sections the tournament selection procedure will be explained, followed by a description of each generator of variation.

7.2.1 Tournament selection

The main procedure is called *tourn_sel*. This procedure takes as input a list of *agents* representing a population of agents and an integer representing the sample size. Each *agent* has been evaluated by the procedure *evaluate* and the *agent's* fitness is stored as part of the data type. The procedure *top* takes as input a list and a predicate, *top* returns the element

with the highest numerical value according to predicate. The procedure *tourn_sel* uses the procedure *pick_sample* (which randomly selects without removal the sample size number of *agents*) to produce a list of agents and then uses *top* to calculate the fittest agent in that list.

7.2.2 Generators of variation

All the generators of variation are procedures that take the same inputs (a list of *agents* and an integer) and produces the same output (a list of *agents*). All the generator of variation procedures use *tourn_sel* to select the parents of the next generation of agents. However, each of these generators can use additional procedures to produce the offspring that make up the next generation.

- *next_gen0* uses *tourn_sel* to pick parents without removal and then copies them to form the new generation
- *next_gen1* uses a procedure called *mutate* which flips a random chosen bit of an *agent*
- *next_gen2* uses a procedure called *recombination* that takes as input two *agents* and produces a single *agent* by a single point crossover and then uses the procedure *mutate* to produce a single point mutation
- *next_gen3* examines each agent picked by the tournament selection algorithm for its generator type and then if the generator type refers to mutation it creates an offspring using the *mutate* procedure and if the generator type is recombination it uses the *pick_mate* procedure to find another similar type of agent and then creates an offspring using the *recombination* and *mutation* procedures
- *next_genp* uses a procedure called *mutate_mutation* which changes the probability of a mutation occurring, thus the probability of a single mutation varies between 1 and 100 (otherwise *next_genp* acts just like *next_gen1*)

7.2.3 Miscellaneous procedures

The procedure *eval_pop* applies the procedure *evaluate* to every *agent* in a list of agents. The procedures *mean_slot* and *count_slot* calculate the average slot value in an agent population and the number of agents with a slot value of 1 respectively. Finally, the procedure *colstatk* takes as input two integers (corresponding to N and K), a generator of variation procedure

and file name. This procedure creates an NK landscape and then collects the mean population fitness of 40 runs of population ranging from 10 to 1280 agents, and stores the results to the specified file.

7.3 How to run the program

The program is written in the Pop 11 programming language and thus requires the Poplog system to run. Assuming that you have access to this poplog environment, to run the program requires the following steps:

1. type 'ved nkexperiments.p' to open up the main file in the poplog visual editor
2. push 'enter' and type '11' and push 'return' to compile the code
3. the procedure *run* is responsible for executing the code - typing *run(N value (int), K value(int), population size(int), number of generations(int), sample size(int), generator of variation (see above)=>* and then push the '*' button on the number pad will cause the program to run and display the results to the screen.

For example, to run an nklandscape where $N = 100$, $K = 0$, with 1000 agents, for 50 generations, with a tournament selection sample size of 5, and using the mutation generator (*next_gen1*) you would type:

run(100, 0, 1000, 50, 5, next_gen1)=>

and then push '*' button on the number pad to execute it.

8. Appendix 2

A brief functional description of the programs used in the design work described in chapter 5 is given in this appendix. The actual programs can be found on the University of Birmingham, School of Computer Science computer network in the directory ‘~cxc/Public/Code’. The code described in this section took approximately a year to develop, however, unlike the code described in the previous section, with reasonable parameters this code takes 15-30 minutes to run.

For clarity the code has been split up and put into seven files. Each file will be described and only the important procedures discussed. However, as the first file contains the main function-calling procedures all of its procedures will be described. Due to the intensive computational nature of this implementation and the number of agents used in the experiments considerable attention was paid to both size and speed, and this dictated the choices of the data structures used etc.

8.1 Main

This file loads the other files and contains the procedures for running the overall program. The first procedure in the file is called *mean_slot* which takes as input a list of data types representing a list of agents and a procedure for accessing an integer data type slot and as output returns the mean slot value of the list of agents.

