Coupling evolution and self-organisation in bio-inspired communication systems

Rob Shipman

Submitted for the degree of EngD

University College London

April 2003

Abstract

The role of self-organisation in biological evolution has received an increasing amount of attention in recent years. This work suggests that biological order is the result of the coupling of natural selection *and* self-organisation. Such a change in perspective has the potential to profoundly affect our view of evolutionary adaptation. However, to date there has been little emphasis placed on the consequences of self-organisation in artificial evolution. This thesis focuses on one such consequence; the generation of neutrality in the mapping from genotype to phenotype. Self-organisation results in large sets of genotypes that produce the same phenotype which can form *neutral networks* that percolate throughout genotype space. An evolving population is able to continually discover new phenotypes through neutral drift on these networks. Such a process may therefore be an important component of evolutionary creativity.

The aim of this work is to develop artificial genotype-phenotype mappings that enhance evolutionary search by introducing beneficial neutrality into the search space. Self-organising mappings are developed that are based on abstractions of natural processes and their properties are analysed in depth to determine whether they can provide any performance enhancement over a more traditional direct encoding. The work is then extended to tackle a real-world problem involving the evolutionary design of telecommunication networks. Domain knowledge in the form of network planning rules is used to "grow" a network in the context of its simulated environment. The parameters of this growth process are encoded in the genotype allowing evolution to tune its dynamics and the impact of this approach is determined through exhaustive enumeration of the resulting search space. Subsequent comparison to a direct encoding on more challenging problems reveals that the approach expedites evolutionary progress and scales well to larger and more complex problems.

Acknowledgements

It has been a privilege to have been immersed in the very creative and stimulating environment of BTexact's Future Technologies Group during the course of this work. The group is full of very talented individuals who bring a diversity of expertise and opinion to any debate. Much of the credit for this environment goes to Mark Shackleton whose technical expertise and insight is matched only by his inter-personal skill. Thanks also to Richard Tateson for answering many naïve biological questions with enthusiasm even after being rudely interrupted by seemingly random queries directed over his office partition. I could not have asked for better people with which to share this experience.

Many thanks also to Peter Bentley who provided a very appropriate home at UCL at a time when I was beginning to doubt whether such a home existed. I feel very fortunate to have found Peter at UCL not least for the rapid and very useful comments on various drafts of this thesis. I'd also like to acknowledge the support of the Engineering Doctorate Centre in Communications at UCL who allowed me the freedom to pursue this work even though the relevance to communications at times may have seemed tenuous. Thanks also to the EPSRC for the industrial scholarship award that has proved invaluable for funding various training and conference trips.

This work has greatly benefited from numerous discussions with colleagues from across the evolutionary computation and artificial life communities. In particular, I'd like to acknowledge Inman Harvey who first triggered my interest in neutral networks. I'd also like to thank Lionel Barnett, Seth Bullock, Phil Husbands, Jason Noble, Tom Smith, Richard Watson and many other friends and colleagues with whom I have shared stimulating discussions that were typically accompanied by equally stimulating foreign beer. I look forward to enjoying your company on many occasions in the future. I cannot conclude without acknowledging my principal collaborators, Mark Shackleton and Marc Ebner. Our very fruitful collaboration was forged in the heat of the Santa Fe summer school, a truly rewarding experience that effectively marked the beginning of this journey. I'm sure that the submission of this thesis will not mark its end.

Finally and most importantly I'd like to thank my family and in particular my parents Brian and Patricia. Their resolute and selfless support in every endeavour I undertake is always appreciated and will never be forgotten. Granddad – I hope Grandma would have been proud, this thesis is dedicated to her.

Rob Shipman, April 2003.

Preface

The work described in chapter 4 was initiated at the 1999 Santa Fe Institute summer school at which a working group focussing on neutral networks was formed. This group involved Marc Ebner, Susan Ptak, Mark Shackleton, Tracy Teal and Richard Watson. It subsequently led to a fruitful collaboration with Mark Shackleton and Marc Ebner in particular which resulted in a number of publications that are listed below:

1. Rob Shipman, Mark Shackleton, Marc Ebner, and Richard Watson, "Neutral Search Spaces for Artificial Evolution: A Lesson from Life," in *Proceedings of the Seventh International Conference on Artificial Life*, Mark A. Bedau, John S. McCaskill, Norman H. Packard, and Steen Rasmussen, Eds. MIT Press, 2000, pp. 162-169.
2. Rob Shipman, Mark Shackleton, Marc Ebner, and Richard Watson, "Neutral Search Spaces for Artificial Evolution: A Lesson from Life," in *Proceedings of the Artificial Life 7 Workshops*, Carlo C. Maley and Ellis Boudreau, Eds., 2000.
3. Rob Shipman, Mark Shackleton, and Inman Harvey, "The Use of Neutral Genotype-Phenotype Mappings for Improved Evolutionary Search," *BT Technology Journal*, vol. 18, no. 4, October, 2000.
4. Mark Shackleton, Rob Shipman, and Marc Ebner, "An Investigation of Redundant Genotype-Phenotype Mappings an Their Role in Evolutionary Search," in *Proceedings of the 2000 Congress on Evolutionary Computation*, 2000, Vol. 1, pp. 493-500.
5. Marc Ebner, Mark Shackleton, and Rob Shipman, "How Neutral Networks Influence Evolvability," *Complexity*, 2001.
6. Marc Ebner, Patrick Langguth, Juergen Albert, Mark Shackleton, and Rob Shipman, "On Neutral Networks and Evolvability," in *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001, Vol. 1, pp. 1-8.

The approach to the evolutionary design of telecommunication networks introduced in chapters 5 and 6 was developed in consultation with BTexact's Planning and Implementation group. This group subsequently became Evolved Networks, a BT spin-out company. The approach was introduced in the following publications:

7. Rob Shipman, Paul Botham, and Paul Coker, "Coupling Developmental Rules and Evolution to Aid in Planning Network Growth," *BT Technology Journal*, vol. 18, no. 4, October, 2000.

8. Erwin Bonsma, Mark Shackleton, and Rob Shipman, "Eos - an Evolutionary and Ecosystem Research Platform," *BT Technology Journal*, vol. 18, no. 4, October, 2000.

More detailed analysis was presented in the following papers:

9. Rob Shipman, Mark Shackleton, and Inman Harvey, "The Use of Neutral Genotype-Phenotype Mappings for Improved Evolutionary Search," *BT Technology Journal*, vol. 18, no. 4, October, 2000.

10. Rob Shipman and Mark Shackleton, "Issues in Designing a Neutral Genotype-Phenotype Mapping," in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*. IEEE Press, 2002, pp. 1360-1365.

In addition, the intellectual property was protected by the following patent which has subsequently been licensed by Evolved Networks:

11. Rob Shipman, "Design of Communication Networks," *EP 01300152.4*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Artificial evolution and self-organisation

The creative power of evolution is evident throughout the biological world. Examples abound of ingenious solutions to problems posed by the environment in which an organism must survive. This creative potential has been well recognised by human designers and it is thus no surprise that there have been numerous attempts to harness the power of evolution over the last 40 years for a diversity of applications that range from the evolution of artificial life forms to evolutionary art work [18,19]. In common with many new ideas and concepts, artificial evolution was independently developed on several occasions. The 1960's saw the birth of two similar techniques, Evolutionary Strategies [30] and Evolutionary Programming [47]. The former was originally developed to optimise manufactured shapes including the bends in pipe work and the structure of nozzles, the latter to produce machine intelligence through use of finite state machines. More recently, the genetic algorithm was developed which more closely resembles biological evolution and has subsequently become the most widely used evolutionary algorithm [39]. It has also spawned genetic programming which was explicitly designed to evolve computer programs [40].

As the underlying inspiration for all these techniques is biological evolution, it is inevitable that they inherit the dominant biological theories of their age. Since the rediscovery of Mendel's particular inheritance first published in 1865 [27], the inheritable particles or genes that encode the information required to construct an organism have come to assume a central role in theories of biological evolution. It is the genes that are passed from generation to generation with novelty introduced through random genetic mutations and potentially sexual recombination. Darwin's incredibly influential theory of natural selection [8] provides the mechanism by which the organism and hence the associated genotype becomes better adapted to its environment. This is the biological background from which artificial evolution was born and thus the typical evolutionary algorithm operates in a very similar way. The free variables of the problem are encoded into an artificial genotype and one or more genotypes are maintained within the evolutionary algorithm. The fitness of a genotype is assessed by determining how well the associated solutions solve the given problem. Genotypes that yield higher quality solutions are preferentially selected as the basis of the next generation and novelty introduced through genetic operators such as mutation and sexual recombination. In this way, generation after generation, the quality of the encoded solutions gradually increases.

This methodology has met with considerable success. However its biological foundations, which view biological order as the sole product of natural selection operating on the genes, have come under increasing question in recent years. It has been argued that natural selection alone is not sufficient to provide a full account of biological order. Rather, it must be viewed in the context of a biological system's natural dynamics which cause the system to self-organise into stereotypical patterns of activity [6,111]. Such processes provide a natural source of order that must be respected by natural selection. In this view, the creative solutions found in the biological world are a product of both natural selection *and* self-organisation. Given that self-organisation may be fundamentally important to biological evolution it may also be necessary to give it due consideration in artificial evolution for it to more fully emulate the power of its natural counterpart. This hypothesis is the overriding theme of this work.

1.2 Neutral networks

Self-organisation is evident at many levels in biological systems, from the spontaneous folding of a biopolymer into an intricate three-dimensional structure to the coordinated activity of many such molecules during biological development. One consequence of this self-organisation is that many different initial states of the system give rise to the same system behaviour i.e. many different genotypes give rise to the same phenotype. It becomes possible, therefore, to modify the genotype without affecting the resulting phenotype. It has been proposed that the majority of biological mutations equate to such *neutral* modifications [53]. The presence of large-scale neutrality changes the picture of evolutionary adaptation on which artificial evolution was based. In addition to genetic changes that result in a different fitness value, an evolving population can also engage in *neutral drift* via a series of neutral modifications. This has the potential of alleviating an ongoing problem with artificial evolution and indeed many other search algorithms; local optima. If no immediate adaptive modifications to the genotype are possible then evolutionary progress will ordinarily halt. However, neutral mutations have the potential to "set the scene" for subsequent adaptive modifications and hence continuing evolutionary progress.

The characteristics and impact of neutrality has been studied in great detail in the context of the folding of biopolymers such as RNA and protein. These studies, reviewed in the following chapter, revealed a number of desirable properties suggesting that molecular self-organisation produces a very amenable evolutionary search space that is largely untroubled by the presence of local optima. Large sets of genotypes produced the same molecular structure or phenotype and were connected by single genetic mutations forming *neutral networks* [85]. It was discovered that drift on these networks allowed for the constant discovery of new phenotypes and thus greatly diminished the probability of becoming trapped at local optima [65]. Encouraging similar

properties in to artificial evolutionary search spaces through appropriate use of self-organisation is a principle concern of this thesis.

1.3 Aims and objectives

The primary objective of this thesis is to determine whether the efficacy of an evolutionary algorithm can be enhanced through the coupling of self-organisation and evolution. There are two main components to this objective. Firstly, to determine whether self-organising genotype-phenotype mappings can be developed that result in search spaces with similar properties to those found in natural search spaces as evidenced by biopolymer folding. Secondly, to determine whether the use of such mappings within an evolutionary algorithm provides any advantage over more traditional encodings that directly represent the free variables of the problem in the genotype. A further objective is to address these questions in the context of a specific real-world application; the evolutionary design of telecommunication networks.

This work focuses on one aspect of the use of self-organisation within artificial genotype-phenotype mappings; the impact of neutrality. Are neutral networks created and what are their characteristics? Does drift on a neutral network enable the discovery of new phenotypes that would otherwise have been impossible? Does neutral drift allow the discovery of higher fitness phenotypes? Can local optima be removed from the search space? Does neutrality introduce any biases into the search space? These are some of the questions that are addressed on route to the primary objectives.

1.4 Original contributions

This work makes a number of original contributions to the field that are summarised below:

- The *development of self-organising genotype-phenotype mappings* that are based on abstractions of natural processes. These mappings are based on a cellular automaton and a random Boolean network both of which are widely used to model natural self-organising processes.
- Demonstration that these abstract genotype-phenotype mappings result in *expansive neutral networks* that allow for the *constant innovation of new phenotypes*.
- Demonstration that the increased accessibility of new phenotypes afforded by the abstract mappings can *alleviate the problem of local optima* in the context of challenging search spaces.
- The development of a *novel approach to the evolutionary design of telecommunication networks*. Rather than directly encode the network within the genotype, this approach encodes the instructions for generating that network which are interpreted in the context of its simulated environment.

- Demonstration that this approach can create a *search space that shares key properties of natural search spaces and contains no local optima*.
- Demonstration that a key aspect of the use of self-organisation within artificial evolution is to *bias the search space in favour of high quality phenotypes*. This effect allowed high quality network designs to be discovered much more rapidly through use of a self-organising mapping in comparison to a direct encoding.
- Demonstration that this approach is *better able to scale to larger and more complex problems* than a direct encoding.

1.5 Thesis overview

In the following chapter, the underlying motivation for this work is expanded on. The concept of self-organisation is introduced and it is shown how self-organising processes are inherent to the physiochemical foundations of biological systems. In addition, examples are given that suggest a crucial role for self-organisation in current biological systems and indeed the origin of life itself. Chapter 2 also draws out the effect of self-organisation on an evolutionary search space; the concept of neutral networks is expanded on and the potential impact on evolution is highlighted. One of the most thorough studies of the impact of neutral networks has focussed on RNA folding. These studies are a primary influence for this work and are reviewed in detail at the close of the chapter.

Chapter 3 reviews existing work within artificial evolution that has considered the impact of neutrality. It is highlighted that neutrality is likely to have played an implicit and unintended role in many artificial evolutionary experiments. Recently however, there have been a number of explicit studies which are reviewed in detail. The majority of these studies introduce neutrality into a genotype-phenotype mapping by allowing for segments of the genotype that are currently unexpressed in the phenotype. While this approach has met with initial success, it is argued that the order arising from a self-organising process must be exploited to gain full advantage of neutrality.

The fourth chapter introduces two self-organising genotype-phenotype mappings. These mappings are based on a cellular automaton which have been widely used to model natural self-organising processes and a random Boolean network which were initially developed as abstract models of genetic regulatory networks. The properties of the mappings are extensively analysed to determine whether their use within an evolutionary algorithm can alleviate the problem of local optima. The effect of the mappings is ascertained using example search spaces that are shown to be problematic for a traditional direct encoding. It is demonstrated that both mappings can substantially improve the performance of a hill-climbing search algorithm in comparison to a direct encoding. It is argued, however, that these results would not scale to more realistic problems.

Chapter 5 uses the knowledge gained during the study of the abstract mappings to develop a novel approach to the evolutionary design of telecommunication networks. Rather than directly encode a network design in a genotype, this approach encodes the instructions for generating a network design. These instructions are interpreted in the context of the network's simulated environment in order to "grow" a network through use of a self-organising process. This approach is applied to a simplified version of a real problem, the growth of the UK's data communications network. The resulting search space is exhaustively enumerated in order to precisely determine the impact of the approach. It is shown how seemingly innocuous design choices can have detrimental effects on the search space and how the design of the growth process can be modified to remove these effects. The end result is a search space that is heavily biased in favour of high quality network designs and that does not contain any local optima.

This approach is extended in chapter 6 to larger scale and more complex network design problems. The overall aim of this chapter is to compare the network growth process with a more traditional direct encoding in which the exact network structure is represented in the genotype. It is shown that the biases introduced into the search space through use of the growth process allow a search algorithm to very quickly discover high quality network designs. On relatively small-scale and low complexity problems the direct encoding eventually allows better network designs to be discovered. However, it is shown that performance using the direct encoding does not scale as well to larger, more complex problems.

The final chapter summarises and discusses the results with a view to determining how well the objectives were met. In addition, suggestions for future work are given.

Chapter 2

Self-Organisation and Neutral Networks

2.1 Introduction

Against a background of ever increasing entropy, biological organisms stand out as remarkable sources of order. The second law of thermodynamics teaches us that the natural tendency of a system is to move towards its most disorderly state, yet the natural world is ripe with systems that maintain an orderly state that is exceptionally well adapted to its environment. The source of this order has been eternally puzzling for mankind. However, since the seminal contributions of Darwin [8], the theory of natural selection has become the accepted view in the biological community. In this view, the organisms we see around us today are the accumulation of innumerable random modifications to the first persistent systems that were capable of reproduction. The systems that are better able to sustain themselves and reproduce in a competitive environment pass their favourable designs on to their offspring and as a result, generation after generation, the species becomes better adapted to its environment.

With the discovery of the structure of DNA [37], the mechanism of information transfer between generations became clear. Biological information is encoded into strings of chemical bases called nucleotides and a particular trait is encoded as a subset of nucleotides called a gene. Natural selection operates on the genes, which are passed from parent to offspring untouched by the adaptations of the organism during its lifetime. This theory has had enormous success in explaining the order in the biological world and there is no shortage of persuasive arguments to support it [92,93]. As a result, the biological spotlight has largely focussed on the nature of genes and the organism has become to be seen as little more than a carrier of genetic information that can be arbitrarily moulded to ensure this information is not lost.

Despite the success of Darwin's theory and the dominance of the view outlined above, it has not been universally accepted as the sole source of biological order. As early as 1917, D'Arcy Thompson eloquently argued that biological order reflected deep physical and mathematical laws and provided a number of examples of the biological expression of these laws [11]. His claim was not that Darwinian evolution was false but rather that it need not labour alone in crafting biological order. Biological systems are built on physical and chemical foundations that naturally exhibit pattern and order that can be exploited by, and likely constrain, evolution. In the context of the burgeoning interest in Darwinian evolution his arguments did not find widespread acceptance. However, in recent years this view has received renewed interest. An extreme

example is the work of Brian Goodwin who views phenomena such as these as the primary source of biological order, reserving only a minor role for natural selection [6]. While few agree with this extreme position, evidence mounts that the order resulting from natural system dynamics has played a significant role in crafting biological systems and may deserve greater attention from the biological community [86,94,111,112]. Such a change in emphasis has the potential to profoundly affect our understanding of evolution.

2.2 Self-Organisation

The emergence of structure in a system in the absence of explicit external control has become known as self-organisation. The structure emerges as a result of the interactions between the internal components of the system, which effectively place constraints on the forms that the system can take. Self-organisation is a common occurrence in physical and chemical systems and there are many familiar examples, from the complex symmetrical form of a snowflake to the water vortexes that are formed when bath tubs are drained. A slightly less familiar but widely studied example is the formation of Bénard cells in a layer of fluid that is driven away from equilibrium through uniform heating as show in Figure 2.1 [56].

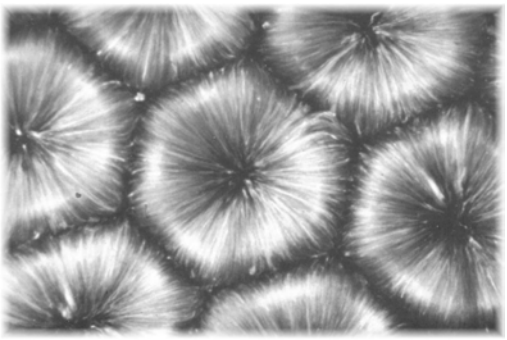


Figure 2.1: Bénard cells in a layer of liquid that is driven away from equilibrium through uniform heating. Liquid rises at the centre of the cells and falls towards their edges. This large-scale structure spontaneously arises when a sufficient temperature gradient exists between the top and bottom of the layer of liquid.

At equilibrium, a layer of fluid is uniform and symmetrical containing no large scale structure. However, if the liquid is heated from below a temperature gradient is created between the top and bottom of the layer. For low differentials the temperature is adequately dissipated through conduction. However, as the gradient increases conduction is no longer sufficient and the liquid suddenly and spontaneously adopts a different form. The lower layer of liquid tends to rise due to its increased temperature while the upper layer tends to fall. The two layers cannot pass through each other and thus convection rolls are formed as the liquid circulates. It is these convection rolls that have become known as Bénard cells. As the liquid is driven further from equilibrium different large-scale structures emerge such as parallel rolls, spirals and target patterns. Each of

these patterns is formed through the coordinated activity of hundreds of billions of molecules without any *explicit* external control. The heat gradient does not contain any code or information dictating the form of the pattern, rather the pattern emerges spontaneously in response to the constraints imposed by the heat gradient.