The second procedure *create_pop* takes as input three integers representing the number of classifier systems in the population, the number of classifier rules per classifier system, and the number of bits in each condition and action of a classifier rule. The procedure *create_pop* returns a randomly generated list of classifier systems that represent a list of agents that meet the input specifications.

The next procedure *proportional_reward* takes as input a classifier system representing an agent, the stimulus given to the agent, a Boolean representing whether or not the agent is capable of reinforcement learning, an integer representing the reward, and an integer representing the number of bits in a condition or action of a classifier rule. As output *proportional_reward* produces nothing, but its action updates the fitness of the agent and, if the agent is capable of reinforcement learning, the rule responsible for behaviour. The

procedure works by rewarding the agent (and possibly the rule) proportionally to the similarity between the given response and the desired response.

The penultimate procedure is called *run*. This procedure takes as input an integer representing the number of rounds, a classifier system agent representing the desired agent, a list of classifier systems representing the current population of agents, a Boolean representing whether or not agents are capable of reinforcement learning, an integer representing the reward, an integer representing the number of bits in a classifier rules condition or action, a procedure representing a generator of variation to be used with the internal genetic algorithm (explained in the *vacfs* file description), an integer representing the probability of imitation, an integer representing the probability of instruction and an integer representing the tournament selection sample size. This procedure runs a population of agents for a specified number of rounds, presenting a stimulus to each agent in turn and rewarding the responses proportionally to the desired responses.

The last procedure is called *go*. This procedure takes as input an integer representing the number of rounds per generation, an integer representing the number of generations, an integer representing the size of the tournament selection sample, an integer representing the number of agents in a population, an integer representing the number of bits in a classifier condition or action, an integer representing the probability of mutation, a Boolean representing whether or not an agent is capable of reinforcement learning, a procedure representing a generator of variation to be used in the internal genetic algorithm, an integer representing the probability of imitation, an integer representing the probability of instruction, and a Boolean representing whether or not the desired agent changes through time. This procedure runs a population of agents for a set number of generations and returns a list of integers representing the mean population fitness at the end of each generation.

8.2 Bitprocs

This file contains the procedures which define and manipulate the abstract data structure. The abstract data type is a bit vector called *twobit*. This file contains procedures for: randomly generating and mutating bits; for creating random *twobit* data types; for matching, mutating, recombining and comparing the similarity of *twobit* data types; recycling *twobit* data types. This last procedure is very important to the overall program as the genetic

algorithms operating between and within the agents produce prodigious amounts of garbage which significantly slow down the program. However, by initialising two sets of data types and then copying between them rather than generating new data types each time significantly reduces the garbage collection problem.

8.3 Classes

This file contains procedures for creating and manipulating the *cfs* data type that represents a classifier system. The main components of a *cfs* data type are a list of classifier rules representing the phenotype, two lists representing the matched and the message list, and an integer representing fitness and a second list of classifier rules representing the genotype (this is initially the same as the phenotypic list, but it cannot be subsequently altered). There are also procedures for mutating and recombining *cfs* data types as well as for recycling them using the procedure *recyclevec* defined in *bitprocs*. There is also a procedure called *development* which takes the list of classifier rules representing the genotype and produces another identical list of rules representing the phenotype.

This file also contains procedures for defining and manipulating the integral data types of the classifier system: that is, the *cfrs* and *msg* data types representing classifier rules and messages respectively. The *cfs* data type has accompanying mutate, recombine and recycle procedures.

8.4 Vacfs

This file contains the procedures for running the performance system and the rule discovery system of a classifier system (see section 5.1.2). The main procedure is called *run_cfs*. The procedure takes as input a *cfs* representing a classifier system, an integer representing the number of bits in a *twobit* data type and a procedure representing a generator of variation. The *run_cfs* procedure produces no output, but uses the procedure *matcher* to see if any of the classifier system's rules match any of the messages on the message list and then uses the procedure *poster* to generate a response to any matched message. *Run_cfs* can also execute the rule discovery system via the procedure *run_iga* that is executed every 100 rounds runs.