Another commonly cited example of self-organisation is the Belousov-Zhabotinsky (BZ) chemical reaction, named after the scientists that discovered it in the 1950's. While performing unrelated chemical experiments Belousov serendipitously discovered that a particular mixture of organic and inorganic chemicals generated chemical oscillations, a phenomenon thought to be impossible at the time. When a homogeneous layer of the solution is left undisturbed, various geometric patterns spontaneously emerge and propagate across the medium. These patterns, such as concentric circles and spirals, oscillate in space and time. They are shown in Figure 2.2.

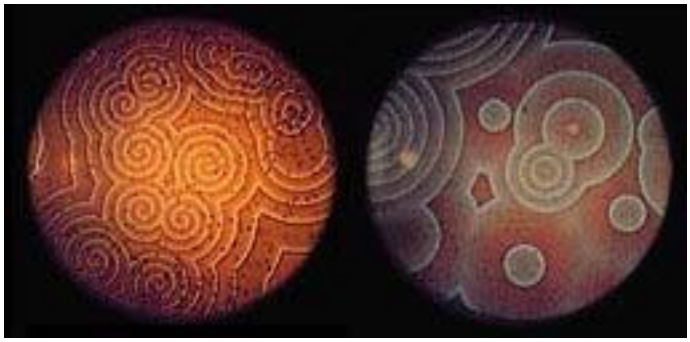


Figure 2.2: Dynamic patterns in the Belousov-Zhabotinsky reaction, spiral patterns are shown to the left of the figure and concentric circles to the right. These patterns spontaneously emerge from a homogeneous layer of a solution containing certain organic and inorganic chemicals.

The BZ reaction is an example of a reaction-diffusion system and relies upon the chemical process of catalysis in which a chemical compound speeds up the rate of a chemical reaction without itself being altered in the process. In particular the reaction involves autocatalysis in which the catalytic reactions make their own catalysts. Left unchecked this positive feedback loop would run out of control and all available reactants would quickly be used. However, the reaction also generates a side-product which in sufficient quantities enables a competing process that inhibits the autocatalytic reaction. The oscillatory nature of the reaction is a result of the competition between these processes. When the autocatalytic reaction is occurring it generates a set of products which diffuse through the medium, when it is inhibited a different set of products is formed. Left undisturbed the chemical system will eventually reach equilibrium, however, if a flow is established in which reactants are continually replaced and the products removed the system will maintain its non-equilibrium pattern forming state. The B-Z reaction is thus a chemical example of the ability of a system to spontaneously exhibit complex, ordered behaviour in the absence of explicit external control.

The spontaneous order seen in the examples given above are not isolated cases, rather pattern formation is at the heart of nature [86]. Given the propensity of the physical and chemical foundations of biological systems to generate patterns we might expect to see examples of similar phenomenon in biology. This chapter highlights examples of pattern formation that may be important components of current biological systems as well as fundamental to the origins of life itself. Firstly however, several important features of self-organising systems are drawn out that will provide context for later work.

State space – an abstract space in which each location uniquely and completely describes the condition of the system. Each variable within the system defines a dimension in state space and the current value of the variables gives a location in that space. As an example, consider a reaction-diffusion system consisting of 100 chemicals that can be in one of two states; present or absent. The state space of this system would be a Boolean hypercube of 100 dimensions with a total of 2^{100} locations or system states.

Attractor – a preferred behavioural pattern that the system naturally progresses towards. The simplest form of attractor is a fixed point in which the system progresses to a single location in state space and remains there until further perturbed. In a chemically reactive system this would correspond to chemical equilibrium. A more complex attractor is the limit cycle in which the system continuously cycles through a subset of states as in the B-Z reaction patterns shown above. More complex still are strange attractors in which the system adopts a chaotic behavioural pattern. The presence of attractors forces the system into a subset of its potential states and thus causes a contraction in effective state space volume.

Basin of Attraction – the region of state space from which the system progresses to a given attractor. Attractors are said to drain areas of state space, these areas correspond to the initial states of the system from which it naturally progresses to that attractor. The drainage areas define the basin of attraction.

Dissipative structures – self-organisation typically occurs in systems that are away from thermodynamic equilibrium and open to the flux of matter and energy. The examples above reveal the possibility of such dissipative systems to generate stable structures such as the Bénard cell and the spiral patterns of the B-Z reaction. These structures are called dissipative structures - a term widely attributed to Prigogine [23].

Interacting components – self-organisation occurs as a result of non-linear interactions between components of the system. There are typically, but not necessarily, many such components. It is often possible to obtain the same behavioural patterns with very different components if the same rules of interaction are adhered to.

2.3 The Origins of Life

The Darwinian process relies on entities that exhibit the properties of multiplication, variation and heredity. That is they must be able to reproduce with heritable modifications. Today's genetic material takes the form of DNA that is replicated by virtue of homologous base pairing. However, DNA is not able to replicate independently but requires a complex system of chemicals to do so. A key requirement is the availability of complex biological catalysts or enzymes. For these reasons it is highly unlikely that DNA was the first example of genetic material. The dominant view is that the first genetic material took the form of ribonucleic acids or RNA, which are able to replicate in a similar way to DNA by virtue of homologous base pairing [42]. However, they also have the ability to act as a catalyst and thus the replication of RNA could potentially be catalysed by other RNA molecules. This hypothesis is further supported by the successful evolution of RNA molecules in a test-tube environment. However, this replication also requires the use of complex enzymes that could not possibly have been present in pre-biotic earth. The replication of RNA using more simple molecules remains an unsolved problem.

Today, natural replication always occurs in the context of a living cell that contains not only the DNA but all the supporting molecular machinery. The Darwinian perspective is that this molecular machinery was gradually gathered around the nude, replicating molecules driven by natural selection. However, the difficulties of producing suitable catalysts and the fact that all free living cells seem to have a minimum level of complexity has led some to question the RNA hypothesis and indeed whether homologous base pairing is a requirement for life at all [111,112]. Their alternative view is that complex systems of interacting chemicals emerged whole with the ability to sustain themselves and that the complexity of today's cells is not a result of the gradual accumulation of a supporting cast for the first replicating molecules but an inherent feature of living systems. The emergence of such a chemical system relies on the phenomenon of catalysis that was discussed in the previous section. As with the B-Z reaction the key process is autocatalysis that enables the formation of sets of chemicals that have the ability to catalyse their own reproduction. Such a chemical system is called an autocatalytic set and forms the basis of this alternative view of the origin of life.

Figure 2.3 shows a simple autocatalytic set containing two molecules. The important feature is that all the reactions in the set are catalysed by products of other reactions from within the same set and hence it exhibits the property of catalytic closure. Thus, given a supply of 'food' molecules from the environment all the products of the set can be manufactured by catalysed reactions and the chemical system is able to sustain itself. A feature of autocatalytic sets is one shared by all living organisms; metabolism. That is, molecules are consumed from the environment and used to build chemical compounds. In the process energy is dissipated during chemical reactions for example. Autocatalytic sets can thus also be categorised as a form of open thermodynamic system in which energy is constantly being exchanged with the environment. This is an important feature as it allows complex but ordered patterns of activity to be maintained

away from chemical equilibrium as was seen with the B-Z reaction in the previous section. This is a property of all free-living cells for whom chemical equilibrium corresponds to death.

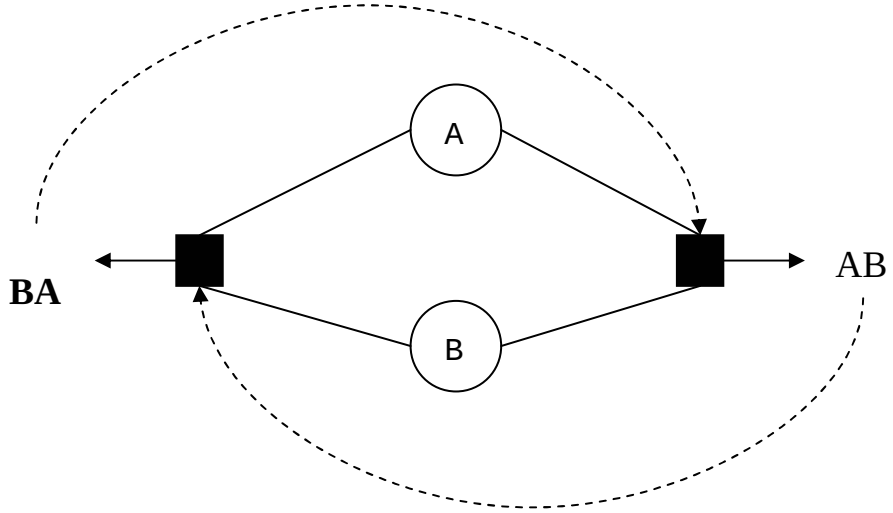


Figure 2.3: A simple autocatalytic set. Molecules A and B react to form AB and BA, which in turn catalyse the reactions that form each other. This chemical system is self-sustaining given a supply of molecules A and B [112].

The ability of autocatalytic sets to metabolise and sustain themselves makes them a candidate to act as a substrate for Darwinian evolution. However, before such a process can take hold these sets must have the ability to reproduce with variations that can be inherited by their progeny. Before reproduction can take place autocatalytic sets must grow. That is, the constituents of the chemical system must be duplicated such that division can occur leaving copies of all the chemical constituents in both resulting systems. This is a natural result of autocatalysis. In the chemical system shown in Figure 2.3, as the compounds AB and BA are formed they will catalyze the formation of yet more AB and BA, the number of catalysts are increased and hence a positive feedback cycle is formed. Thus, with a sufficient supply of molecules A and B all the constituents of the chemical system will increase in volume thus allowing division into two separate systems that can grow individually. Such a scenario also allows for the inheritance of variations. For more complicated chemical systems, the division may result in slight differences to molecular constituents of the separated systems. The resulting systems may still exhibit collective catalysis but with slightly different properties to the original system.

A prominent proponent of this viewpoint is Stuart Kauffman who has studied mathematical abstractions of systems of this kind. An intriguing result of this work is that as molecular diversity and complexity increases the emergence of autocatalytic sets is a probable rather that unlikely outcome [111]. Thus, rather than the vastly improbable accumulation of catalysts to support RNA replication the first living systems may have been a probable outcome of the chemically complex primordial soup. Autocatalytic chemical systems may have spontaneously

emerged which grew in complexity and were eventually refined into today's organisms. While this is an enticing idea, it is not without its difficulties. Autocatalytic sets must replicate as a whole i.e. there is no encoding of the information and thus the heredity is limited. In order to allow unlimited heredity the information must be compressed as in RNA or DNA molecules [42]. However, it is possible that autocatalytic sets may have contained RNA molecules whose role could subsequently be modified into information encoders. It has also been suggested that autocatalytic sets could have been "invaded" by RNA molecules to allow their replication much like today's viruses [22].

Autocatalytic sets have the potential of exhibiting all the properties required of the first living systems. They are persistently displaced from equilibrium by the flow of molecules and energy through them, energy that is dissipated in order for the system to sustain itself. As we have seen, a feature of such dissipative systems is their propensity to spontaneously exhibit ordered patterns of activity and thus in this view such spontaneous order is the foundation on which biological organisms are built, an order that must be respected by natural selection.

2.4 Developmental Biology

Today's biological organisms are the expressions of intricate developmental processes that transform a single cell into a complex dynamical form. Understanding this process is a formidable task and is the subject of much biological research. It is possible to decompose development into four sub-processes; pattern formation, morphogenesis, cell differentiation and growth [48]. Much of the research into these processes focuses on the specific details of a particular system and this work has shed much light on the mechanisms of development. A common view is that the developmental process is under tight genetic control and that the order within an organism is largely specified by the genes. However just as in physics and chemistry, developmental systems may also be influenced by the spontaneous emergence of order in self-organising systems. This is a concept that is expanded on in this section.

2.4.1 Pattern Formation

Following fertilisation, an egg undergoes a period of rapid cell division that produces a number of identical cells. The formation of a pattern from this homogenous state is an important component of development. A seminal contributor to the understanding of this process was Alan Turing who was interested in biological patterns such as the spots on a butterfly's wings and the stripes of a Zebra. Turing investigated how such spatially ordered patterns could be generated from an initially homogeneous state and postulated that the fundamental process behind these patterns was a reaction-diffusion chemical system such as the B-Z reaction [5].

Consider the simplest example of such a chemical system consisting of two chemicals A and B that are able to diffuse through some medium. Chemical A is an “activator” as it catalyses the formation of both chemicals and chemical B is an “inhibitor” as it inhibits the formation of both chemicals. Starting from an initially homogeneous state, a perturbation may occur in which a small amount of chemical A is added. The addition of this chemical will activate the production of both chemicals and hence a local peak in concentration will develop. A higher concentration of chemical A will be present due to the initial perturbation, a difference which may be exaggerated by chemical A being a more efficient auto-catalyst than a catalyst of chemical B. Consider the case in which chemical B diffuses more quickly than chemical A. Chemical B will quickly diffuse away from the local peak and inhibit the formation of A, thus creating an isolated peak of chemical A surrounded by relatively high concentrations of chemical B. This process can be repeated in other areas of the medium which are away from the influence of the inhibitor. This reaction-diffusion system thus has the ability to spontaneously create regular, stationary patterns of chemical concentrations from an initially homogeneous state. Turing’s initial work focussed on abstract models however the effect has subsequently been created in real chemical systems confirming the ability of such systems to generate patterns often witnessed in biological systems [87].

Of course, the ability of Turing patterns to form biological patterns does not necessarily mean that these patterns are actually created via such a mechanism in real biological systems. In reality a number of different mechanisms are likely to be at work which may not involve such spontaneous pattern formation. One example is the ability of cells to acquire information that is related to their position along some developmental axis. This is achieved by use of a diffusing chemical or morphogen that is emitted at one end of the axis and reduces in concentration as it diffuses away from the source. Cells along the axis will thus be subjected to different concentrations of the chemical depending on their position and the concentration has the ability to alter the expression of genes within the cell. Such a mechanism plays an important role in the well-studied development of the fruit fly *Drosophila* [26].

Whether Turing patterns complement mechanisms such as this in biological development remains an open question. Concrete proof will require identification of the specific chemicals involved, which has yet to be achieved. However, there are a number of examples that are very suggestive of the formation of Turing patterns. One such example is the angelfish [104], which has a parallel striped body pattern. However, these stripes are not laid down at an early stage of development as with the patterning of the Drosophila body pattern. As the fish grows the stripes widen as would be expected. However, as its size passes a threshold the pattern spontaneously changes and new stripes appear in between the old stripes. This strongly suggests that a chemical reaction-diffusion system is ongoing and that spontaneous pattern formation is one tool used within biological development.

2.4.2 Morphogenesis

Morphogenesis literally means the creation of form and involves the generation of a complex three-dimensional shape from a collection of cells. In the embryo, cells are held together by specific cell adhesion molecules. Changes in the shape of the embryo and cell migration are due to changes in cell adhesion and the forces generated by the cell, which are the result of the expression of different genes. As in early pattern formation, positional information derived from chemical gradients is likely to be an important component in influencing the changes in gene expression required for morphogenesis. However as with pattern formation, morphogenesis may also be influenced by spontaneous pattern formation.

It has been postulated that structuring of the body plan is influenced by spontaneous instabilities much like those that arise in chemical Turing patterns [35]. The diffusing chemicals cause certain types of cells to clump together and hence exert a force on the surrounding medium. These forces cause instabilities that may result in discontinuities that could account for limb patterns as well as other patterns such as the polygonal patterning of bird's feathers and reptile scales. The mathematical model is a form of reaction-diffusion system in which the interactions are between mechanical and chemical properties of the system rather than chemical interactions alone.

Another example of such a "mechanochemical" model is of the algae *acetabularia acetabulum*, whose morphogenesis has been studied in some detail [6,7]. The organism is more commonly known as the Mermaid's cap due to its structure consisting of a short stalk terminated by a cap of detailed structure. The Mermaid's cap is a single-celled organism containing four main components – the cell wall, the nucleus, the cytoplasm and a large fluid filled chamber called the vacuole, which exerts pressure on the cell wall to maintain its shape. The studies revealed that the cytoplasm, utilising products from the nucleus, was responsible for orchestrating the construction of several rings of leaf like structures called whorls. One important feature of the generation of this form is the calcium concentration within the cytoplasm. This concentration affects the cytoplasm's mechanical state and its resistance to deformation. Variances in calcium concentration can thus result in changes to the shape of the cell. In turn, a change in mechanical state of the cytoplasm also affects the calcium concentration. The interactions between these two features of the developing organism can be modelled in the same way as the interactions between chemicals in the BZ reaction, for example. Such a model allows the patterns that emerge through the interactions of calcium concentration and mechanical state to be investigated. These studies suggested that just as spiral patterns and concentric circles are patterns that are spontaneously generated in the BZ reaction, the whorl in the Mermaid's cap is a similar pattern that spontaneously emerges from the interactions between calcium concentration and mechanical state.

An intriguing example of the role of self-organisation in morphogenesis comes from an organism called the cellular slime mould which has a unique life cycle with two very distinct phases. When

the organism's bacterial food is in good supply the slime mould operates as a collection of independent amoeba that move around their environment engulfing the bacteria. However, when the bacteria are in short supply a very different behaviour is triggered. Starving cells periodically emit a chemical that stimulates neighbouring cells to do two things. Firstly, they emit the same chemical and then move towards the origin of the received signal. A number of amoebae in the population may initiate such a process and thus the population as a whole begins to move towards one of a number of chemical emitting sources. This reaction-diffusion process results in patterns of activity that are very much like the patterns observed in the BZ reaction and is indeed mathematically equivalent. A Petri dish covered with a layer of starving amoeba would look very similar to a Petri dish covered with a layer of BZ solution even though the details of the substances are very different. The aggregated amoebae gather together and form a multi-cellular organism that becomes progressively more complicated in form. The resulting structure consists of a base, a stalk and a fruiting body that contains spores that are able to survive the difficult conditions. When conditions are more favourable the spores are released and the life cycle repeats. The chemical patterns that spontaneously occur in a reaction-diffusion system thus form an integral component of the cellular slime mould's development.

While these examples are suggestive of a role for self-organisation in morphogenesis, surface similarities between physiochemical and biological patterns do not necessarily imply a common cause. This point is highlighted by the example of the honeycomb structure of a beehive. This structure is exceptionally well optimised to minimise the amount of wax and labour required to build it. D'Arcy Thompson proposed that the structure did not arise due to progressive improvements of the bee's manufacturing ability but rather was simply the hexagonal pattern that naturally arises in response to surface tension as bubbles are packed [11]. The bees may simply have been creating bubbles in the liquefied wax and letting self-organisation do its work. While this was an elegant solution, it turned out to be incorrect. Bees actually use a sophisticated set of genetically specified tools to perform this construction task.

Construction of a beehive could not however be carried out without interactions between the individuals that make up the colony. No single bee co-ordinates the construction of the honeycomb or has any concept of the structure that it is contributing to. Rather each bee carries out a set of simple genetically coded actions. The overall result of the actions of thousands of co-ordinated and interacting bees is the ordered honeycomb structure. Examples of such self-organisation are common in colonies of social insects and indeed the survival of the colony depends on it. Regulation of temperature in a beehive, the construction of a termite's nest and the foraging behaviour of ants are some of the many examples.

2.4.3 Cell differentiation

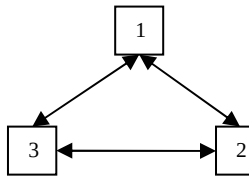
Cell differentiation leads to distinguishable cell types with specialised characteristics. The properties of each cell type are defined by different patterns of gene activity, which in turn

determine the proteins produced within the cell. Each cell manufactures a basic set of proteins that enable it to perform the basic cell functions. However, different cell types manufacture additional so-called "luxury" proteins that allow the cell to carry out specialised functions. The pattern of gene expression that allows the creation of these proteins may be influenced by external factors such as chemical gradients as we have already seen. Such external influences may cause a particular gene to be expressed or inhibited for example. However, the effect of such a change may not simply be the presence or absence of a single protein but be more wide-ranging.

Genes have the ability to regulate each other. In effect they can turn each other 'on' and 'off' by coding for transcription factors that bind to control regions defining whether or not a gene is expressed [20]. This effect allows the formation of complex regulatory networks. In the simplest case a sequence of DNA might encode two genes with the ability of inhibiting each other. Thus, in one cell type one of the genes may be active and in an alternative cell type the other gene may be active. Vastly more complex genetic circuits such as these are evident in the cell types of today's organisms. *Drosophila* has approximately 15,000 active genes, an estimate that rises to anything between 30,000 and 120,000 for humans. Genomic systems of this size create vast interconnected genetic regulatory networks with their own intrinsic dynamics.

It is possible to get a grasp on the nature of these dynamics using an abstract model called a Boolean network. This idealisation allows a genotype to be thought of as a network of nodes and directed edges. Each node represents a particular gene that can be either on or off and each edge represents a regulatory interaction. Thus, an arrow leading from one node to another indicates that the former either inhibits or promotes the expression of the latter. A given gene is typically regulated by a number of other genes, each of which can be in one of two states. Thus with L regulating inputs there are 2^L possible input configurations, some of which will result in the expression of the gene and others will not. The rules governing this process can be generated using familiar Boolean logic. For example, an AND-rule could be used in which the gene is expressed if all the regulating inputs are present. The OR-rule could also be used in which the gene is expressed providing at least one of the inputs is present. A simple network of this type is shown in Figure 2.4.

2	3	1
0	0	0
0	1	0
1	0	0
1	1	1



1	2	3
0	0	0
0	1	1
1	0	1
1	1	1

1	3	2
0	0	0
0	1	1
1	0	1
1	1	1

1	2	3	1	2	3
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1

t **t + 1**

Figure 2.4: A simple Boolean network. The formation of each of the 3 molecules is regulated by the other 2 molecules according to the given rule tables. The rule table of molecule 1 is the Boolean AND function whereas the rule table of molecules 2 and 3 is the Boolean OR function. The dynamics of this network can be captured by the state transition table shown on the right. From each state of the network at time *t*, this table gives the next state at time *t + 1*.

These Boolean networks are a discrete approximation to the dynamics of a genetic regulatory network. Each node is simultaneously updated and as a result the input configurations of each node may be altered. Iterative updates will therefore result in the network traversing a sequence of network states. The dynamics of the network can be captured in the form of the state transition table shown to the right of Figure 2.4. Starting in any of the 8 possible initial states the cycle of states the network passes through can be ascertained by consulting this table. The state cycles for this network are shown in Figure 2.5.

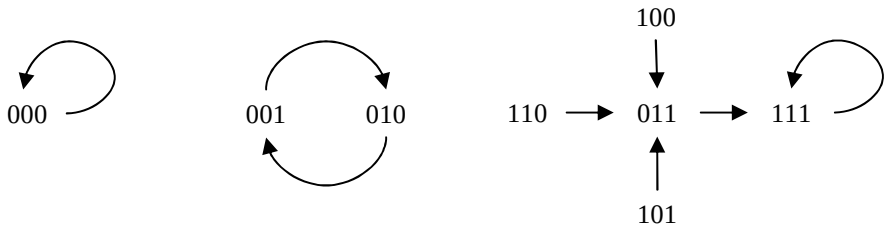


Figure 2.5: State cycles for the Boolean network shown in Figure 2.4. All 8 initial states settle on to one of three final behaviours. Two of these are fixed points at 000 and 111, the third is a limit cycle endlessly cycling between two states.

These state cycles reveal that there are only three possible final behaviours of the network. Two of which are fixed points, that is a single state at which the system remains unless further

perturbed and the third is a limit cycle, which is a subset of states that the system continually cycles between. In the two leftmost state cycles shown in Figure 2.5, any perturbations will result in the system being knocked in to a different state cycle and hence a different final behaviour. These state cycles are thus unstable. In contrast, the third state cycle reveals a far more stable behaviour that is characterised by the presence of an attractor. When the system is initialised in any of five possible states the natural dynamics result in the system converging on the fixed point attractor corresponding to the expression of all 3 genes. This attractor has a basin of attraction consisting of four states i.e. half the state space of the system and is thus relatively stable, changing any individual '1' to a '0' will perturb the system away from the attractor but not away from the basin of attraction and thus the system will quickly progress back towards the fixed point at which all genes are expressed.

This network is very simplistic and its dynamics are heavily influenced by the specific rule tables and interactions that were chosen. It is possible, however, to study more generic properties of these systems by assigning the rule tables and interactions at random. Such a network is called a random Boolean network (RBN) [111] and forms the basis of one of the mappings developed in chapter 4 of this thesis. An RBN is characterised by two variables, N and K. The former indicates the number of nodes in the network and the latter the number of regulatory inputs per node. With K randomly chosen inputs there are 2^K possible Boolean rule tables, one of which is chosen at random for each node. Such a system allows large numbers of different networks to be investigated with different settings of the N and K parameters.

Extensive investigations of this type have revealed that the value of the K parameter is crucial in defining the nature of the systems state cycles [111]. When K = 1 and each gene is only regulated by a single input, small sub-networks of genes tend to form and the activity quickly settles into attractors consisting of a very small number of states. Many of these are fixed point attractors corresponding to chemical equilibrium and thus this class of network does not exhibit the more complex chemical patterns of interest. At the other extreme, when K = N and every gene is regulated by every other, the networks are maximally disordered. In effect, the next state of the network is a random choice among the 2^N possible states. However, even in this case attractors are present. The number of attractors is approximately equal to N/*e*, where *e* is 2.7182 i.e. the base of natural logarithms. Thus, a system containing 80,000 genes would result in approximately 30,000 attractors. However in contrast to K=1 networks, these attractors consist of very long state cycles that could not be completed in any reasonable time frame and are thus not biologically realistic.

Thus, one extreme of the K parameter leads to an inflexible network that tends towards chemical equilibrium and the other extreme creates a maximally disordered network. However, interesting properties emerge when the K parameter is set to 2 causing each gene to be regulated by two other molecules. In this case the number of attractors is approximately equal to the square root of

the number of nodes, N. Thus, an 80,000 gene network would result in approximately 300 attractors, far fewer than for K=N networks. In addition the cycle lengths of the attractors are relatively short; on average they are also approximately equal to the square root of N. Thus, there is very large contraction of the volume of state space that the system occupies – of the possible $2^{80,000}$ states the system is compressed into around 300.

However, an important question remains - how likely is it that in real organisms genes have this few regulatory inputs? Developments in experimental biology allow us to examine patterns of gene regulation in real organisms. These investigations reveal that the number of regulatory inputs is typically relatively low but not necessarily 2. However, another important fact emerges. Regulated genes and many other biochemical processes are governed by canalising functions. That is, at least one of the regulatory inputs has a value that alone suffices to guarantee that the regulated gene will have a given value. For example if a given gene is regulated by two different molecules, the presence of one of the molecules may be sufficient to guarantee that the gene will be expressed independently of the other molecule. In the Boolean idealisation the OR function is an example of a canalising function. The presence of any of the inputs alone is sufficient to guarantee that the output will also be present. The significance of this is the effect canalising functions have on the dynamics of the genetic regulatory network and the random Boolean network model. When the rule tables governing the next state of a node are chosen to be canalising functions, the networks exhibit ordered behaviour. Thus even if the number of regulatory inputs is greater than 2, canalising functions force the network into an ordered state.

While the idealisations made in the RBN model may limit its relevance to real chemical systems which exhibit graded rather than digital responses, it seems likely that the dynamics of self-organising genetic regulatory networks impose some constraints on patterns of gene expression. Evolution may not have a free hand in tailoring genetic expression. Thus in cell differentiation, pattern formation and morphogenesis the same pattern emerges. Darwinian evolution may not be able to create arbitrary form, characteristic or behaviour but must work under the constraints imposed by the intrinsic dynamics of a self-organising process. This change in perspective has the potential of wide ranging impact on the biological sciences that may ultimately demand a new theory of biological order incorporating both natural selection and self-organisation. However, an immediate consequence is on our view of evolutionary dynamics.

2.5 Evolutionary Dynamics

For many years the overriding metaphor used to visualise evolutionary dynamics has been the fitness landscape introduced by Sewall Wright [108]. In this metaphor, an evolving population is thought of as moving on a landscape that typically has a number of peaks and troughs. The peaks represent genotypes that are well adapted to their environments i.e. that have a high fitness, and the troughs represent genotypes of relatively low fitness. Each location in this abstract space represents a unique genotype and hence it is termed genotype space. Events such as mutation and

recombination modify the genotypes and hence change their location in the space. Changes that result in a genotype of higher fitness result in an up-hill movement and tend to be selected for and preserved in the population. The continual pressure of natural selection thus has the effect of pulling the population towards the peaks.

This metaphor has been very successful in aiding our understanding of evolution. However, an evolutionary process operating on such a landscape suffers from a major drawback – the presence of local optima. Under the influence of natural selection, a population will climb any hill in its vicinity. However, it may turn out to be a foot hill rather than a mountain and the population can become isolated in these locally optimal but globally sub-optimal regions of the landscape. A landscape containing such local optima is shown in Figure 2.6. As the population climbs the hill it converges into a small area of genotype space resulting in a population of very similar genotypes. Sexual recombination thus has little effect, producing a genotype that also resides in this isolated area of genotype space. Mutations allow small changes to the genetic constitution of the population but every step leads downhill from the local optimum. A higher mutation rate allows larger jumps to be made in the landscape, which may allow small valleys to be negotiated but large hopeful leaps are unlikely to find genotypes of higher fitness. In addition, as the mutation rate increases further the disruption becomes too large for natural selection to withstand and the population loses its current optima due to the onset of an error catastrophe [52].

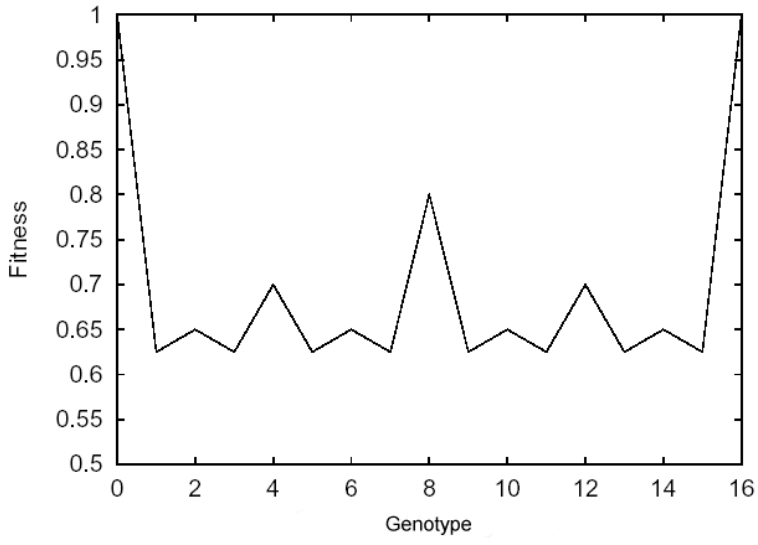


Figure 2.6: An example fitness landscape containing local optima. Even numbered genotypes are locally optimal but only genotypes 0 and 16 are also globally optimal.

Local optima thus have the potential to result in an end to evolutionary innovation, an outcome which does not resonate with the seemingly boundless creativity of natural evolution. The above metaphor is of course a simplification. One major omission is the fact that a biological species never evolves in isolation on a static landscape but in the context of a complex ecosystem. Each

species within this ecosystem is attempting to climb its own hill and in so doing deforms the landscapes of the other species. Such co-evolution fundamentally changes the picture of adaptive, hill-climbing walks on fitness landscapes. However, even without such complications the emerging appreciation of self-organisation suggests that the fitness landscape metaphor may be incomplete.

2.6 Neutrality

The fitness landscape metaphor outlined above maps a fitness value directly to the genotype, a process that intentionally disguises the complexities of the intermediary processes. The overall genotype-fitness mapping can be decomposed into two separate mappings; the genotype-phenotype mapping and the phenotype-fitness mapping as shown in Figure 2.7. This separation recognises the fact the genotype is expressed in a phenotype, which in turn is subject to natural selection. Thus, it is the phenotype that is assigned a fitness that may be a measure of its reproductive success for example. Both these intermediary processes have the potential of introducing their own particular properties into the overall genotype-fitness mapping. One of these properties is *neutrality*, which arises due to the fact that some changes to the genotype may have little effect on the phenotype or that some phenotypic changes may not effect its survival or reproductive success and hence fitness.

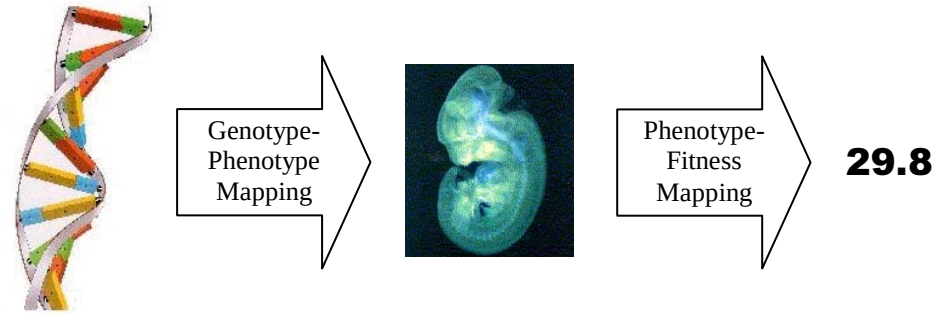


Figure 2.7: The decomposition of the mapping from genotype to a fitness value into two separate mappings; the genotype-phenotype mapping and the phenotype-fitness mapping. Both of these processes may introduce properties to the overall mapping, one such property is neutrality.

At a genetic level, neutrality is common in biological organisms. The genotypes of many organisms contain large amounts of DNA that is not transcribed and plays no obvious role in encoding proteins; it has thus been termed junk DNA. Whether this is truly "junk" is an open question, it may play some structural role for example. However, much of this DNA has become far removed from the regions of DNA that code for proteins. The possibility of it affecting the phenotype and its fitness by generating proteins is thus often remote. The genetic code itself also exhibits neutrality; the amino acids making up a protein are specified by codons composed of three nucleotides that can be one of four types. There are thus 4^3=64 possible codons, 61 of which

code for amino acids. However, there are only 20 different amino acids and thus some codon mutations are neutral as they do not change the amino acid or resulting protein. Larger scale neutrality, however, arises at a higher level than the genetic code. The neutral theory of molecular evolution proposes that the majority of changes at the molecular level are the result of the random drift of genotypes as opposed to changes that have been selected for [53]. Many natural amino acid replacements are neutral with respect to the function of the protein. In addition, such changes need not be strictly neutral in that they have absolutely no effect on fitness, rather they must be neutral with respect to natural selection in a dynamic environment. Ohta claimed that small differences in fitness are not visible to selective pressures and thus a band of fitness's around the current optimum are *effectively* neutral [114].

However, the potential scale of neutrality in biological systems becomes apparent with the consideration of self-organisation. In this chapter we have seen a number of examples that suggest a role for self-organisation at many different levels in biological systems. In these systems many different initial states give rise to the same system behaviour. For example, many different genetic configurations in the random Boolean network give rise to the same pattern of gene expression. Changes to the genotype influence the dynamics of the self-organising process but many changes may not be sufficient to perturb the system away from the basin of attraction associated with the current attractor. The attractors define the observable behaviour of the system, i.e. the phenotype, and thus the basin of attraction defines a set of genotypes that give rise to the same phenotype. An effect of self-organisation is thus to introduce large-scale neutrality into the genotype-phenotype mapping.

In many cases neutrality is also a feature of the mapping of these phenotypes to fitness values as different phenotypic features may not lead to an appreciable difference in an organism's survival or reproductive prospects. Despite these many potential sources of neutrality, it is not currently emphasised in the dominant models and theories of evolution. Nonetheless, its consequences have the potential to be wide ranging. In the following section one such consequence is highlighted that has the potential of alleviating the problem of local optima; neutral networks.

2.7 Neutral Networks

The presence of large sets of genotypes that map onto the same phenotype and ultimately the same fitness adds a new dimension to the fitness landscape metaphor. In addition to climbing or descending a fitness gradient, a population also has the ability to drift at the same fitness level. Such neutral drift opens up the possibility for an evolving population to escape areas of genotype space from which no immediate progress is possible. Rather than become trapped at local optima, the population may be able to drift to new areas of the landscape that allow continued progress. In order to increase the probability of such an occurrence it would be beneficial to allow extended periods of neutral drift. Isolated neutral mutations allow small movement in genotype space but in many cases this may not sufficiently increase the probability of finding higher-fitness

phenotypes. However, a series of neutral changes allows access to far larger areas of genotype space which increases the probability of discovering areas from which new fitness peaks may be reached.

Maynard-Smith analogised this process with a popular word game in which one word is transformed into another via a series of individual letter changes that leave a valid word at each stage [43]. For example, WORD can be transformed into GENE by the series of transformations WORD – WORE – GORE – GONE – GENE. Valid words can be thought of as functional proteins and the letters as the constituent amino acids, which are ultimately genetically defined. Thus if each replacement were neutral i.e. if each protein was functionally equivalent, then significant movement would be possible in genotype space without changing the fitness. All the genetic primitives have been changed but the functional phenotype is unaltered. In order to allow neutral drift such as this therefore, it is not only important for the phenotypes to be represented by sets of genotypes but also that these sets are connected by neutral genetic changes forming so-called *neutral networks* [85].

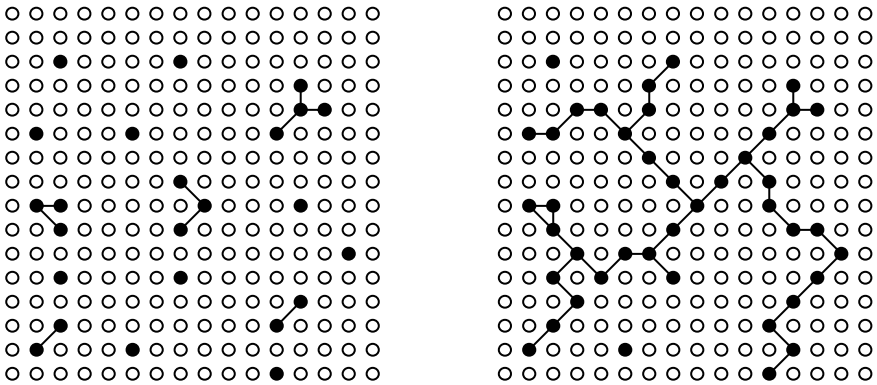


Figure 2.8: The formation of a giant component in genotype space. Shaded circles represent the genotypes within a neutral set and the edges represent neutral transitions between genotypes. When the ratio of edges to vertices is low (left of figure) the genotypes are isolated into small subsets. However as this ratio increases beyond 0.5, random graph theory predicts the formation of a giant connected component that contains the majority of the genotypes (right of figure). This component corresponds to a neutral network that can percolate throughout genotype space.

The likelihood of the formation of connected networks such as these can be investigated through the use of mathematical models derived from random graph theory. A mathematical graph is composed of a set of vertices that are connected by edges. In this case, each vertex can be thought of as one of the genotypes in a particular neutral set and edges can be thought of as representing neutral transitions between genotypes. Thus, subsets of vertices are formed that are composed of genotypes reachable from each other via neutral genetic modifications i.e. that represent neutral networks. Studies of these abstract graphs have revealed that as the ratio of edges to vertices passes a threshold of 0.5, a giant component is formed that includes a large majority of the

vertices [71]. Equivalently, as the average number of neutral transitions per genotype passes 0.5 the number of genotypes in the largest subset rapidly increases from a relatively small number to encompass a large majority of the genotypes in the neutral set. With sufficient neutrality therefore, large neutral networks are formed that percolate throughout genotype space. This process is visualised in Figure 2.8.

The percolation of neutral networks allows a population to move extensively through genotype space. Such extended movement greatly increases the number of accessible phenotypes and thus may be an important component in allowing continued evolutionary innovation. The diffusion of a population of a neutral network is visualised in Figure 2.9. The figure depicts genotype space sub-divided into a number of neutral networks. At stage 1, the population has recently discovered a new neutral network which it gradually drifts along until at stage 8 it comes within proximity of a neutral network of higher fitness. A single individual discovers this neutral network at stage 9 and the pressures of natural selection quickly establish the population there.