There are three procedure representing generators of variation: an asexually reproducing genetic algorithm, a sexually reproducing genetic algorithm and a sexually

reproducing genetic algorithm with sexual selection. Each generator of variation uses tournament selection (see appendix 1) to pick the parents. The procedure *run_ind_ga1* represents an asexually reproducing genetic algorithm where parent rules are picked using the tournament selection algorithm, then the procedure *genops1* copies and uses the procedure *mutate_cfr* to produce the offspring. The new rules are copied over low fitness rules in the rule population.

The procedure called *run_ind_ga2* represents a sexually reproducing genetic algorithm. This procedure picks two parents using the tournament selection algorithm and then uses the procedure *genops2* to produce a single offspring via the actions of *crossover_cfr* followed by the action of *mutate_cfr*. The sexually reproducing genetic algorithm with sexual selection is a variation of *run_ind_ga2* where the second parent is picked by the tournament selection algorithm on the basis of similarity to the first parent multiplied by fitness. Thus the tournament selection picks a partner for the first parent which is biased by the similarity between the classifier rules.

8.5 Spp_ga

This file contains the procedures for creating new populations of classifier agents. The main procedure is called *run_spp_ga* and takes as input two lists of *cfs* that represent populations of agents. The procedure uses tournament selection to pick the parents and then uses the procedure *spp_ga_ops* to create a new classifier system via the action of recombination and mutation. The procedures which act on the *cfs* data type are defined in the file classes. The list of *cfs* forms the genotype of each *cfs* and these lists are what are crossed by the procedure *crossover_cfs*.

8.6 Remove

This file contains one procedure called *remove* which takes as input a list and an item and returns a list where every occurrence of item has been removed.

8.7 Social

This file contains two procedures called *imitation* and *instruction*. The first procedure, *imitation*, takes as input two lists of *cfs* representing the current generation and the last

generation of agents. Both lists of *cfs* are run in parallel by *run_cfs* allowing the agents in the current population to imitate the behaviour of the agents from the last generation. The procedure *imitation* generates a list of agents from the last generation which have produced behaviour in the current round and then each agent of the new generation has a probability of imitating another agent (an input of the *run* procedure - see section 8.1). If an agent is selected to imitate another agent, then using tournament selection an agent is picked from the list of behaviour producing agents and that agent's current behaviour is copied with mutation into the imitating agent's rule list over a low strength rule.

The procedure *instruction* takes as input two lists of *cfs* representing populations of agents (same as *imitate*). Each member of the last generation has a probability of instructing an agent from the next generation (an input of the *run* procedure). If an agent is selected to instruct, then a 'fit' rule is picked by the tournament selection procedure and an agent is randomly picked from the new generation. The 'fit' rule is copied from the instructor over a low fitness rule of the instructed and the mutation procedure is applied.

8.8 How to run the program

As with the code described in the previous appendix, this program is written in the Pop 11 programming language and thus requires the Poplog system to run. Assuming that you have access to this poplog environment, to run the program requires the following steps:

1. type 'ved main.p' to open up the main file in the poplog visual editor
2. push 'enter' and type '11' and push 'return' to compile the code
3. the procedure *go* is responsible for executing the code (see section 8.1 for a description of the list of the parameters)

For example, to run a scenario where agents interact for 100 rounds, over 50 generations, with a tournament selection size of 10, with agents with 20 rules each with 15 bits, with a generator of variation capable of mutation, with the capability of reinforcement learning and rule discover, and with agents with the ability to imitate and instruct each other with a probability of 0.1, in a dynamic world type:

`go(100, 50, 10, 20, 15, true, run_ind_ga1, 10, 10, true)=>`

and then push '*' button on the number pad to execute it. The data from the run will be displayed on the screen.

9. Glossary

- **Alleles** - one of a group of genes that occur at a particular gene locus (position on homologous chromosomes); in genetic algorithms it refers to the different forms a gene can take (Belew and Mitchell, 1996).
- **Architecture** - a set of functional mechanisms with causal links between them.
- **Design** - a design is a description of an architecture.
- **Epigenesis** - the concept that an organism generates from the new appearance of structures and functions, as opposed to the hypothesis that an organism develops by the unfolding and growth of entities already present in the egg at the beginning of development (Belew and Mitchell, 1996).
- **Epistasis** - a form of gene interaction whereby one gene interferes with the phenotypic expression of another nonallelic gene (Belew and Mitchell, 1996).
- **Gene** - ambiguous term with two at least two sense: 1) in development it refers to nucleic acid sequences; 2) in reproduction it refers to heritable features.
- **Genotype** - the genetic constitution of an individual (Belew and Mitchell, 1996).
- **Niche** - set of requirements.
- **Phenotype** - the detectable expression of the interaction of the genotype and the environment constituting the visible characters of the organism (Belew and Mitchell, 1996).
- **Pleiotropy** - the capacity of allelic substitutions at one gene locus to affect more than one aspect of the phenotype (Abercrombie *et al*, 1990).

10. List of references

- Abercrombie, M. Hickman, M. Johnson, M. L. & Thain, M. 1990. *The New Penguin Dictionary of Biology*. 8th edition. London: Penguin Books.
- Ackley, D. H. & Littman, M. L. 1994. A Case for Lamarckian Evolution. In Langton, C. G. *Artificial Life III*. Reading, MA: Addison-Wesley. pp. 3-10.
- Atkins, P. W. 1994. *The Second Law*. New York: Scientific American Books.
- Baldwin, J. M. 1896. A New Factor in Evolution. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 59-80.
- Bates, J. Loyall, A. B. & Reilly, W. S. 1991. Broad agents. *Sigart Bulletin*. 2(4):38-40.
- Bateson, G. 1963. The Role of Somatic Change in Evolution. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 111-124.
- Beaudoin, L. 1995. *Goal Processing in Autonomous Agents*. Unpublished Ph.D. Thesis. University of Birmingham.
- Belew, R. K. & Mitchell, M. 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley.
- Belew, R. K. Mitchell, M. & Ackley, D. H. 1996. Computation and the Natural Sciences. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 431-440.
- Bonner, J. T. 1980. *The Evolution of Culture in Animals*. Guildford: Princeton University Press.
- Bonner, J. T. 1995. *Life Cycles*. Sussex: Princeton University Press.
- Boyd, R. & Richerson, P. J. 1985. *Culture and the Evolutionary Process*. London: University of Chicago Press.
- Braitenberg, V. *Vehicles: Experiments in Synthetic Psychology*. London: MIT Press.

- Cairns-Smith, A. G. 1991. *Seven Clues to the Origin of Life: a scientific detective story*. Cambridge: Cambridge University Press.
- Calvin, W. H. 1987. The brain as a Darwin Machine. *Nature*. 330: 33-34
- Campbell, D. T. 1974. Evolutionary epistemology. In P. A. Schilpp (ed.). *The Philosophy of Karl R. Popper*. Illinois: Open Court. pp. 413-463.
- Campbell, D. T. 1981. Levels of organisation, selection, and information storage in biological and social evaluation. *The Behavioural and Brain Sciences*. 4: 236-237.
- Cavalli-Sforza, L. L. & Feldman, M. W. 1981. *Cultural Transmission and Evolution: A Quantitative Approach*. Guilford: Princeton University Press.
- Cohen, J. & Stewart, I. 1995. *The Collapse of Chaos: Discovering Simplicity in a Complex World*. London: Penguin.
- Cohen, J. 1995. Who do we blame for who we are? In J. Brockman and K. Matson (eds.). *How Things Are: A Science Tool-Kit for the Mind*. London: Weidenfeld and Nicolson. pp. 51-60.
- Darwin, C. 1859 (1985). *Origin of Species*. London: Penguin.
- Dawkins, R. 1982. Replicators and Vehicles. In King's College Sociology Group (eds.). *Current Problems in Sociology*. Cambridge: Cambridge University Press. pp. 45-64.
- Dawkins, R. 1983a. *The Extended Phenotype*. Oxford: Oxford University Press.
- Dawkins, R. 1983b. Universal Darwinism. In D. S. Bendall (ed.). *Evolution from Molecules to Man*. Cambridge: Cambridge University Press. pp. 403-425.
- Dawkins, R. 1988. *The Blind Watchmaker*. London: Penguin.
- Dawkins, R. 1989a. The Evolution of Evolvability. In C. Langton (ed.). *Artificial Life VI*. Santa Fe: Addison-Wesley. pp. 201-220.
- Dawkins, R. 1989b. *The Selfish Gene*. 2nd edition. Oxford: Oxford University Press.
- Dawkins, R. 1994. Selfish Genes. In T. A. Bass (ed.). *Reinventing the Future: Conversations with the World's Leading Scientists*. Wokingham: Addison-Wesley.
- Dawkins, R. 1995. *River Out of Eden*. London: Weidenfeld & Nicolson.