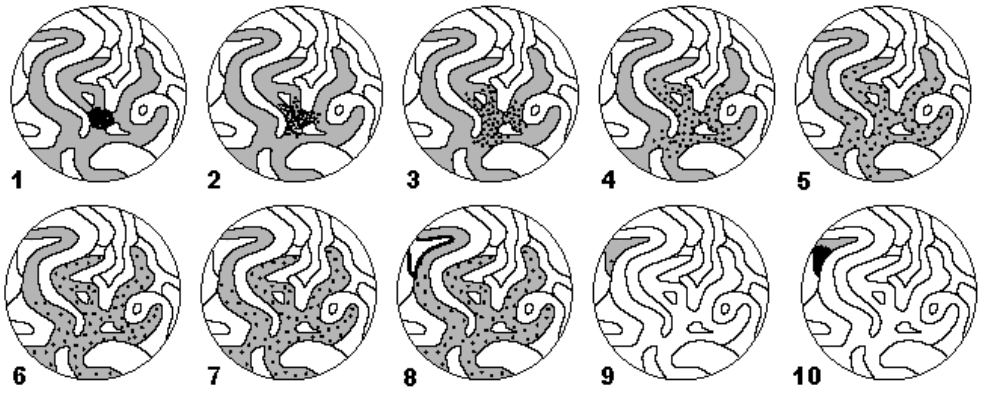


Figure 2.9: Visualisation of a population's diffusion on a neutral network. At stage 1, the population has recently discovered a higher fitness neutral network (shaded). The population gradually drifts along this network until at stage 8 it comes within proximity of a neutral network of higher fitness. This neutral network is first discovered at stage 9 and the population is quickly established there.

In this view therefore rather than continuous gradual increases in fitness, evolutionary dynamics are dominated by periods of little or no gain as a population moves along a neutral network punctuated by rapid gains as a neutral network of higher fitness is discovered. These punctuated equilibrium dynamics were first proposed to describe sudden morphological changes in the palaeontology record [106]. While the validity of this punctuated equilibrium theory remains in question, the associated evolutionary dynamics are commonly observed. Examples include the evolution of bacterial colonies [101] and the evolution of biopolymers. The latter have been widely studied and the impact of neutral networks on the evolution of biopolymers analysed in

depth. While some of this work has involved the study of protein molecules [2,3,82], the primary focus has been RNA molecules and is described in detail in the following section.

2.8 RNA Folding

RNA molecules are biopolymers composed of four units called nucleotides. These units are the four different bases – A for adenine, U for uracil, C for cytosine and G for guanine. The bases are subject to similar base-pairing rules as were discovered by Watson and Crick for DNA molecules, adenine pairs with uracil (A-U) and guanine with cytosine (G-C). The base pairs allow segments of a sequence to bind with other segments within the same sequence, which in turn causes the molecule to fold back on itself into a three-dimensional structure. This structure defines the chemical interactions of the molecule and hence its functional role. RNA molecules are highly evolved and crucial components of today's organisms playing a wide variety of functional roles. Their evolution, however, need not take place exclusively within an organism but can also take place in isolated environments. Such *in vitro* evolution aims to produce RNA molecules that play certain functional roles. It is possible as the base-pairing also allows RNA molecules to replicate via a complementary negative sequence in a similar way to DNA. Thus, in these isolated environments the RNA molecule can act as both genotype and phenotype.

In this scenario the genotype-phenotype mapping reduces to the folding of an RNA molecule from its primary nucleotide sequence into a specific structure. This structure can be observed at different levels of resolution. The final structure of the molecule, called the tertiary structure, is geometrically defined in terms of distances and coordinates. However, it can be decomposed into secondary structures that provide the scaffolding for the tertiary structure. These secondary structures refer to the topology of contacts that arise from specific base-pairing as shown in Figure 2.10.

Stacking pair Hairpin loop

Figure 2.10: Several building blocks of RNA secondary structures. The secondary structure defines the topology of contacts that arise through specific base-pairing within the primary nucleotide sequence.

Not only is the secondary structure very useful for interpreting molecular function, selection pressures also become observable at this level of resolution in terms of conserved base pairs. For these reasons the secondary structure can be usefully considered as a phenotype. An important consequence of this choice is the fact that the phenotype is computable from the nucleotide

sequence. Algorithms have been developed that allow the rapid computation of the secondary structures [34] that can also be implemented in parallel [32]. These algorithms allow extensive studies to be made that reveal a number of properties of the genotype-phenotype mapping, many of which are independent of the particular algorithm that is used [57].

One study of this kind involved the exhaustive folding of all G-C and A-U sequences up to a chain length of 30 [135]. This gave a complete picture of the genotype-phenotype mapping for sequences of this type. This data was compared to that derived from analytical techniques in which random graph theory was used to model the mapping [9,10] and that from the statistics of large samples derived by folding random sequences of fixed chain length [85]. In all these cases a number of common properties emerged which are highlighted below [66,79-81,83,84,133,134]:

2.8.1 Large scale neutrality

The primary nucleotide sequence is made up of chains of one of four bases, these chains can become relatively long and thus the size of genotype space becomes very large. For a chain length of 30 the number of sequences and hence the size of genotype space is 4^{30} or approximately 10^{18}. However the number of secondary structures that each sequence maps onto is very much smaller. The analysis revealed that an upper bound to the number of secondary structure shapes is given by [9,33]:

$$S_n = 1.48 \times n^{-3/2}(1.85)^n \qquad \textit{Equation 2.1}$$

Where n is the length of the nucleotide sequence and S_n is the number of shapes. Thus, for a chain length of 30 the number of shapes is approximately 917,665 - a vastly smaller number than the number of sequences. On average approximately 10^{12} sequences map onto each shape and there is thus very large scale neutrality.

2.8.2 Common structures

The sets of sequences folding into each possible shape are not of equal size - the sequences forming some shapes occur very much more frequently than others in genotype space. These shapes are termed common structures, which can be straightforwardly defined as those that are represented by a greater or equal number of sequences than the average shape. Thus for the 4 base alphabet and chain length of 30 described above, common structures are those that are represented by at least 10^{12} genotypes. An exhaustive analysis of restricted sequences consisting of only G and C bases revealed the extent of this skew in distribution [135]. For 2 base sequences of length 30 a total of 218,820 different shapes are produced. Only around 10% of these are common structures, however, over 93% of sequences folded into one of these common structures leaving only around 7% of genotype space for the rare structures. As the chain length increases these differences are further emphasised as the number of common structures decreases but the fraction of sequences forming those structures tends to unity. Thus, with sufficiently large chains

almost all sequences fold into a small number of secondary structures. From these results it is clear to see that the evolution of RNA molecules is dominated by the common structures and the rare structures play little role.

2.8.3 Neutral networks

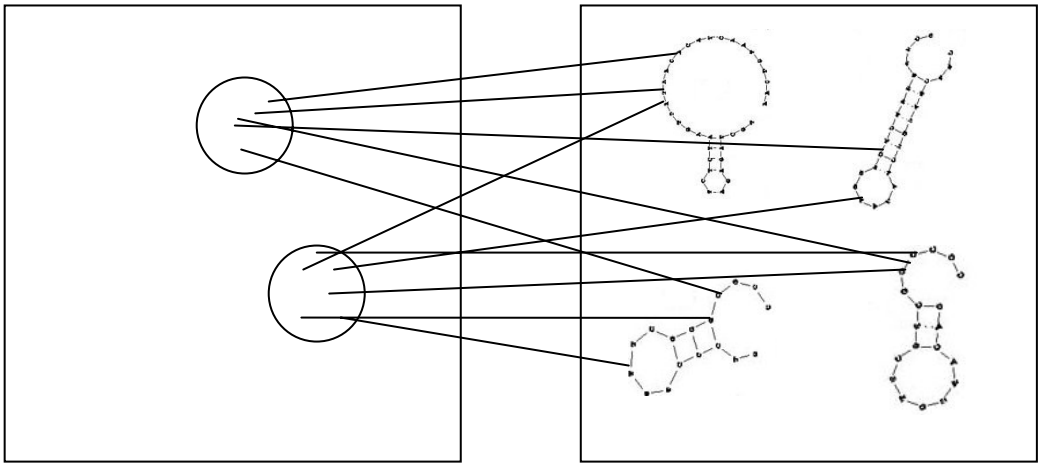
As was described earlier, neutral networks are formed by sets of genotypes that are connected by neutral genetic modifications. In this case, RNA molecules replicate asexually and thus the neutral neighbourhood is defined solely by mutation events. The analysis focuses on the most probable of these - single-point neutral mutations. The common structures were found to have a high fraction of neutral neighbours i.e. many mutations of a sequence folding into a common structure would result in a sequence that also folded into the same structure. The random graph models predicted a critical threshold of neutral neighbours at which the neutral set would be fully connected [9], this is shown below:

$$\lambda^* = 1 - \alpha^{1/1-\alpha} \qquad \textit{Equation 2.2}$$

Where α is the number of bases in the alphabet and λ^* is the threshold of neutral neighbours at which a fully connected neutral set emerges. Thus for a 2-base alphabet such as the study involving only G and C bases, the threshold of neutral neighbours is 0.5. As the alphabet grows to the 4-bases of natural RNA molecules, the threshold drops to 0.37. This requirement is satisfied by all common structures and thus large, fully-connected neutral networks are formed that percolate throughout genotype space. An evolving population is thus able to move very large distances in genotype space through neutral drift. In contrast, the rare structures do not meet these requirements and the neutral networks associated with them are small, fragmented and isolated.

2.8.4 Shape space covering

An important consideration when an evolving population is drifting along a neutral network is the likelihood of encountering the neutral networks associated with other, potentially higher fitness, phenotypes. That is, the accessibility between the neutral networks associated with each of the common structures. In the RNA mappings, it was found that each of the common structures could be found within a small radius of any arbitrarily chosen sequence [132]. Thus, from any location in genotype space only a relatively small distance needed to be travelled along a neutral network in order to encounter each of the common structures. This is visualised in Figure 2.11.



Genotype space Phenotype space

Figure 2.11: Shape space covering in the RNA folding landscapes. A very large number of genotypes fold into each common secondary structure. From any arbitrary location in genotype space only a small fraction of the space needs to be searched in order to find sequences that fold into each of the common structures.

The radius from any arbitrary sequence within which all common structures are discovered was termed the covering radius. For a sequence of length 30 the covering radius was found to be 4. That is, all common structures were within 4 mutations of any arbitrary sequence. For sequence lengths of 100 the covering radius rose to 15 mutations. These statistics suggest that new common structures would be readily found as a population drifted along a neutral network. This hypothesis has been supported by performing random neutral walks [65]. These experiments simulated an individual drifting along a neutral network and assessed its single-point mutation neighbourhood along the way. The number of different secondary structures discovered in this neighbourhood gave a measure of the accessibility between neutral networks. The results revealed that the number of secondary structures discovered increased linearly; every neutral mutation gave access to similar number of new structures. In addition, the innovation of new secondary structures did not show signs of slowing but continued for the 1000 neutral steps that were taken.

The results suggest therefore that long periods of neutral drift are not required before encountering previously undiscovered phenotypes but rather that there is a *constant innovation* of phenotypes along the neutral walk. Consequently, neutral drift does increase the probability of discovering higher fitness phenotypes. This probability may be further increased when a population rather than an individual inhabits the neutral network. Not only does this increase the effective neighbourhood that is being assessed but the population tends to split into a number of sub-populations that diffuse independently along the network. These so-called molecular quasi-species thus allow different regions of the neutral network to be explored by the same population [64].

2.9 Summary

In this chapter, the background and motivation for subsequent work has been described. The key points are highlighted below:

- Natural systems have a propensity to self-organise and evolution may have always worked within the context of such self-organisation. Evidence was presented that suggests a role for self-organisation in the origins of life as well as current developmental biology.
- Self-organisation constrains evolution. From a large number of initial states, a self-organising system naturally converges towards one of a relatively small number of preferred behaviours. Evolution cannot create arbitrary form, characteristic or behaviour but must work within these constraints.
- The coupling of self-organisation and evolution changes our perception of evolutionary adaptation. Many genotypes result in the same phenotype and thus many genetic changes are neutral. Evolutionary dynamics are dominated by periods of stasis punctuated by rapid gains.
- With sufficient neutrality, the sets of genotypes mapping onto a given phenotype are connected into *neutral networks* that percolate throughout genotype space. Neutral drift on these networks allows a population to escape from locally optimal but globally sub-optimal regions of the fitness landscape.
- Extensive studies of the fitness landscapes arising in the evolution of RNA molecules reveal that such neutral drift allows constant innovation of new phenotypes.

A primary aim of this work is to design self-organising genotype-phenotype mappings that share the key properties of natural genotype-phenotype mappings as evidenced by RNA molecules. This work is described in later chapters. Firstly however, the following chapter reviews work within artificial evolution that has investigated the effects of neutrality.

Chapter 3

Neutrality in Artificial Evolution

3.1 Introduction

A common practice in artificial evolution is to encode potential solutions in such a way that changes to the genotype directly affect the current phenotype or solution. As an example, consider the problem of evolving the topology of a network together with the locations of its constituent nodes. In this example there are a fixed number of nodes that may be placed anywhere on a two-dimensional grid and may be connected by a fixed number of links. A typical encoding would specify the coordinates of each node together with indexes to the two nodes that each link connects as shown in Figure 3.1.

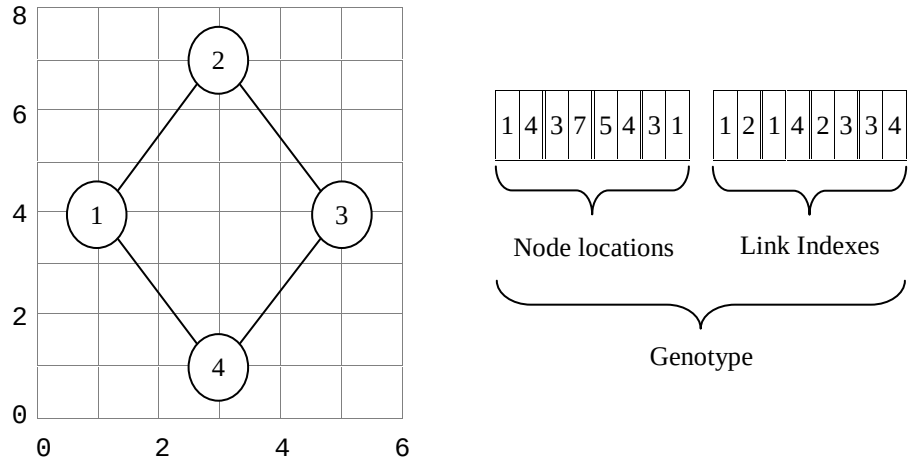


Figure 3.1: An example of a direct encoding. The phenotype consists of a number of nodes connected by links. The genotype directly specifies the location of each node together with link indexes and thus every genetic change results in a phenotypic change.

Thus in this example, the network can be fully specified by a series of variables and a change to any one of the variables generates a corresponding change in the network. The prevalence of direct encodings such as these in artificial evolution is a natural consequence of the dominant view of evolution that has been inherited from biology. This view is epitomised by the fitness landscape metaphor introduced in the previous chapter in which each genotype is directly assigned a fitness value and the intermediary genotype-phenotype and phenotype-fitness mappings are deemphasised. This lack of emphasis has also been a feature of artificial evolution. This thesis adds to the work that is beginning to change this situation.

As was seen in the previous chapter, a common feature of biological mappings is the presence of large scale neutrality. Over recent years this knowledge has begun to filter through to the artificial evolutionary community and several studies have been performed to assess the impact of neutrality. These studies are the subject of this chapter. Although much of this work is relatively recent, neutrality has been a feature of artificial evolution since its inception. One reason for this is that many encodings that are not explicitly designed to contain neutrality do so implicitly.

3.2 Implicit Neutrality

The encoding shown in Figure 3.1 was presented as an example of a direct encoding, however, closer inspection reveals that the encoding results in significant neutrality. Given that the nodes and links are equivalent they can be exchanged without affecting the phenotype. For example, node 1 could be exchanged with any of the other nodes in the network without affecting the overall network design. Such an exchange would result in a shuffling of the genetic information and hence different genotypes that represent the same phenotype as shown in Figure 3.2.

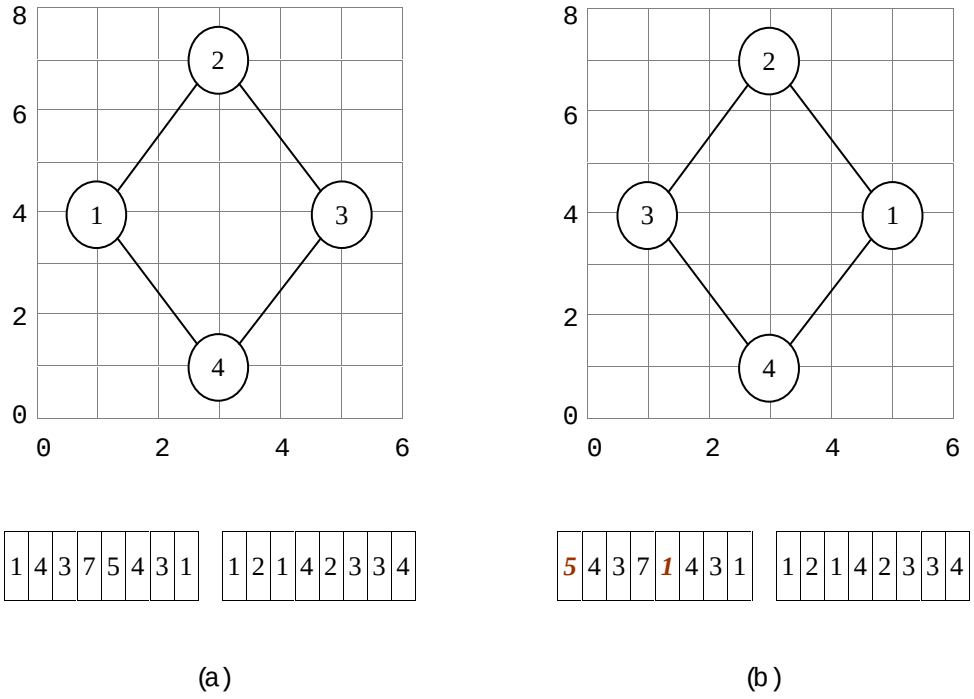


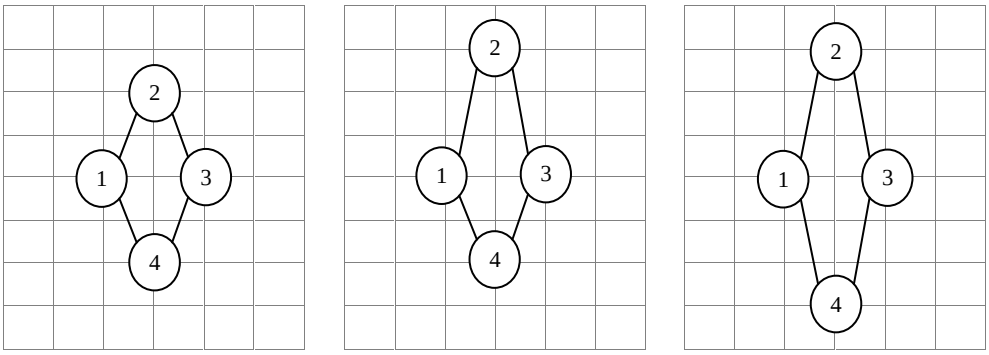
Figure 3.2: Implicit neutrality in the encoding introduced in Figure 3.1. (a) The original network and associated genotype. (b) Nodes 1 and 3 have been exchanged resulting in an equivalent network but a different genotype, the modifications are indicated in italics. Equivalent networks such as these result in significant neutrality in this encoding.