- Dawkins, R. 1996. *Climbing Mount Improbable*. London: Penguin.
- Dennett, D. C. 1986. *Brainstorms: Philosophical Essays on Mind and Psychology*. Brighton: Harvester Press.
- Dennett, D. C. 1995. *Darwin's Dangerous Idea: Evolution and the Meanings of Life*. New York: Simon & Schuster
- Dennett, D. C. 1996. *Kinds of Minds: Towards an Understanding of Consciousness*. London: Weidenfeld & Nicolson.
- Donnart, J. & Meyer, J. 1994. A Hierarchical Classifier System Implementing a Motivationally Autonomous Agent. In D. Cliff, P. Husbands, J. Meyer, and S. Wilson (eds.). *From Animals to Animats III*. London: MIT Press.
- Dunbar, R. I. M. 1982. Evolutionary tautology. In King's College Sociology Group (eds.). *Current Problems in Sociology*. Cambridge: Cambridge University Press. pp. 9-28.
- Eldridge, N. 1995. *Reinventing Darwin: The Great Evolutionary Debate*. London: Weidenfeld & Nicolson.
- Emmet, E. R. 1991. *Learning to Philosophise*. London: Penguin.
- Fisher, R. A. 1930 (1958). *The Genetical Theory of Natural Selection*. New York: Dover.
- Gabora, L. 1995. Meme and Variations: A Computational Model of Cultural Evolution. In *1993 Lectures in Complex Systems*. Reading, MA: Addison-Wesley.
- Gabora, L. 1997. The Origin and Evolution of Culture and Creativity. *Online Journal of Memetics*. URL: http://www.cpm.mmu.ac.uk/jom-emit/vol1/gabora_1.html.
- Goldberg, D. E. 1989. *Genetic algorithm in search, optimisation, and machine learning*. Wokingham: Addison Wesley.
- Goodwin, B. 1994. *How the leopard changed its spots*. London: Weidenfeld & Nicolson.
- Gould, S. J. 1991. *Ever Since Darwin: Reflections in Natural History*. London: Penguin.
- Gould, S. J. 1997. 'What is Life?' as a problem in history. In M. P. Murphy and L. A. J. O'Neill (eds.). *What is Life? The Next Fifty Years: Speculations on the future of biology*. Cambridge: Cambridge University Press. pp. 25-39.