In addition to node substitutions, each of the links can also be exchanged with one another without affecting the overall phenotype. In total there are 4! = 24 neutral node configurations

each of which has 4! = 24 neutral link configurations. Thus, 24*24 = 576 genotypes represent the same phenotype. Further neutrality may result from the choice of fitness function, which defines the phenotype-fitness mapping. As an example, consider a fitness function that calculates the inverse of the maximum distance that any of the locations within the grid are away from their nearest node. This may be done so that the geographical coverage of the network can be maximised for example. One way to approximate this distance would be to sum the horizontal and vertical distance between one location and another. Thus, the network shown in Figure 3.1 would have a fitness of 1/4 as each corner of the grid is a distance of 4 away from the nearest node and no other locations are at a greater distance. However, there are a number of other networks that also produce the same fitness. For example if the locations of nodes 2 and 4 are held constant, there is relative freedom in the placement of nodes 1 and 3 in order to maintain the same fitness.

It can be seen therefore that typical choices of encoding and fitness function can result in the introduction of significant neutrality and thus without intention neutrality is likely to have been a feature of many artificial evolution experiments. However, neutrality alone is not sufficient to allow the beneficial properties highlighted in the previous chapter in which the discovery of new phenotypes was allowed for through neutral drift. This can be seen by considering the neutrality in the genotype-phenotype mapping of this example. As we have seen, 576 genotypes encode for the phenotype shown in Figure 3.1. However, these do not form a connected neutral network that would allow neutral drift and the discovery of new phenotypes. Exchanging nodes 1 and 3 required two simultaneous changes to the genotype and other exchanges would require even more improbable events. An exchange of nodes 1 and 4 for example would require 4 simultaneous changes to the genotype. These networks would thus tend to be represented in genotype space as a number of single, isolated genotypes that do not form neutral networks. Higher mutation rates may allow some connectivity within the neutral set but the connectivity would likely remain sparse.

In contrast, the implicit neutrality resulting from the phenotype-fitness mapping does have the potential to allow beneficial neutral drift as shown in Figure 3.3. Considering only single changes to the genotype, no immediate increase in fitness is possible from the phenotype shown in Figure 3.3(a). However, the neutrality resulting from the phenotype-fitness mapping allows a stage-setting neutral mutation to be made and a subsequent adaptive mutation that generates the higher fitness phenotype shown in Figure 3.3(c). This neutrality exhibits two properties that enable it to play a beneficial role. Firstly, the neutral genotypes are connected by probable changes to the genotype. In this case, single-point mutation was the only genetic operator considered. However, other genetic operators may result in different neighbourhood relationships between the neutral genotypes. Secondly, neutral drift increases the probability of the discovery of new phenotypes. In this case, the neutral mutation moved the genotype to within a single mutation of a higher fitness phenotype whereas previously two simultaneous mutations were required.



Fitness = 1/5 (a)

Fitness = 1/5 (b)

Fitness = 1/4 (c)

Figure 3.3: An example of neutral mutation allowing a higher fitness phenotype to be discovered. The fitness is calculated as the inverse of the maximum summed horizontal and vertical distance from any point in the grid to the nearest node (a) Each corner of the grid is a distance of 5 away from the nearest node and thus the network's fitness is 1/5 (b) Single changes to the original phenotype cannot increase its fitness, however, a neutral change is made to node 2 resulting in only the bottom left and right corners being a distance of 5 away from the nearest node (c) The previous neutral change allows a subsequent change to node 4 that results in a higher fitness phenotype.

As evidenced by this example, beneficial neutrality may have been an inherent part of many artificial evolutionary systems and some of the work reviewed later in this chapter aims to assess the extent to which this is true. Firstly however, another method of implicitly introducing neutrality into artificial evolutionary systems is discussed; self-adaptation.

3.3 Self-adaptation

A common feature of many evolutionary algorithms is the ability to self-adapt the genetic operators such as mutation and recombination [76]. A given mutation or recombination rate may be appropriate for the specific characteristics of one problem but may be less appropriate for alternative problems. In addition, it may be desirable to modify these rates during a single evolutionary run – a technique that is used to good effect in the simulated annealing algorithm [103]. In order to achieve this, the parameters controlling the genetic operators can be encoded in the genotype along with the solution representation. In this way, the genetic operators themselves are subject to the same selection pressures as the solution and thus beneficial operators are evolved along with the solution. This technique is an inherent part of evolutionary strategies in which a number of strategy parameters are encoded that control the mutation rate [28]. Although this is the most prominent example, such self-adaptation has also been employed in all the other main paradigms of evolutionary computation. For example, the self-adaptation of both mutation probabilities [115] and crossover operators [36] has been explored in genetic algorithms and self-adaptive crossover operators have been explored in genetic programming [77].

A feature of such self-adaptation is that changes to the encoded operator parameters are neutral with respect to the phenotype. These parameters have no direct effect on the phenotype but alter the dynamics of the search, which may in turn contribute to the discovery of new phenotypes. This point was emphasised by Toussaint and Igel who claim that this is the main advantage of neutral encodings [60]. As an example, consider a self-adaptive mutation-based evolutionary algorithm in which a single mutation per genotype is performed for each mutation event. The evolutionary algorithm may quickly become isolated at a local optimum where every single mutation leads to a less fit phenotype. However a mutation to the parameter controlling the mutation rate could lead to the number of mutations per genotype being increased, which may allow the local optima to be escaped as larger steps could then be made in genotype space.

Neutral mutations to the operator parameter effectively change the neighbourhood relationships in genotype space. In some respects this is similar to the effect of drift on neutral networks, which allows the effective neighbourhood of a genotype to become the neighbourhood of the neutral network on which it resides rather the neighbourhood of the individual genotype. With self-adaptive parameters the neighbourhood of a genotype could potentially become the entire genotype space as the mutation rate can be increased to such an extent that any genotype can theoretically be reached from any other. However, there is an important distinction between these two effects. A neutral genotype-phenotype mapping can fundamentally change the nature of the landscape. This was seen in the RNA landscapes discussed in the previous chapter. Not all possible phenotypes played a role in evolution; rather genotype space was primarily divided into a number of neutral networks that represented the common molecular structures. These networks percolated throughout genotype space and as a result all were accessible within a relatively small distance of any arbitrary genotype. Thus, the neutrality fundamentally biased the search space in favour of certain phenotypes. An increased mutation rate may allow small valleys to be negotiated but does not produce an equivalent biasing effect.

In biology, evolution has been able to build on whatever the most probable molecular structures happen to be in the context of the laws of chemistry and physics. A challenge in artificial evolution is to design the mappings in such a way as to ensure that the most probable phenotypes and hence the percolating neutral networks represent solutions of high fitness. Addressing this challenge is a major theme of this work.

3.4 Hardware Evolution

3.4.1 On-chip evolution

One of the first analyses of the impact of neutrality on artificial evolution resulted from the on-chip evolution of electronic circuits [31]. In this work the configuration of a Field Programmable Gate Array (FPGA) was evolved in order to perform a tone-discrimination task [4]. The FPGA

consisted of an array of 64 x 64 reconfigurable cells each of which was connected to its four immediate neighbours; north, east, south and west. The function unit of each cell took up to three inputs that could be sourced from any of these neighbours. This unit could be configured to implement any Boolean function on two inputs or multiplexer function on three inputs. Each cell also produced four outputs in each direction. These outputs could be driven from either the output of the function unit or directly from one of the neighbours. Thus, a cell could directly 'route' an input to an output and hence connect two of its neighbours together or place the result of its logical operation on that output. The chip was supplied with a single input and a single output whose locations were fixed throughout.

The task chosen for this experiment was to discriminate between two tones of differing frequencies; 1 KHz and 10 KHz. Thus, when one of these frequencies was presented at the input to the array of cells the circuit's task was to indicate which of the frequencies was present by changing the voltage at its output. Ideally, one of the frequencies would generate a minimum voltage output and the other a maximum voltage output. In conventional electronics this is a trivial task; however, it was made difficult in this case as the circuit was not given access to a clock by which the period of the input could be timed and thus a continuous-time recurrent arrangement of cells needed to be evolved. Potential solutions were evaluated by supplying test tones at each frequency and calculating the average difference between the output voltages generated for each frequency, a circuit was rewarded for maximising this difference.

The configuration of the FPGA was controlled by the bits held in an on-chip memory, which could be written from software running a host computer. Not all of the cells were used in the experiment, only a 10 x 10 corner was under evolutionary control and the remainder of the cells were held at constant values. The configuration of these 100 cells was directly encoded in a linear bit-string genotype. Each cell required 18 bits to determine its four outputs and its logical function unit and thus the length of the genotype was 1800 bits.

Analysis of successfully evolved circuits revealed that only a relatively small subset of the components available to evolution was actually used in the final functioning circuit. Approximately two-thirds of the FPGA cells did not affect the circuit's performance and thus up to 1200 of the 1800 bits could be mutated without affecting fitness. It was hypothesised that the large-scale neutrality inherent to the phenotype-fitness mapping may have been of use in the evolutionary history of the circuit and an important factor in allowing evolution to discover the final configuration. This hypothesis was supported by calculating the variance of each gene within the population over a number of generations. As would be expected, areas of the genotype corresponding to the functional part of the circuit showed low variance indicating that selection pressure had held them relatively constant. In contrast, much of the neutral areas showed high variance indicating that they had been free to vary and under little selection pressure. However, some neutral areas showed relatively low variance indicating that they had been held constant for

significant periods and hence were likely to have formed part of the functioning circuit at some point during its evolutionary history.

This is an important feature of neutrality, for it to be of use it must have the potential of forming part of the functional phenotype at some point during evolution. It need not be used in the final functional phenotype but must have the ability to form part of *a* functional phenotype that may provide a bridge to a fitter phenotype that was previously inaccessible. The analysis suggested that the allowance of many more components than were necessary for the final circuit did create this type of neutrality in this example. However, in some cases the probability of a cell forming part of the phenotype would be very low. Each cell connected only to its nearest neighbours and it would thus be unlikely for cells that were a relatively large distance away from the currently functional part of the circuit to become an integral part of the circuit. In these experiments, the functional part of the circuit inevitably tended to form around the cells that received input and provided output. The integration of cells away from this area would also require the integration of a number of intermediary cells and would thus become more unlikely as the distance away from the functional area grew. Although the amount of neutrality was relatively high in this example, that with the potential of being beneficial was likely very much smaller. In this thesis it is argued that self-organisation produces a more fundamental restructuring of the landscape that can provide further enhancements to evolutionary search.

3.4.2 Simulated chip evolution

An alternative approach was taken in another example of hardware evolution in which potential solutions were assessed in simulation rather than on the actual hardware [127,129-131]. This work also used an array of cells that could be configured to perform Boolean logic functions on two inputs or multiplexer functions on three inputs. However, the fact that the experiments were conducted in simulation allowed the configuration of the cellular array to be more flexibly specified. The connectivity between the cells within the array was no longer restricted to nearest neighbours thus alleviating the problems of isolated non-functional areas of the circuit having little chance of becoming part of the functional circuit. In addition, a larger number of inputs and outputs were specifiable and could connect to a greater number of cells within the array.

The aim of the work was the evolutionary design of combinational circuits and particularly multipliers. These circuits were composed of atomic function building blocks such as the basic logic gates and one-bit multipliers. However, alternative approaches were also explored in which these building blocks were instead small sub-circuits that were inferred from other evolved designs [125]. In other experiments the initial population of genotypes was seeded with conventionally designed solutions that were then further optimised by evolution [126]. This approach resulted in a three-bit multiplier that was 23% more efficient than the conventional design in terms of the number of gates that were used [128].

In this case, a single row of 35 cells was used in which a directed graph was established that disallowed any recurrent connections. Thus, the inputs to each cell could be drawn from one of the 6 external inputs or from the output of cells from earlier in the graph. Within these constraints arbitrary connectivity was possible. Similarly, the 6 external outputs could be driven from any of the cells from within the array or directly from any input. The configuration of the array was encoded into a genotype that consisted of a linear string of integers representing two different types of gene. These genes defined the cells functionality and connectivity. Thus each cell was encoded by a maximum of four integers, three defining the inputs to a cell (one of which was not used in this example) and one defining the logical function of the cell. In addition, each output was encoded by a separate integer defining the cell or input to which it was connected. Thus, a 35 cell array with 6 outputs was fully specified by a genotype of 146 integers.

Although this encoding directly specified the configuration of the circuit there were a number of sources of neutrality; *input redundancy* resulting from inputs to the cell that were not used by the defined logical unit, *cell redundancy* resulting from cells that were disconnected from the functional circuit and *equivalency* resulting from circuits or sub-circuits that can be substituted with a logically equivalent alternative that has the same number of gates. Input redundancy was not a source of potentially useful redundancy in this case. The logic gates were restricted to two-inputs and thus the spare third input could never form part of the functional phenotype. In addition, equivalency suffers from a similar problem to the evolution of network topologies discussed in section 3.2. Although several logically equivalent examples of a given circuit may be possible they are likely to be significantly different from each other and transition between them would be difficult through neutral modifications to the genotype. The different examples would thus be isolated in genotype space rather than form neutral networks that allow potentially beneficial neutral drift.

Thus, it is likely that the most significant source of neutrality resulted from cell redundancy as with the on-chip experiment highlighted in the previous section. Indeed, analysis of the different sources of neutrality suggested that the majority of neutral drift took place on the neutral networks resulting from cell redundancy [127]. Evidence was also presented that suggested this neutral drift played a beneficial role. Neutral mutations were turned ‘off’ by disallowing any neutral moves in genotype space and the fitness of the final circuit compared to that of an equivalent process in which neutral moves were allowed. The average fitness of the circuit obtained after a given number of generations was found to be significantly higher in the latter case. These results were not conclusive as disallowing neutral moves opened up the possibility of disallowing *any* exploration of the space in a given generation and hence effectively resulted in fewer generations being used for the non-neutral case. Although this effect may have influenced the final fitness attained the results remain very suggestive of a beneficial role for drift on the neutral networks resulting from cell redundancy.

3.5 Robot Control Systems

3.5.1 Khepera robot simulations

Another area in which the impact of neutrality has been investigated in artificial evolution is the evolutionary design of control systems for mobile robots. The first example of this work concerned the evolution of a neural network for a Khepera mobile robot engaged in a simulated navigation task [113]. The Khepera robot has 8 proximity sensors around its circular periphery together with 2 motor-driven wheels. The task of the neural network was to take input from these sensors and drive the motors in such a way as to allow the robot to circumnavigate a corridor without collision with the walls. Neutrality was explicitly introduced into the neural network through the adoption of regulatory genes. A tree based genotype was used containing both regulatory and coding genes that specified the properties of neurones within the network. When a regulatory gene was activated the coding genes beneath it in the tree were expressed in the final neural network. When inactive, the coding genes lay dormant and hence any mutations to them were neutral. Thus, the possibility was open for dormant areas of the genotype to accumulate modifications that would prove beneficial when they were expressed in the final neural network through mutation to the regulatory gene. Evidence was presented suggesting that this possibility proved beneficial to the evolutionary process. Over a series of 30 trials, successful controllers were evolved in all but one case when regulatory genes were employed. In contrast, only 7 trials were successful when all neurones were expressed at all times.

The role of regulatory genes in this example is quite different from their role in biological genetic systems. In biological systems large numbers of genes within a cell are involved in complex networks of regulation. The dynamics of these genetic regulatory networks result in different patterns of gene expression within different cells as described in the previous chapter. Genes that are not expressed in one cell type do not lie dormant but are likely to be expressed within other cell types and are thus not free from selection pressure. Regulation does not have the effect of switching out areas of the genotype that are free to undergo arbitrary neutral modifications, rather it has the effect of creating a dynamic in which different patterns of gene expression can be manifested. The utility of switching out areas of the genotype must thus be questioned. Genes that are entirely free from selection pressure can be mutated arbitrarily without affecting fitness. The possibility of these mutations producing a sub-component that can then be usefully integrated into the phenotype is equivalent to the probability of random search discovering such a sub-component. This may be possible in some cases when the number of unexpressed genes is relatively low; however, as the number of unexpressed genes increases the probability of such an occurrence rapidly diminishes. This is a difficulty that will be discussed further later in this chapter.

The Khepera robot used in the above example was also the subject of another investigation in which neural network control systems were evolved to allow the robot to navigate a maze using cues received from its environment [95]. In this work the robot was tasked with making a left or right turn at a T-junction depending on the position of a light in the approaching corridor. If the light was detected on the left of the corridor the robot should turn left and vice versa. In order to achieve this, the robot could not rely on purely reactive behaviour as the position of the light needed to be memorised and used at the appropriate time.

Relatively small neural networks were evolved to perform this function. These networks consisted of only 10 neurones each with 3 links that could be arbitrarily connected to any of the neurones in the network. Neutrality was not explicitly introduced through the addition of regulatory genes, however, genotypes representing successful controllers were found to contain significant amounts of neutrality. Up to 60% of the genotype could be mutated without effect to fitness. Analysis was again performed that suggested this neutrality played an important role. The genotypes of successful controllers were pruned such that the remaining genotype specified only the functional controller i.e. the redundancy in the genotype was removed. When evolution was attempted with these reduced size genotypes, successful controllers were far less readily found. With the full complement of available neurones successful controllers were discovered in 90% of cases, this figure dropped to 30% for the smallest of the pruned genotypes.

As with the hardware evolution examples, the neutrality in this case resulted from an over-specification of phenotypic components that were not all part of the functional phenotype at any one time. Although an explicit switch was not used to prevent genes from being expressed, the non-functional components were effectively switched out by the evolved wiring of the network. An equivalent problem to the use of an explicit switch therefore also arises when the switch is implicit in the configuration of the phenotype. Random search must produce a sub-component that can subsequently be integrated into the functional phenotype.

3.5.2 GasNets

Both the above examples used relatively standard neural networks to implement the robot control system. A more complex neural network has recently been developed that employs an analogue of chemical signalling between neurones in addition to the electrical signalling resulting from synaptic connectivity. These networks mimic the diffusion of Nitric Oxide in real brains, neurones are able to emit gases that diffuse over a given radius and are able to change the behaviour of other neurones in a concentration dependent fashion [75]. Networks of this type have been successfully evolved to act as control systems for mobile robots in a number of different tasks and the gas signalling mechanism appears to not only increase the speed of evolution but also allow the evolution of more complex behaviours than are possible with more traditional networks [116].

A more complex genotype-phenotype mapping was required for this network that not only specified the electrical connectivity of the network but also parameters relating to the gas release and diffusion. It was used to generate networks that existed on a conceptual two-dimensional Euclidean plane with each neurone occupying a genetically determined coordinate. Rather than exactly specify the connectivity of the network, the genotype encoded a range for each neurone and connections were established with any other neurones that fell within that range. This technique was less direct than the previous mappings and opened up the possibility of beneficial neutral modifications to the range parameters. For example, in situations where the range parameter could not be immediately mutated in such a way as to beneficially alter network connectivity, a neutral modification may have been possible which changed the range such that subsequent beneficial modifications were possible. Similarly, the radius of gas emission could be modified to change the set of neurones that the emissions from a given node would affect.

These parameters were crucial in forming the functional phenotype and thus could not be mutated arbitrarily as would be the case for unexpressed genetic information. The fact that the parameters were always under the influence of selection pressure prevented random changes from modifying them to such an extent that the probability of them subsequently playing a functional role was diminished. Neutrality resulting from an over-specification of neurones was however also possible in this network and indeed some mutations resulted in the addition or removal of neurones thus opening up the possibility of self-adaptation of this form of neutrality.

The presence of neutrality in this genotype-phenotype mapping prompted analysis to determine whether this was a source of the observed increased evolvability of the networks [119,120]. A neutral phase in the successful evolution of a mobile robot that performed a visual shape discrimination task was identified and the dynamics of the population were analysed during this period. It was discovered that the population moved considerably through genotype space and genetic divergence increased, thus indicating exploration of neutral networks. However, this neutral drift did not appear to increase the probability of discovery of fitter phenotypes right up to the point at which a fitter phenotype was actually discovered. The authors thus concluded that nothing useful was occurring during the neutral phase, the population was moving but not to an area of genotype space that resulted in a significantly increased probability of the discovery of higher fitness phenotypes. This analysis was restricted to a single evolutionary run and thus the conclusions can not be generalised. However, this work again highlights the important point that neutrality alone is not sufficient to provide a beneficial role in evolutionary optimisation; it must be introduced carefully to yield beneficial properties on the search space.