- Gould, S. J. and Vrba, E. S. 1982. Exaptation - a missing term in the science of form. *Paleobiology*, 8: 4-15.
- Grandison, A. *et al.* (eds.) 1994. *Collins English Dictionary*. 3rd edition. Glasgow: HarperCollins Publisher.
- Harel, D. 1987. *Algorithmics: the spirit of computing*. Wokingham: Addison-Wesley.
- Harvey, I. 1997. *Personal communication*.
- Harvey, I & Thompson, A. 1996. Through the Labyrinth Evolution Finds a Way: A Silicon Ridge. In *Proceedings of The First International Conference on Evolvable Systems: From Biology to Hardware*. Springer-Verlag.
- Hebb, D. 1949 (1966). *The Organisation of Behaviour*. New York: John Wiley & Sons.
- Hinton, G. E. & Nowlan, S. J. 1987. How Learning Can Guide Evolution. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 447-453
- Hofstadter, D. R. 1986. *Metamagical Themas: Questing for the Essence of Mind and Pattern*. London: Penguin.
- Holland, J. H. 1986. Escaping Brittleness: the possibility of general-purpose learning algorithms applied to parallel rule-based systems. In R. S. Michalski, J. G. Carbonell & T. M. Mitchell (eds.). *Machine learning, an artificial intelligence approach. Volume II*. Los Altos, California: Morgan Kaufmann. pp. 593-623.
- Holland, J. H. 1994. *Adaptation in Natural and Artificial Systems*. 2nd edition. London: MIT Press.
- Holland, J. H. 1995. *Hidden Order: How Adaptation Builds Complexity*. Wokingham: Addison-Wesley.
- Hopcroft, J. E. & Ullman, J. D. 1979. *Introduction to Automata Theory, Language, and Computation*. London: Addison-Wesley.
- Hull, D. L. 1988a. Interactors versus Vehicles. In H. C. Plotkin (ed.). *The Role of Behaviour in Evolution*. London: MIT Press. pp. 19-50.

- Hull, D. L. 1988b. *Science as a Process: An Evolutionary Account of the Social and Conceptual Development of Science*. London: University of Chicago Press.
- Humphrey, N. K. 1976. The Social Function of Intellect. In P. P. G Bateson & R. A. Hinde (eds.). *Growing points in ethology*. Cambridge: Cambridge University Press. pp. 303-317.
- Huynen, P. F. Stadler, P. F. & Fontana, W. 1996. Smoothness within ruggedness: the role of neutrality in adaptation. *Proc. Natl. Acad. Sci. USA*, 93:397-401.
- James, W. 1880. Great men, great thoughts, and the environment. *Atlantic Monthly*. 46: 441-449.
- Jeffers, J. N. R. 1982. *Modelling*. London: Chapman and Hall.
- Johnston, T. D. 1982. Selective Costs and Benefits in the Evolution of Learning. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 315-358.
- Kauffman, S. A. 1993. *The Origins of Order: Self-Organisation and Selection in Evolution*. Oxford: Oxford University Press.
- Kauffman, S. A. 1995. *At Home in the Universe: The Search for Laws of Self-Organisation and Complexity*. London: Viking.
- Koza, J. R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. London: MIT Press.
- Koza, J. R. 1994. *Genetic Programming II: Automatic Discovery of Reusable Programs*. London: MIT Press.
- Koza, J. R. 1995. Gene Duplication to Enable Genetic Programming to Concurrently Evolve Both the Architecture and Work-Performance Steps of a Computer Program. In C. S. Mellish (ed.). *Proceedings of 14th International Conference in AI*. San Mateo, California. Morgan Kaufmann. pp. 734-740.
- Krebs, J. R. & Davies, N. B. 1987. *An Introduction to Behavioural Ecology*. 2nd edition. Oxford: Blackwell Scientific Publications.

- Lamarck, J. B. 1914. Zoological Philosophy. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 39-57.
- Lamb, D. A. 1988. *Software Engineering: Planning for Change*. London: Prentice-Hall.
- Langton, C. G. 1989. Artificial Life. In C. Langton (ed.). *Artificial Life VI*. Santa Fe: Addison-Wesley. pp. 1-47.
- Lewontin, R. C. 1970. The units of selection. *Annual Review of Ecology and Systematics*, 1: 1-18.
- Lewontin, R. C. 1981. On constraints and adaptation. *The Behavioural and Brain Sciences*. 4: 244-245
- Lloyd Morgan, C. 1896. On Modification and Variation. In R. K. Belew and M. Mitchell (eds.). *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 81-90.
- Losse, J. 1993. *A Historical Introduction to the Philosophy of Science*. 3rd edition. Oxford: Oxford University Press.
- Lumsden, C. & Wilson, E. O. 1981. *Genes, Mind, and Culture*. London: Harvard University Press.
- Magee, B. 1997. *Confessions of a Philosopher*. London: Weidenfeld & Nicolson.
- Magee, B. 1988. *The Great Philosophers: An Introduction to Western Philosophy*. Oxford: Oxford University Press.
- Magee, B. 1990. *Popper*. London: Fontana Press.
- Manning, A. and Stamp Dawkins, M. 1992. *An Introduction to Animal Behaviour*. 4th edition. Cambridge: Cambridge University Press.
- May, R. M. 1977. Population genetics and cultural evolution. *Nature*. 268:11-13.
- Mayley, G. 1996. The Evolutionary Cost of Learning. In P. Maes, M. Mataric, J. Meyer, J. Pollack, and S. Wilson (eds.). *From animals to animats 4*. London: MIT Press. pp. 458-467.