Subsequent work modified the genotype-phenotype mapping and the gas diffusion mechanism in particular, with an explicit intention of increasing beneficial neutrality [117]. In the original

model the gas concentration decayed exponentially over the radius of its effect and thus different concentrations were present at each point. Any modifications to parameters such as the radius would therefore be likely to affect the concentration of the gas at all the nodes that were under its influence. This reduced the probability of mutations to these parameters being neutral. In order to increase this probability, a new mechanism was introduced that replaced the decaying exponential function with a uniform model such that any point within the gas's radius of effect would be influenced by an equal gas concentration. Changes to the radius would therefore not affect the concentrations at any of the influenced nodes and hence would more likely be neutral. In addition, a further modification was made that allowed the centre of the gas diffusion cloud to lie anywhere on the plane as opposed to being restricted to the nodes position. This dispersed model thus reduced the coupling between the two forms of signalling mechanism; electrical and gaseous connectivity were determined by entirely different mechanisms.

Experiments were performed in order to determine whether these modifications increased the speed of evolution as measured by the average number of generations required to discover a successful controller. It was found that neither the uniform model nor the dispersed model individually resulted in significantly improved performance. However, when both mechanisms were introduced together the speed of evolution was significantly faster. The authors thus concluded that both the neutrality resulting from the uniform diffusion and the decreased landscape ruggedness resulting from the dispersed diffusion was necessary to yield a beneficial effect.

While the interplay between ruggedness and neutrality is undoubtedly important, this may not have been the sole cause of this effect. The dispersed model not only reduced the coupling between the two signalling mechanisms but also had the potential of introducing a new source of neutrality. A change to the position of the centre of a gas cloud may not have affected the set of neurones that come under the gases influence. Nonetheless, with the original decaying diffusion mechanism it is unlikely that such a change would be neutral as the neurones would experience a different concentration of gas. However, if the dispersion is combined with a uniform gas concentration then such changes would be neutral. Thus the combination of effects had the potential of increasing the degree of potentially beneficial neutrality, which may have been an alterative cause of the results presented by the authors. This example highlights the need for techniques that that allow the neutrality as well as the ruggedness of a landscape to be measured and initial progress has been made in this direction by the same authors [118,121].

3.6 Pre-adaptation

A pre-adaptation is a characteristic evolved by an ancestral species or population that serves an adaptive though different function in a descendant species or population [102]. Such pre-adaptations were the focus of another study that explored the role of neutrality in artificial evolution [70]. In this work, artificial agents were evolved that inhabited a virtual grid

environment containing a zone in which food was plentiful together with two landmarks that could be used by the agents to guide themselves to this zone. The genotype specified instructions for the growth of a neural network that was formed via an analogue to axonal branching, which is a feature of the development of real nervous systems. The networks contained three different types of neurones, sensory neurones to allow the agents to assess the current environmental state, motor neurones to allow actions to be taken and intermediary neurones to allow additional processing. Agents were evolved to maximise their fitness by increasing the amount of time they spent in the food zone during a given run.

Neutrality was present at a number of different levels within this system. Firstly, at the genetic level as some areas of the genotype were not expressed in the final neural network. Secondly, at the phenotypic level as the nature of the growth process resulted in parts of the neural network that played no functional role in the agent's behaviour. Thirdly, a behavioural level of organisation was identified. A given neural network defined the set of possible behaviours that an agent could perform. However, in the context of its environment only a subset of this behaviour could ever used by the agent i.e. the agent may not experience the full range of sensory stimuli that are possible and thus may not engage in its full range of behaviour. Thus a further level of neutrality was introduced; different networks may yield a different set of possible behaviours but in the context of the current environment the same subset of behaviour may actually be used. Finally, two different behaviours had the potential of resulting in the same fitness. This system thus resulted in a hierarchical organisation consisting of four levels; genetic, phenotypic, behaviour and fitness all of which were a potential source of neutrality.

In order to assess the impact of mutation on this system the lineage of the best individual was identified and each parent/offspring pair examined. It was found that almost all offspring were genetically distinct from their parents. However, the probability of such a distinction gradually decreased at higher levels of organisation. Thus, each level of organisation was masking the effect of a proportion of mutations and only a small fraction remained that actually affected fitness. Evidence was presented suggesting that the remaining neutral mutations while not immediately beneficial did allow subsequent transition to functional networks of higher fitness.

The link to pre-adaptation in this study is questionable as the transition to a higher fitness network was not due to a prior adaptation that was subsequently modified to perform a different role but rather a previously neutral area that became functional. Nonetheless, it was an interesting example of the role of the neutrality and in particular of its effect on mutational robustness. Studies of molecular evolution have suggested that evolution naturally produces molecules that exhibit high degrees of mutational robustness [15]. In the previous chapter we saw that high degrees of neutrality also allow the formation of neutral networks and thus neutrality has the potential of playing a dual role; buffering phenotypes against the effect of deleterious mutation

while allowing constant evolutionary innovation along a neutral network. It is the later that is primarily of interest in this thesis.

3.7 Genetic Programming

Genetic programming (GP) was first introduced by Koza as a variant of a genetic algorithm in which functional computer programs are the target of evolution [44]. In GP, the program is represented by a hierarchical tree of nodes representing the functions and terminals of the program. The latter equate to leaf nodes and can represent the programs constants and variables, the former take input from a number of other nodes and perform some operation on this data. Some of these inputs may lead to further functions and thus a hierarchical parse tree can be formed that defines a program which can subsequently be created and executed. Genetic operators act on the trees in order to create variations by for example mutating the function a given node performs or recombining sub-trees in order to generate novel combinations of existing trees. Each candidate program is evaluated by providing input data and assessing the algorithmic output. Genetic programming has become one of the major sub-sets of evolutionary computation and has also resulted in several studies relating to neutrality. This work is highlighted below.

3.7.1 Constrainthandling

As the phenotype within GP is a functional computer program it must obey the grammatical constraints inherent to the chosen programming language. Such constrained optimisation problems are common in artificial evolution and a number of constraint handling techniques can be employed [124]. One possibility is to penalise constraint violating solutions and hence exert an added selection pressure such that they are removed from an evolving population. Harder constraints can also be imposed that prevent such solutions from occurring or forcibly correct them. However, these techniques have the potential of introducing undesirable features to the landscape and could for example exacerbate the problem of local optima by restricting movement in genotype space. In order to combat these problems, an approach has been suggested that employs a neutral genotype-phenotype mapping within GP enabling arbitrary movement within genotype space [136].

In this approach, the genotype takes the form of a binary string that encodes a set of functions and terminals. The genetic operators arbitrarily modify this string without regard for the grammatical constraints of the programming language. These constraints are instead implemented within the genotype-phenotype mapping which produces the nearest feasible solution for any string that would otherwise generate an infeasible solution. In this way, previously inaccessible regions of genotype space are replaced with neutral sets of genotypes and detrimental effects on the landscape may be minimised as neutral drift can allow continued movement in genotype space. Initial experimentation with this system proved that it was capable of generating solutions to

simple mathematical problems. However, no direct comparison was made with a non-neutral system to ascertain whether the neutrality had played a beneficial role. A potential problem with this approach is that the neutrality is defined by the constraints of the problem. Neutral sets are guaranteed to be formed for any constrained problem but these sets may not be connected into neutral networks allowing exploration of the space and hence discovery of phenotypes that would otherwise not have been possible. In order to create an opportunity of ensuring this, greater control over the introduction of neutrality would be desirable and thus a looser coupling between genotype and phenotype may be required.

3.7.2 Cartesian genetic programming

The standard GP representation implicitly contains two forms of neutrality; functional redundancy and introns. The former refers to the fact that a given function can be implemented in a number of different ways and the latter refers to functions that are semantically redundant to the programs behaviour i.e. functions that are executed but would not change the outcome of the program if they were removed. An alternative representation has recently been proposed that adds a third source of neutrality resulting from unexpressed genes; this approach has been termed Cartesian Genetic Programming (CGP) [38] and was originally developed to evolve electronic circuits as described in section 3.4.

In CGP, a program is represented as a directed, acyclic graph. Nodes represent functions whose outputs can from part of the input of subsequent nodes in the graph. However in contrast to the standard parse tree representation, a graph allows its nodes to be unconnected from any other nodes i.e. their outputs do not form an input for any of the other nodes in the graph. These unconnected nodes are therefore neutral with respect to the programs behaviour. An instructive analysis of the effect of neutrality in CGP was carried out in the context of the evolution of a simple even-3-parity program i.e. a Boolean function that takes 3 inputs and returns *true* iff an even number of its inputs are *true* [123]. This program was evolved using a function set consisting of four 2-bit Boolean logic gates and a terminal set consisting of the three Boolean inputs. The graph consisted of 100 nodes each of which was represented by 3 genes defining the type of the function together with its inputs. There are 256 possible Boolean functions that define an output for each of the 2^3=8 possible states of the inputs, which correspond to the possible functional phenotypes and thus with 300 genes operating with a large alphabet there was very large scale neutrality in the genotype-phenotype mapping.

Fitness was determined by calculating the number of correct outputs for all 8 combinations of the three inputs and was thus quantised to be any integer value between 0 and 8. The most common phenotype was of fitness 4 and phenotypes of higher fitness became progressively less common. The 9 fitness values resulted in 9 neutral networks in genotype space and analysis was carried out

that studied the likelihood of evolution finding transition points between these networks with different mutation rates and different degrees of restriction on neutral drift. The latter was enforced by placing limits on the number of different genes between the genotype at which a new fitness was first discovered and any subsequent genotype produced by neutral drift. The analysis revealed a correlation between the permissible range of neutral drift and the probability of success. At the most common fitness levels i.e. fitness 4 and 5, transitions were relatively easy to discover. However, as fitness increased the probability of success reduced in line with the permissible range of neutral drift. Indeed, the final transition to fitness 8 was only discovered in 100% of cases when the amount of neutral drift was allowed to approach that of the genotype length i.e. when the majority of genes could be modified during neutral drift.

An interesting effect of mutation rate was also discovered. Higher mutation rates allowed evolution to compensate for restricted neutral drift at low fitness levels, however as fitness increased, a higher mutation rate was not able to make such compensation. This suggests that evolutionary innovations along a neutral network could not be replicated by simply remaining in the same area of genotype space and increasing the mutation rate in this case.

These results provide further evidence of the benefit of allowing evolution access to more components than are necessary in the final phenotype. However, the relative lack of success of experiments with reduced neutral drift may have been influenced by the experimental design. Neutral drift was restricted by disallowing neutral modifications that resulted in the threshold of genetic variance being exceeded. However, with such an occurrence it was possible for an entire evolutionary generation to effectively be lost. At the limit of neutral drift, if all mutations were neutral then there was no movement in genotype space and with high levels of neutrality the probability of such an occurrence was relatively high. Thus, introducing neutrality into the search space but disallowing neutral mutations may have hampered the search process by artificially restricting exploration of genotype space. A fairer comparison may have been to compare CGP with an alternative encoding that did not contain any neutrality; however, the design of such an encoding is a difficult task in itself. Despite this potential effect, these results remain strongly suggestive of a beneficial role for the neutrality within CGP.

3.8 NK Fitness Landscapes

In addition to the studies exploring the role of neutrality in the evolution of artefacts outlined above, studies have also been performed in the context of more abstract fitness landscapes based on the NK model introduced by Kaufmann [111,112]. This model is intended to capture the essence of real genetic systems such that the characteristics of the resulting fitness landscapes can be explored. The model consists of *N* loci that can be thought of as representing the traits of an organism or at a lower level, individual genes. Each locus can be in one of a number of states,

which might represent the different alleles of a gene for example. In addition each locus interacts with *K* other loci, which are typically chosen at random. These interactions can be thought of as representing epistatic interactions. Each locus makes a contribution to fitness that is dependent not only on the state of that locus but the states of each of the *K* neighbouring loci. Defining the number of possible states of the loci as *A*, there are A^{K+1} possible fitness contributions for each loci and the fitness of the genotype is given by the average contribution over all loci.

In order to study generic properties of systems of this kind, Kaufmann chose to assign the fitness's at random with each of the NA^{K+1} possible fitness contributions randomly assigned a number between 0 and 1. As each fitness contribution is given by a random number that is almost certainly unique, the standard NK landscapes do not contain any neutrality; the modification of any locus is almost guaranteed to affect fitness. However, several extensions of the model have been proposed that introduce neutrality into the model such that generic properties of neutral landscapes and the dynamics of evolving populations on such landscapes can be studied.

Two different methods of creating neutral NK models have been proposed; NK*p* landscapes [49,50] and NK*f* landscapes [55]. The former introduces a new parameter, *p*, that varies in the range 0 to 1 and represents the probability of a given fitness contribution being 0. Thus, when *p* is 1 all fitness contributions are zero and the landscape reduces to a single neutral network of zero fitness and when *p* is 0 the landscape is identical to the standard NK model. Intermediate values of *p* introduce variable degrees of neutrality into the landscape thus allowing it to be tuned. The latter model takes a different approach, the fitness contributions are quantised into a set of *f* discrete levels that are integers within the range 0 to *f-1*. The overall fitness is then scaled by a factor of $1/N(f - 1)$ in order to bring it within the range 0 to 1. This quantisation again allows the probability of neutral mutation to be tuned. When *f* is low it is much more likely that a change at a given locus will result in an equivalent fitness contribution as these contributions are drawn from a small set. As $f \rightarrow \infty$, the model becomes equivalent to the standard NK model.

The differences between these approaches are reflected in differences of the fitness distributions within the corresponding landscapes [68], however certain generic features emerge that have strong parallels with features of the RNA landscapes highlighted in the previous chapter. The NK*f* landscapes revealed an equivalent phenomenon to that of common RNA structures. That is, there were a relatively small number of common phenotypes that were represented by the majority of genotypes. The common phenotypes thus dominated the evolutionary dynamics. In addition, it was shown that higher degrees of neutrality typically resulted in the achievement of higher fitness values. Another important property emerged from studies of the NK*p* landscapes in which evolving populations were shown to diverge in genotype space during periods of neutral drift. This allowed the continual innovation of new phenotypes i.e. each neutral step gave access to a constant number of new phenotypes. This constant innovation property was also observed in the RNA landscapes.

The studies of these abstract landscapes suggest that the properties discovered in RNA landscapes may be general features of highly neutral mappings. However, a challenge within artificial evolution is to ensure that designed mappings introduce similar properties to a landscape. There are a number of design choices that must be made in the construction of such mappings, many of which are dependent on the nature of the particular problem at hand. Any one of these decisions could result in the introduction of landscape features that are undesirable from the point of view of neutrality. Thus, while the knowledge gained from abstract landscapes is instructive they can give little guidance in the construction of mappings for real problems within artificial evolution. Making progress in this direction is a principle aim of this thesis.

3.9 Optimising Neutral Evolution

In designing an artificial evolutionary system there are a number of decisions that must be made. In addition to the nature of the encoding, features such as population size, mutation rate and selection strategy must be chosen. These choices remain something of a black art within the field and the presence of neutrality further complicates the issue. In order to address these problems several studies have been undertaken that aim to mathematically analyse evolutionary dynamics on abstract neutral landscapes. The lessons learnt from such analyses have the potential of allowing the various parameters of an evolutionary algorithm to be optimised for different problems. This work is the subject of this section.

3.9.1 The royal road

The first study of this kind focussed on so-called royal road fitness functions, which capture many of the features of neutral evolution while remaining mathematically tractable [16]. The royal road consists of N blocks of K bits such that the total length of the bit string is $L = NK$. Each of the N blocks has a particular desired configuration and when the block is in that configuration it makes a contribution to the overall fitness of the genotype. Thus, fitness increases in a step-wise manner with each step equating to the discovery of a new aligned block. Maximum fitness is achieved by only a single configuration when all blocks are individually aligned. The royal road thus provides a simple example of a neutral mapping; many mutations will be neutral as they do not change whether or not a block is aligned. Evolutionary dynamics on the royal road are characterised by periods of stasis, termed epochs, in which these neutral mutations occur followed by rapid gains as a new aligned block is discovered.

Their theoretical approach was termed statistical dynamics with analogy to classical statistical mechanics. Rather than focus on the distributions of individuals within a population, this approach focused on the distribution of fitnesses in the population. Thus, the fitness landscape was coarse-grained into a set of neutral networks equating to each of the $(N+1)$ possible fitness

values. The fine-grained details were removed from the analysis by making a maximum entropy approximation. That is, in the limit of infinite populations any one of the possible fine-grained states is equally likely for a given fitness value. Thus, under this approximation the evolutionary system can be deterministically and self-consistently described in terms of macro variables, which equate to the percentage of individuals at each possible fitness level. The resulting state space is thus much more manageable, consisting of only *(N+1)* variables. This is equivalent to the use of macroscopic variables such as temperature, pressure and volume in physics.

Given this knowledge of the state space, full knowledge of the evolutionary system dynamics can be ascertained by deriving equations that describe the system trajectories within this space. These trajectories are defined by the genetic operators and selection scheme that is used. In this case, analysis was performed in the context of a mutation-based evolutionary algorithm with a fitness-proportional selection scheme. Given this system level knowledge, a number of quantities were mathematically derived including the expected duration and stability of an epoch and the expected time between evolutionary innovations. Thus, this powerful approach allows insight to be gained into the likely performance of different algorithms on certain classes of problems.

3.9.2 The royal staircase

This work was built on in a subsequent study that modified the landscape to form the so-called royal staircase fitness function [14]. As with the royal road, this function consists of *N* blocks of *K* bits each of which has a desired configuration. However, in this case an aligned block only makes a contribution to fitness if all previous blocks are also aligned. Thus, the genotype is read from left to right and the fitness equates to the consecutive number of aligned blocks. This landscape was chosen to implement the intuitive idea that the proportion of genotypes in a neutral network falls exponentially with increasing fitness i.e. good solutions are very much more rare than poor solutions.

The same statistical dynamics approach was used to study this landscape, which allowed the average number of fitness function evaluations to reach the global optimum to be determined as a function of mutation rate. Given this data, it was straightforward to determine the optimal mutation rate for landscapes of this type. Further analysis was also performed that took into account the effect of finite population sizes and thus allowed the average number of fitness function evaluations to be determined as a function of both mutation rate and population size [13]. These results suggested that the population size should be as low as possible as individuals did not explore neutral networks individually but were genetically correlated and thus high population size wasted computational resources without yielding any beneficial effect.

While plausible, these results do not resonate with the observed population dynamics in the studies of RNA evolution explored in the previous chapter. These studies suggested that genetically diverse quasi-species tended to form on a neutral network thus allowing different areas of a neutral network to be simultaneously explored [64]. However, the results were supported by a similar mathematical study that focused on landscapes with several realistic and plausible assumptions; namely that the probability of a point-mutation taking a sequence to a higher fitness neutral network is very small compared to the probability of it being neutral or deleterious and that the only non-negligible fitness increasing mutations are those to the next highest network [51]. This work also suggested that the optimal evolutionary search strategy was to use a single individual that engaged in hill climbing with neutral drift.

Analyses of this nature promise to yield very useful insights into the dynamics of evolution on stereotypical landscapes. These insights will lead to a more general and quantitative understanding of evolution that will allow considered choices to be made in the design of an evolutionary algorithm. However, as with any mathematical analyses they depend on certain assumptions. In this case the assumptions regard the nature of the landscape on which a population is evolving. In particular, they assume the presence of connected and percolating neutral networks with access points between them that allow the global optimum to found through a series of transitions between networks. While this is the kind of landscape that is witnessed in studies such as RNA folding and is the kind of landscape that would be desirable in artificial evolution, it is by no means guaranteed. It is relatively straightforward to define abstract landscapes that implement the desired properties but it is much less straightforward to do so when evolving artefacts. The mapping between genotype and phenotype and subsequently fitness is crucial in creating landscapes of the type studied in these mathematical analyses and thus a greater understanding of the consequences of the choices made in these mappings would be highly desirable. Gaining such an understanding is a theme of this thesis.

3.10 Discussion

Neutrality is likely to have played a role in artificial evolution since its inception as the nature of many encodings implicitly results in the presence of neutrality in the search space. However, there is an increasing realisation that the principled introduction of neutrality may yield significant advantage to the evolutionary process and a number of examples of such work have been highlighted above. A very common approach is to allow areas of the genotype that are unexpressed, that is they play no functional role in the phenotype and are thus free from selection pressure. This may result from the provision of an explicit evolutionary switch that controls whether regions of the genotype are expressed in the phenotype or implicitly in the architecture of the phenotype i.e. the connectivity of phenotypic components is such that subsets of these components do not form part of the functional phenotype.