- Maynard Smith, J. & Szathmary, E. 1995. *The Major Transitions in Evolution*. Oxford: W. H. Freeman.
- Maynard Smith, J. & Szathmary, E. 1997. Language and Life. In M. P. Murphy and L. A. J. O'Neill (eds.). *What is Life? The Next Fifty Years: Speculations on the future of biology*. Cambridge: Cambridge University Press. pp. 67-77.
- Maynard Smith, J. 1976. Evolution and the Theory of Games. *American Scientist*. 64: 41-45.
- Maynard Smith, J. 1977. The Limitations of Evolutionary Theory. In J. Maynard Smith. 1993. *Did Darwin Get it Right*. London. Penguin. pp. 180-191.
- Maynard Smith, J. 1982. Genes and Memes. In J. Maynard Smith. 1993. *Did Darwin Get it Right*. London. Penguin.
- Maynard Smith, J. 1987. When Learning Guides Evolution. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 455-457.
- Maynard Smith, J. 1994. *Evolutionary Genetics*. Oxford: Oxford University Press.
- Mayr, E. 1974. Behavioural Programs and Evolutionary Strategies. *American Scientist*. 62: 650-659.
- Moravec, H. 1989. Human Culture: A Genetic Takeover Underway. *Artificial Life VI*. Santa Fe: Addison-Wesley. pp. 167-199.
- Newell, A. 1982. The Knowledge Level. *Artificial Intelligence*. 18: 87-127.
- Newell, A. 1992. Précis of the Unified Theory of Cognition. *The Behavioural and Brain Sciences*. 15: 425-492.
- Newell, A. 1994. *Unified Theories of Cognition*. London: Harvard University Press.
- Nussbaum, M. 1988. Aristotle. In B. Magee (ed.). *The Great Philosophers: An Introduction to Western Philosophy*. Oxford: Oxford University Press. pp. 34-54.
- Piaget, J. 1980. Adaptation and Intelligence: Organic Selection and Phenocopy. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 295-308.

- Plotkin, H. C. & Odling-Smee, F. J. 1981. A multiple-level model of evolution and its implications for sociobiology. *The Behavioural and Brain Sciences*. 4: 225-268.
- Plotkin, H. C. 1988. Behaviour and evolution. In H. C. Plotkin (ed.). *The Role of Behaviour in Evolution*. London: MIT Press. pp. 1-17.
- Plotkin, H. C. 1991. The Testing of Evolutionary Epistemology. *Biology and Philosophy*. 6: 481-497.
- Plotkin, H. C. 1995. *Darwin Machines and the Nature of Knowledge*. London: Penguin.
- Popper, K. R. 1986. *Objective Knowledge: An evolutionary approach*. Revised edition. Oxford: Clarendon Press.
- Rees, A. R. & Sternberg, M. J. E. 1987. *From Cells to Atoms: An illustrated introduction to molecular biology*. Oxford: Blackwell Scientific Publications.
- Ricklefs, R. E. 1990. *Ecology*. 3rd edition. New York: W. H. Freeman and Company.
- Ridley, M. 1990. *The Problems of Evolution*. Oxford: Oxford University Press.
- Russell, B. 1959. *Wisdom of the West: A historical survey of western philosophy*. London: Macdonald.
- Salamon, R. 1996. Increasing Adaptivity through Evolutionary Strategies. In P. Maes, M. Mataric, J. Meyer, J. Pollack, and S. Wilson (eds.). *From animals to animats 4*. London: MIT Press. pp. 411-420.
- Salthe, S. N. 1993. *Development and Evolution: Complexity and Change in Biology*. London: MIT Press.
- Schull, J. 1996. William James and the Broader Implications of a Multilevel Selectionism. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 243-256.
- Simon, H. A. 1982. *The Sciences of the Artificial*. 2nd edition. London: MIT Press.
- Simon, H. A. 1996. *The Sciences of the Artificial*. 3rd edition. London: MIT Press.
- Simpson, G. G. 1953. The Baldwin Effect. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 99-109.