A substantial amount of evidence has been presented above that suggests this is a fruitful approach and allows the evolution of phenotypes more reliably or more quickly than would otherwise have been possible. However, this use of neutrality is quite different from the neutrality that has been studied in biological systems such as RNA molecules or that arises in models of genetic regulatory networks such as the random Boolean network. In these systems, the natural self-organising dynamics result in the presence of attractors or preferred states of the system. The neutrality results from many initial states resulting in the same attractor. Although the precise state of each individual gene may not be crucial, these genes remain an integral part of the dynamical process. For example, genes that are not expressed in one cell type are not free from selection pressure as they are likely to be expressed in other cell types. These genes are an integral part of a genetic regulatory network whose dynamics lead to different patterns of gene expression in different cells.

In the previous chapter it was suggested that these dynamics provide a source of order that constrains the phenotypes that are available to evolution. A similar effect is not likely to be achieved by allowing unexpressed regions of a genotype that are free from selection pressure. This approach does not bias the set of possible phenotypes and is not a source of order. Rather, unexpressed regions undergo random search with the hope that they may be integrated into the functional phenotype at a later stage. In natural evolutionary systems, this is equivalent to the presence of so-called “junk DNA” that in the absence of selection pressure quickly diverges from coding regions of DNA and is highly unlikely to ever again code for proteins.

While this approach may be successful for small problems it may ultimately be limited. Consider an extreme example in which a genotype is duplicated and a genetic switch introduced that determines whether the original or the duplicate will be expressed as the phenotype [69]. Each genotype is independently capable of specifying the required phenotype and thus the possibility of the currently unexpressed copy randomly generating the required phenotype through neutral drift exists. The genetic switch could then be mutated and the search would be successful. While this is a possibility, the probability of such an occurrence is negligible for any realistic genotype. Local optima do not exist in this space but transitions between many phenotypes are extremely unlikely. The probability is increased if smaller regions of the genotype are unexpressed and subsequently switched back in to the phenotype but the same principle applies.

In order to gain full advantage from neutrality it may be necessary, therefore, to introduce a dynamical process in to the genotype-phenotype mapping; a process that introduces an equivalent order to that arising from the dynamics of molecular folding or genetic regulatory networks for example. Biology has been able to build on whatever common phenotypes are produced in the context of our chemical and physical laws. However, in artificial evolution the genotype-phenotype mapping needs to be designed in order to encourage the common phenotypes to be those of high fitness. In addition, these common phenotypes must be represented by expansive

and percolating neutral networks such that they are probable discoveries as a population drifts along a neutral network. This work is concerned with developing mappings of this nature.

3.11 Summary

In this chapter a number of studies that have explored the impact of neutrality in artificial evolutionary systems have been reviewed. A number of key features were drawn out and are highlighted below:

- Neutrality has likely been an implicit feature of many artificial evolutionary systems due to the nature of typical problem representations.
- There is an increasing realisation that the principled introduction of neutrality may yield significant advantages to artificial evolution. These benefits include the speed and reliability at which high fitness phenotypes can be discovered.
- A very common approach is to allow unexpressed regions of a genotype that are free from selection pressure. This may either be through the explicit use of a genetic "switch" or implicitly in the configuration of the phenotype.
- This approach does not exploit the order that arises as a result of the dynamics of a self-organising process in which the system tends toward preferred states or attractors. Such processes are the root of the neutrality studied in RNA evolution and genetic regulatory networks for example.
- In order to fully exploit neutrality, such dynamics may need to be introduced into artificial genotype-phenotype mappings.

The remainder of this work is concerned with developing mappings of this nature. In the following chapter, several abstract genotype-phenotype mappings are introduced that mimic natural self-organising processes. The aim of this work is to generate the landscape properties that result from natural self-organising processes by introducing abstractions of those processes into artificial genotype-phenotype mappings.

Chapter 4

Abstract Genotype-Phenotype Mappings

4.1 Introduction

In has been argued that self-organisation is highly likely to play a role in natural evolutionary systems. One effect of self-organisation is to introduce neutrality into the search space as many genetic configurations give rise to the same final system behaviour or phenotype. However, it was seen in the previous chapter that the neutrality resulting from this natural order has not yet been fully explored in artificial evolutionary systems. In this chapter, two genotype-phenotype mappings are developed that are based on natural self-organising processes. The motivation behind this approach is to generate search spaces that exhibit the properties that are found in natural search spaces through the use of abstractions of natural processes.

A major influence in the design of these mappings was the extensive study of the landscapes created by the folding of RNA molecules highlighted in Chapter 2. This work revealed a number of beneficial properties resulting from the presence of large-scale neutrality. One method of encouraging similar properties into artificial mappings would thus be to use abstractions of these well studied processes. The mathematical models used within the studies allow the pertinent structure of RNA to be ascertained given a nucleotide sequence, i.e. a phenotype to be calculated given a genotype, and could thus be used as a basis for an artificial mapping. However, there are difficulties with this approach as the algorithms are computationally intensive and would thus result in a significant performance penalty for any evolutionary algorithm incorporating them. It is difficult to justify these penalties as the precise details of the mapping are not likely to be crucial; several different models of RNA-folding exist that yield very similar large-scale landscape properties. In addition, we have seen in the previous chapter that abstract fitness landscapes such as the NK*p* landscape can be created via very simple mechanisms that produce similar properties to those witnessed in RNA-folding. For these reasons, the two mappings introduced in this chapter are not specific models of natural processes but rather computationally efficient, abstract models that ignore many of the details of natural systems but capture their essence and ability to self-organise into orderly states. They are based on a cellular automaton and a random Boolean network and are described in detail in the following section.

It was seen in Chapter 2 that the large-scale neutrality inherent to RNA folding resulted in neutral networks that percolated throughout genotype space. Movement on these neutral networks

allowed the constant innovation of new phenotypes. It is this property that is the primary statistic used to explore the properties of these abstract mappings. Such constant innovation would be highly desirable in artificial evolutionary systems. Ultimately however, the discovery of new phenotypes must allow higher fitnesses to be reached. The ability of the mappings to achieve this is also assessed using several different fitness functions.

4.2 The Mappings

In order to compare the ability of the different mappings to allow discovery of new phenotypes, it is important that the same phenotypes are possible for each mapping i.e. that phenotype space is identical. In order to achieve this, the phenotypes were represented using a binary encoding of a fixed length in each case. For a string of length L the number of phenotypes was given by:

$$P_N = 2^L \qquad \textit{Equation 4.1}$$

Given this constraint, the design of a mapping involved both the choice of a genetic encoding and a process that generated each of the phenotypes using that encoding. The former defined genotype space and the latter the genotype-phenotype mapping. For a direct encoding, genotype space and phenotype space are equivalent i.e. exactly one genotype maps onto each phenotype. However in order to generate potentially beneficial neutrality more complex processes are required. It is for this purpose that the cellular automaton and random Boolean network were employed.

4.2.1 The Cellular Automaton Mapping

4.2.1.1 Cellular Automata

Cellular automata (CA) were first developed by Von Neumann as models of self-reproducing machines [45]. They consist of a number of simple, locally-interacting computational elements or cells that self-organise into stereotypical patterns of activity [1,109]. These properties make them well suited as models of natural self-organising processes and they have been widely used for this purpose. Examples include fluid flow or lattice gas models [41] and the reaction-diffusion chemical systems highlighted in chapter 2 [54]. In addition, they have been used to model biological phenomena such as animal patterning [25], ecosystem dynamics [74] and immune responses [29].

Each cell within the CA takes the form of a finite state automaton whose state transitions are defined by simple computational rules that are influenced by the state of neighbouring cells. Although many CA architectures are possible [109], one common approach is to arrange the cells

in to a one-dimensional array in which the state transitions are governed by the current state of a cell together with those of its two immediate neighbours. An example of such a CA is shown in Figure 4.1.

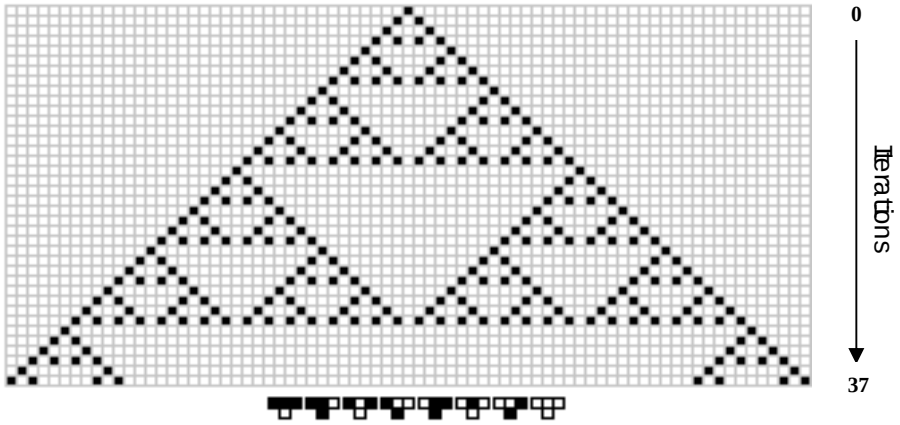


Figure 4.1: A one-dimensional two-state cellular automaton consisting of 75 cells organised into a horizontal array. The state of the CA in successive iterations is shown flowing down the image. The state transition rules are shown below.

The figure shows a one-dimensional array of 75 two-state cells that obey the state transition rules shown beneath the image. These rules define the next state of a cell given each of the 8 possible states of that cell's neighbourhood i.e. the 8 combinations of its own state together with those of its two immediate neighbours. The CA is initialised with a single "live" cell and synchronously updated for a number of iterations resulting in a well ordered pattern of activity. The precise pattern that is produced is dependent on both the initial state of the CA and the state transitions rules. Changes to each of these can not only change the details of the pattern that is produced but can also generate qualitatively different behaviour that ranges from fixed-point to chaotic attractors [110]. The effect of a single change to the transition rules of the above CA is shown in Figure 4.2.

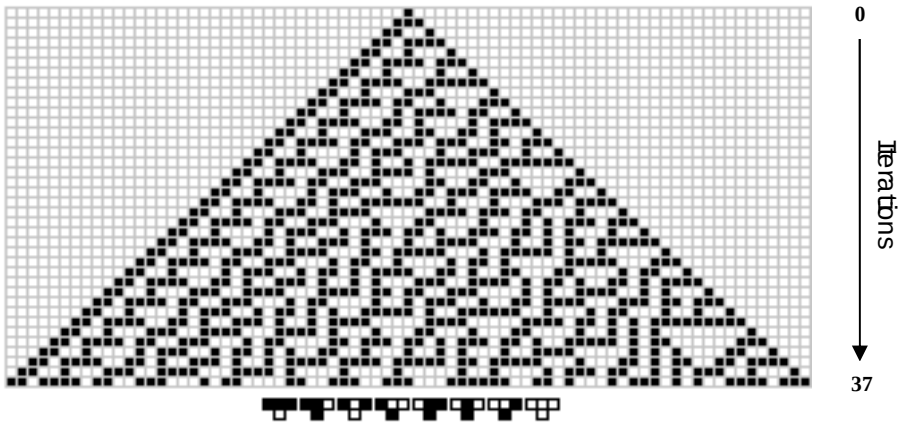


Figure 4.2: The effect of a single change to the CA's state transition rules. A very different behavioural pattern is produced.

Given the ability of cellular automata to act as models of natural self-organising systems they are good candidates to form the basis of the abstract genotype-phenotype mappings of interest in this chapter. One possibility of exploiting the CA's pattern generating ability would be to identify the attractors in the dynamics and equate each attractor to a phenotype. However, there are a number of difficulties with this approach. Chief among these being that the type and number of attractors is difficult to control. A different approach was therefore taken in which the CA was updated for a fixed number of iterations and the resulting state taken as the phenotype. The actual number of iterations was chosen to be 20 to allow any initial fluctuations to settle down while minimising the computational overheads of the mapping [58,59,63,97-99].

In this mapping the number of two-state cells in the CA defines phenotype space i.e. the number of cells is equal to the length of the phenotype *L*. The genotype-phenotype mapping equates to the dynamics of the CA over the 20 iterations and the resulting state gives one of the 2^L phenotypes. The genotype encodes the information required to initialise the CA, which is explained in greater detail in the following section. The mapping is summarised in Figure 4.3.

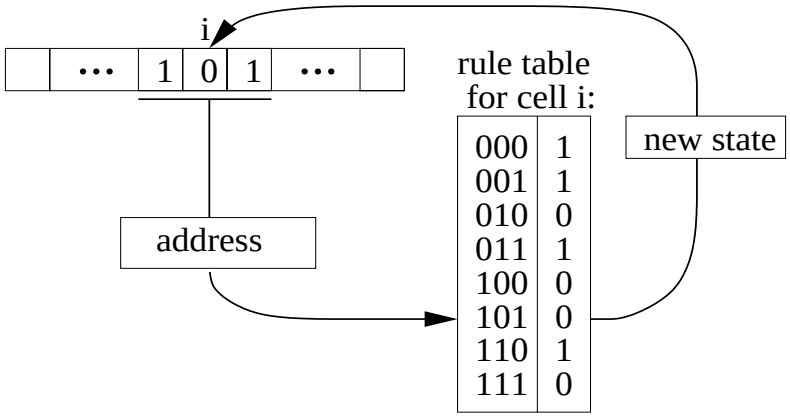


Figure 4.3: Summary of the cellular automaton mapping. A cells state together with those of its immediate neighbours defines an address into that cells' rule table which gives its next state. The automaton is updated for a fixed number of iterations and the resulting state of the CA gives the phenotype.

4.2.1.2 Genetic encoding

One important feature of the CA that requires encoding in the genotype is its initial state. Changes to the encoded initial state would result in a different pattern of activity and hence may result in a different final state or phenotype after the 20 iterations. However, it is not likely that all phenotypes could be generated by such an encoding as the CA's dynamics contain a number of attractors. By definition therefore, a number of initial states would result in the same final

behaviour pattern and would thus be likely to produce the same phenotype if the number of iterations was large enough to allow the CA to settle into its attractor.

As highlighted in the previous section however, the pattern produced by the CA is not only affected by the initial state but also by the transition rules. These rules can also be encoded in the genotype in order to increase the range of phenotypes that can be produced by the mapping. In addition, greater flexibility can be achieved by encoding independent state transition rules for each cell i.e. by using a non-uniform cellular automaton [67]. Such a mapping is very likely to allow the production of all 2^L phenotypes. The resulting genotype is depicted in Figure 4.4.

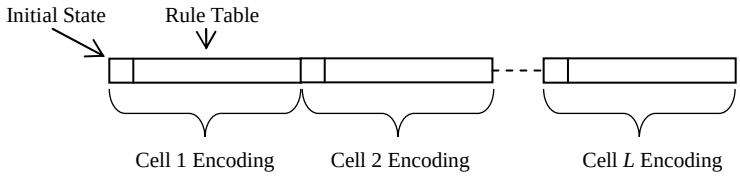


Figure 4.4: The genetic encoding for the cellular automaton mapping.

A cell's next state is dependent on its own state together with that of each of its neighbours and rules are required for each possible combination of these states. Thus the size of the rule table is 2^{K+1} bits, where K is the number of neighbours. In addition to the rule table, each of the L cells required its initial state to be encoded. For the two-cell neighbourhood used in this mapping, the total length of the binary genotype is thus given by Equation 4.2.

$$G_{CA} = (1 + 2^{K+1})L = 9L \qquad \textit{Equation 4.2}$$

4.2.2 Random Boolean Network Mapping

4.2.2.1 Random Boolean networks

The random Boolean network (RBN) was introduced in chapter 2 as an abstract model of a genetic regulatory network. It consists of an array of cells that are regulated by a number of other cells within the network. In its simplest form each cell can be in one of two states, "on" or "off", and a cell's regulatory inputs define which of these two states a cell adopts. As was described in chapter 2, a natural consequence of these regulatory webs of interaction is to place constraints on the behaviours that a network can exhibit; the network self-organises into one of a relatively small number of attractors. This behaviour makes them good candidates to act as a self-organising genotype-phenotype mapping. However, equating these attractors with phenotypes suffers from the same problems as for the cellular automaton mapping; predicting and controlling the nature of the attractors is problematic. A similar solution can however be adopted in which

the network is updated for a fixed number of iterations and the resulting state interpreted as the phenotype. As for the CA mapping, the number of iterations was chosen to be 20. This mapping was introduced by Shipman et al. [98] and subsequently explored in detail in a number of related papers [58,59,63,97,99].

The RBN can be thought of as a generalisation of a CA in which the constraints on cell neighbourhood are relaxed. With the CA, a cells' neighbourhood was restricted to its immediate neighbours. However, the cells within the RBN can be regulated by any of the cells within the network. In addition, a cell's own state need not influence its next state. The RBN mapping is depicted in Figure 4.5.

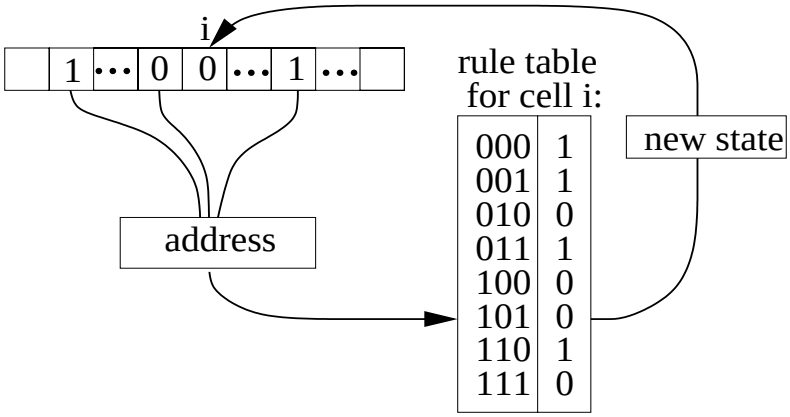


Figure 4.5: Summary of the RBN mapping. The initial state and rule tables are encoded in the genotype as for the CA mapping. However, in this case a cells' neighbourhood is no longer restricted to its immediate neighbours.

4.2.2.2 Genetic encoding

The genetic encoding of the RBN mapping is very similar to that of the CA mapping. However, extra information is required to specify the regulatory inputs. The resulting genotype encodes the initial state, the rule table and indexes for the regulatory inputs for each cell as shown in Figure 4.6. Each of the K regulatory inputs is encoded as an index to one of the L cells in the network and thus requires $log_2(L)$ bits. The total number of bits required to encode the regulatory inputs is thus $Klog_2(L)$ for each cell. For the 3-cell neighbourhood used in this mapping, the total length of the binary genotype is given in Equation 4.3.

$$G_{RBN} = (1 + 2^K + K\log_2(L))L = (9 + 3\log_2(L))L \qquad \textit{Equation 4.3}$$

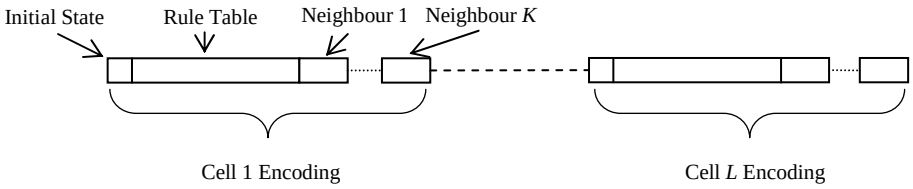


Figure 4.6: Genotype structure for the random Boolean network mapping. In addition to a cell's initial state and state transition table, its regulatory inputs are also genetically encoded.

The encoding of regulatory inputs greatly increases the size of genotype space, for an 8-bit phenotype the CA mapping results in 2^{72} genotypes whereas the RBN mapping results in 2^{144} genotypes. As phenotype space is identical in each case there is a far greater degree of neutrality in the RBN mapping.