- Skinner, B. F. 1969. The Phylogeny and Ontogeny of Behaviour. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. (pp. 257-290) Santa Fe: Addison-Wesley.
- Sloman, A. 1969. How to derive Better from Is. *American Philosophical Quarterly*. 6(1): 43-52.
- Sloman, A. 1978. *The computer revolution in philosophy: Philosophy, science, and models of mind*. Atlantic Highlands, NJ: Humanities Press.
- Sloman, A. 1994. Explorations in Design Space. In A. G. Cohn (ed.). *ECAI94 11th European Conference on Artificial Intelligence*. Amsterdam: John Wiley. pp. 578-582.
- Sloman, A. 1996a. Actual Possibilities. In L. C. Aiello & S. C. Shapiro (eds.). *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference*. Morgan Kaufmann.
- Sloman, A. 1996b. *Personal communication*.
- Sloman, A. 1997. What sort of architecture is required for a human-like agent? In M. Wooldridge & A. Rao (eds.). *Foundations of Rational Agency*. Kluwer Academic.
- Smith, P. K. & Cowie, H. 1991. *Understanding Children's Development*. Oxford: Blackwell Publishers.
- Todd, P. M. 1996. Sexual Selection and the Evolution of Learning. In R. K. Belew and M. Mitchell (eds.). *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 365-393.
- Tomasello, M. Kruger, A. C. & Ratner, H. H. 1993. Cultural learning. *Behavioural and Brain Sciences*. 16: 495-552.
- Waddington, C. H. 1942. Canalization of Development and the Inheritance of Acquired Characteristics. In R. K. Belew and M. Mitchell (eds.). 1996. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Santa Fe: Addison-Wesley. pp. 91-96.
- Waldrop, M. M. 1994. *Complexity: The emerging science at the edge of order and chaos*. London: Penguin.

- Wilcox, J. R. 1995. *Organisational Learning Within A Learning Classifier System*. Unpublished M.Sc. Thesis. University of Illinois.
- Williams, G. C. 1966. *Adaptation and Natural Selection: A Critique of Some Current Evolutionary Thought*. Princeton: Princeton University Press.
- Williams, G. C. 1992. *Natural Selection: Domains, Levels, and Challenges*. Oxford: Oxford University Press.
- Williams, G. C. 1996. *Plan and Purpose in Nature*. London: Weidenfeld and Nicolson.
- Wilson, S. W. 1994. ZCS: A Zeroth Level Classifier System. *Evolutionary Computation*. 2: 1-18.
- Wilson, S. W. 1995. Classifier Fitness Based on Accuracy. *Evolutionary Computation*. 3(2): 149-175.
- Wolpert, D. & Macready, W. 1995. *No Free Lunch Theorems for Search*. Technical Report SFI-TR-95-02-010. Santa Fe Institute.
- Wolpert, L. 1969. Positional information and the spatial pattern of cellular differentiation. *Journal of Theoretical Biology* 25:1-447.
- Wolpert, L. 1995. Triumph of the embryo. In J. Brockman and K. Matson (eds.). *How Things Are: A Science Tool-Kit for the Mind*. London: Weidenfeld and Nicolson. pp. 61-67.
- Wolpert, L. 1997. Development: Is an egg computable or could we generate an angel or a dinosaur. In M. P. Murphy and L. A. J. O'Neill (eds.). *What is Life? The Next Fifty Years: Speculations on the future of biology*. Cambridge: Cambridge University Press. pp. 57-66.
- Zilberstein, S. 1996. Using Anytime Algorithms in Intelligent Systems. *AI Magazine*. 17(3): 73-83.