4.3 Phenotype Accessibility

It has been argued that drift on neutral networks gives access to many more phenotypes than would be possible with a direct encoding when no such neutral drift is possible. The aim of this section is to assess whether this holds true for the CA and RBN mappings. In order to determine which phenotypes are accessible from each neutral network within genotype space, a measure of accessibility must be defined. In any evolutionary algorithm, the likely transitions between phenotypes are influenced not only by the structure of the search space but also the genetic operators that are used to navigate that space. Many such operators are possible, recombination and mutation being two of the most common. While some of these operators have the potential to allow large movement in genotype space, the focus of this study is on the basic structure of the search space created by the two mappings. For this reason, the study is restricted to local movement in genotype space generated by the mutation operator.

Many evolutionary algorithms use a mutation operator in which each gene is mutated with a given probability. With such a scheme it is *theoretically* possible to change the entire genotype in a single reproductive event and hence every phenotype is theoretically accessible from every other. However, for any realistic genotype the probability of the majority of these transitions is negligible; a typical mutation event will result in far fewer genetic mutations and hence smaller movements in genotype space. This study considers only the smallest of these movements, those that are generated by single-point mutations. If constant innovation of phenotypes can be demonstrated for such restricted movement in genotype space, this will be a strong case to suggest that it would also occur in any real evolutionary algorithm using these mappings. Multiple mutation events and more sophisticated operators would only further increase the accessibility between phenotypes.

For a direct encoding the number of phenotypes that are accessible from a genotype of length L using only single-point mutation is given by:

$$P = (A-1)L \qquad \textit{Equation 4.4}$$

Where A is the number of genetic alleles, for the binary genotypes used in this study the number of accessible phenotypes is thus equivalent to the length of the genotype i.e. each gene can be individually mutated to result in a new phenotype. If all of these phenotypes are of lower fitness, this genotype corresponds to a local optimum. For the CA and RBN mappings to have the potential of alleviating the problem of local optima, they must allow access to a greater number of phenotypes than the direct encoding on average. A key tool in assessing whether this was the case is the random neutral walk [65]

4.3.1 The Random Neutral Walk

Consider an individual within an evolving population that resides on a particular neutral network. Each gene within this individual has the potential of being mutated. Some of these mutations will be neutral and will cause the individual to move along the neutral network; others will result in different phenotypes and allow the population to probe the boundaries of the neutral network. As the population drifts along the neutral network, new boundaries are opened up for exploration which may allow the discovery of new, better adapted phenotypes.

The random neutral walk mimics this process by modelling a single individual engaged in neutral drift. A series of steps along the neutral network are taken by randomly choosing one of the possible neutral mutations. At each step, the boundaries of the neutral network are assessed to give a measure of the accessibility of different phenotypes from that neutral network. A number of neutral walks can be performed on each of the neutral networks within genotype space to give a good indication of the ability of neutral drift to allow access to phenotypes that would otherwise have not been possible. The process is visualised in Figure 4.7 and formalised below for a walk on the neutral network associated with a phenotype P_0:

1. Randomly choose a genotype G_0 mapping onto phenotype P_0
2. Generate all single-point mutants of G_0 and store in list M_0
3. Generate a list of neutral neighbours N_0 consisting of all members of M_0 that map on to P_0.
4. Log all remaining non-neutral members of M_0
5. Randomly choose a member of N_0 to become G_0.
6. Repeat from step 2 until a given number of steps have been performed or N_0 is empty.

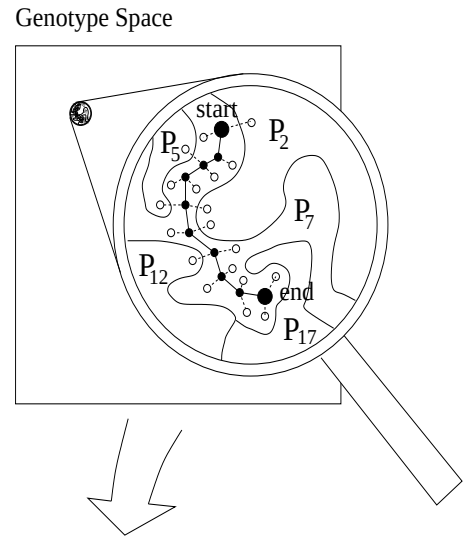


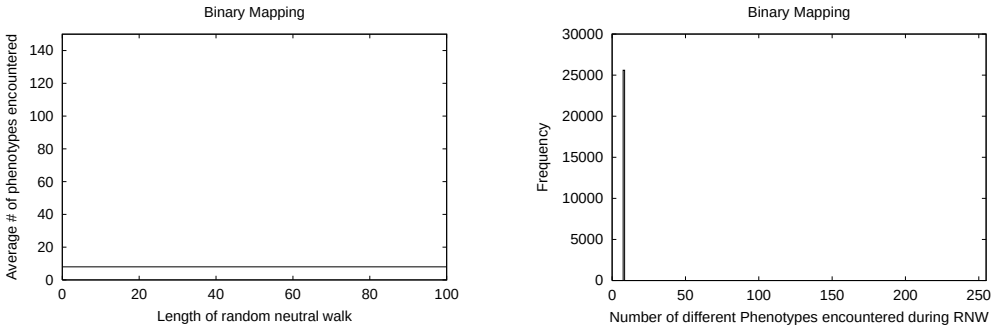
Figure 4.7: Visualisation of a random neutral walk in genotype space. Neutral mutations are chosen at random and the phenotypes that are within a single mutation of each resulting genotype are assessed.

4.3.2 8-bit Phenotype Spaces

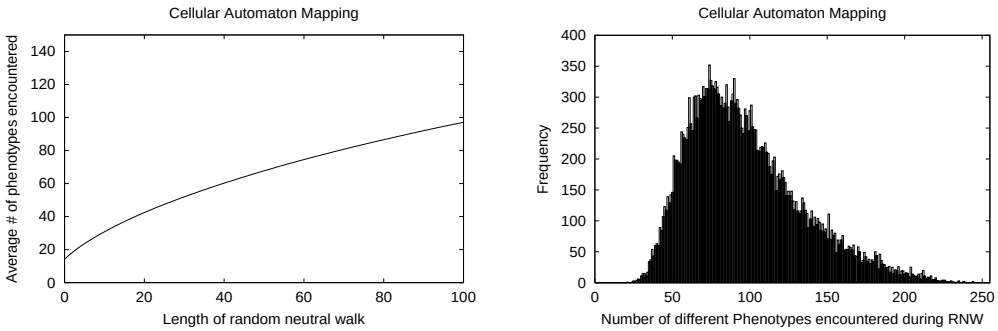
An 8-bit phenotype space results in a total of 2^8=256 possible phenotypes. Any realistic problem would likely generate larger phenotype spaces, however, these small spaces reduce the computational overheads required to generate the statistics while not diminishing their relevance to larger problems. Even such a small number of phenotypes results in large genotype spaces for the two mappings, the CA mapping produces 2^{72} genotypes a number that rises to 2^{144} for the RBN mapping. Each of these genotypes maps on to one of the 256 phenotypes and hence the genotype spaces are sub-divided into 256 neutral networks representing each of the possible phenotypes. In this section, the extent and inter-connectivity of these neutral networks is assessed using the random neutral walk. All the statistics presented below were generated from 100 independent walks on each of the 256 neutral networks resulting in a total of 25,600 walks for each mapping.

4.3.2.1 Average phenotype accessibility

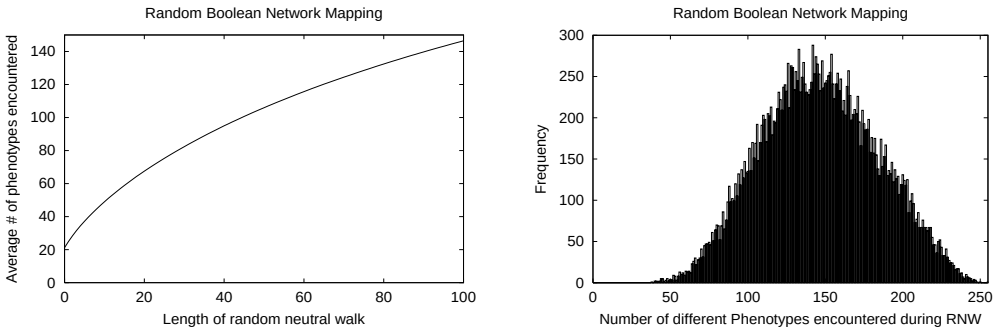
The large genotype spaces created by both the CA and RBN mappings make exhaustive enumeration and hence an exact measure of the average phenotypic accessibility impossible. However, statistics collected on the neutral walks allow a good approximation to be obtained. These statistics are shown in Figure 4.8.



(a) Direct encoding



(b) CA mapping



(c) RBN mapping

Figure 4.8: The number of new phenotypes found on a series of 25,600 random neutral walks for each mapping. The graphs to the left show the average number of phenotypes found as a function of walk length. The histograms to the right bin the number of phenotypes reachable after 100 steps of each neutral walk.

As would be expected, the direct encoding resulted in the discovery of exactly 8 new phenotypes on all walks. Each single-point mutation generated a new phenotype and in the absence of neutrality, further moves in genotype space were not possible. In contrast, the CA mapping generated substantial degrees of neutrality; on average $2^{72}/2^{8} = 2^{64}$ genotypes mapped onto each phenotype allowing the discovery of many more phenotypes through neutral drift. The initial high rate of discovery tailored off to become relatively constant towards the end of the walks. The number of phenotypes discovered after 100 steps averaged approximately 100. The random Boolean network generated even more extensive neutrality with $2^{144}/2^{8} = 2^{136}$ genotypes mapping onto each phenotype. This increased level of neutrality was also reflected in the number of phenotypes discovered, after 100 steps of the neutral walk approximately 150 new phenotypes were discovered on average.

These results demonstrate the ability of the two mappings to give access to many more phenotypes than a direct encoding. They also suggest an added benefit from the higher levels of neutrality within the RBN mapping. However, closer examination reveals that this benefit was larger due to the inherent biases of the experimental method. At each step of the neutral walk, every possible mutation was generated and hence the entire neighbourhood of the current genotype was assessed. This is very different to a real evolutionary algorithm in which this neighbourhood is sampled rather than exhaustively enumerated. The 144-bit RBN genotype was double the length of the CA genotype, which consisted of only 72 bits. Thus, at each step of the walk 72 more genotypes were assessed for the RBN mapping which equates to 7,200 over the course of a 100 step walk. The fact that some of these assessments will inevitably be of the same genotype does not remove the inherent advantage gained by a larger genotype.

A more balanced measure of the performance of the two mappings can be obtained by relating the total number of genotypes examined during a walk to the total number of new phenotypes found to yield a probability of a phenotype discovery. Thus, a total of 72*100 = 7,200 genotypes were assessed for the CA mapping. This resulted in the discovery of approximately 100 phenotypes on average and hence a probability of discovery of 100/7,200 = 0.01. For the RBN mapping the probability of phenotype discovery was approximately 150/(144*100) = 0.01. Thus, by this measure the performance of the two mappings was equivalent.

4.3.2.2 Large-scale structure of genotype space

Both the CA and RBN mappings encode not only the initial state of the automaton but also independent rule tables for each cell. It was claimed that this approach was very likely to allow all required phenotypes to be produced. The results presented in the previous section revealed that new phenotypes were discovered but did not show whether there was a bias to any particular sub-set of phenotypes or whether certain phenotypes were never found. In this section, statistics are presented that aim to give an insight into whether such biases exist.

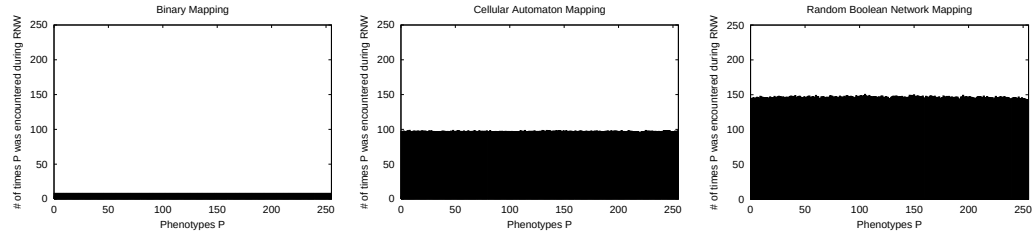


Figure 4.9: The average number of times each phenotype was discovered on the random neutral walks.

Figure 4.9 shows the average number of times that each phenotype was discovered on the random neutral walks. In each case, the average for each phenotype was equivalent to the overall average number of discoveries presented in the previous section. Thus, for the CA mapping each phenotype was discovered approximately 100 times and for the RBN mapping approximately 150 times. These results indicate that there was no dominant sub-set of phenotypes that were discovered more frequently than others; across all the neutral walks the discovery of each phenotype was equally likely and no biases were apparent.

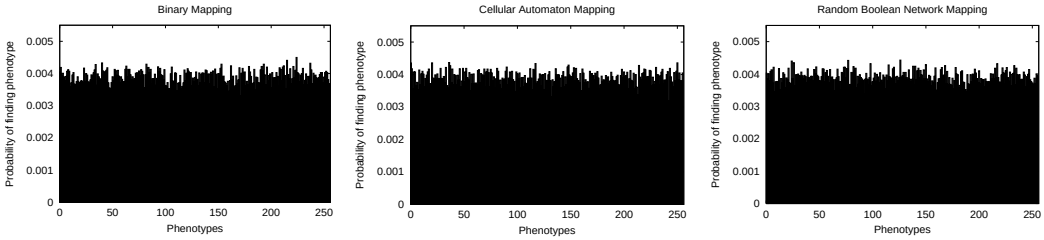


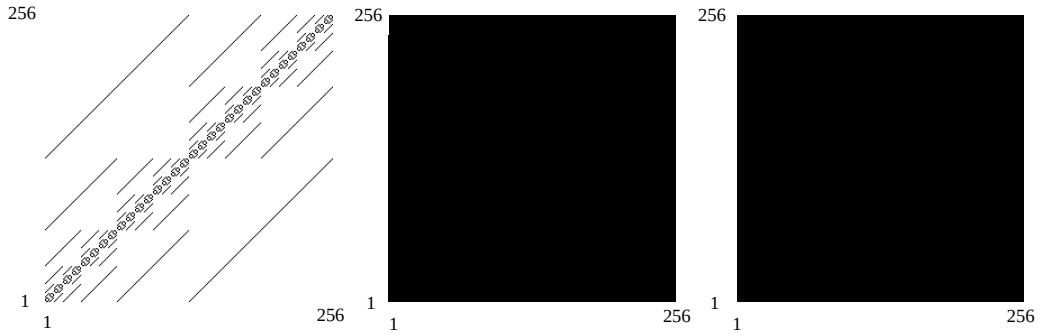
Figure 4.10: The phenotypes discovered by 100,000 random samples of genotype space for each mapping.

Figure 4.10 presents a further independent set of statistics that were generated by taking 100,000 random samples of genotype space. The figure shows the number of phenotypes that were discovered for each mapping. The results for both the CA and RBN mappings are qualitatively similar to those for the direct encoding for which it is known that each phenotype is equally likely to result from a random sample. The results also suggest therefore that there was no inherent bias to any particular sub-set of phenotypes and support those from the neutral walks.

4.3.2.3 Local structure of genotype space

The results presented above give an insight into the global properties of genotype space. In this section, a greater insight into the local properties of genotype space is developed. Although new phenotypes may be readily discovered it may be the case that certain phenotypes can never be reached from some neutral networks, which may be detrimental to the search process. In order to determine whether this was the case, accessibility plots were generated. These plots indicate

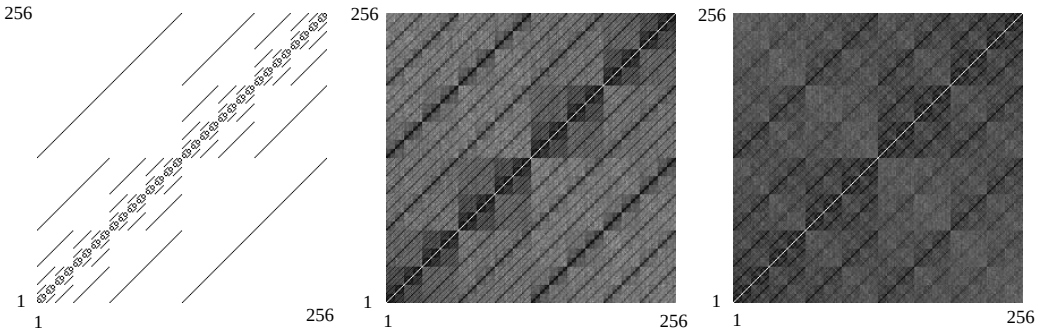
exactly which phenotypes were found during the neutral walks on each of the 256 neutral networks and are shown in Figure 4.11.



(a) Direct encoding (b) CA mapping (c) RBN mapping

Figure 4.11: Inter-phenotype accessibility for each mapping. The horizontal axis gives the phenotype for which neutral walks were performed and the vertical axis the phenotypes discovered on those walks.

As would be expected, the inter-phenotype accessibility is very low for the direct encoding. Only 8 new phenotypes were accessible from any given phenotype. This is in stark contrast to both the CA and RBN mappings for which the accessibility plots were fully populated, indicating that every phenotype was accessible from each of the neutral networks. Thus, local optima are not present in these landscapes as neutral drift on each neutral network allows access to all other neutral networks. However, it is important to determine not only the theoretical possibility of transitions between neutral networks but also the *probability* of those transitions as the most probable transitions will have a greater impact on the evolutionary process. Figure 4.12 enhances the accessibility plot to indicate the frequency with which phenotypes were discovered. This is achieved by adopting a grey-scale, the darker the data point the more frequent the discovery. This approach was introduced by Bullock [107].



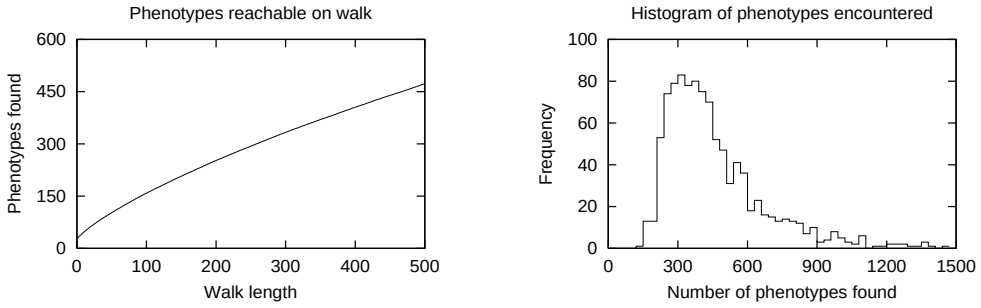
(a) Direct (b) CA (c) RBN

Figure 4.12: Grey-scaled inter-phenotype accessibility for each mapping. The shade of a data point indicates the frequency with which the associated phenotype was discovered; the darker the shade the more frequent the discovery. The shading is log-scaled.

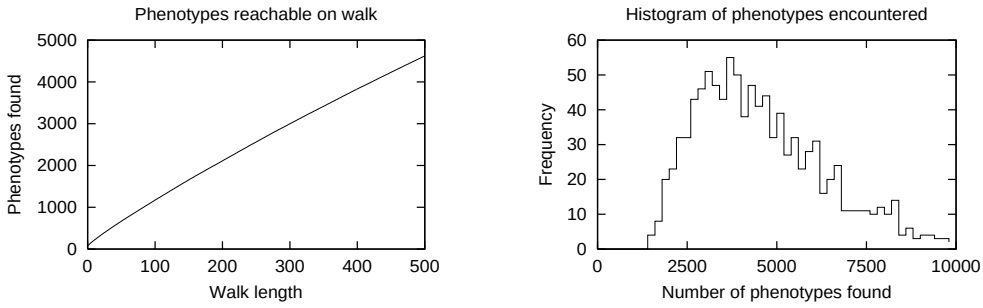
The figure shows that although all transitions are possible for the two mappings, they are not equally probable. The most probable transitions appear to be influenced by the transitions that would have been possible using a direct encoding. Thus although no large-scale structure or biases are apparent in the landscape, localised structure is apparent. This structure may have an influence on the performance of an evolutionary algorithm.

4.3.3 16-bit Phenotype Spaces

In order to gain some insight into the scalability of the two mappings, a further set of experiments were performed for 16-bit phenotype spaces. This increased the number of phenotypes from 2^8 = 256 to 2^{16} = 65,536 and resulted in a corresponding increase in the size of genotype space. The CA mapping required a genotype of length 144-bits and hence resulted in 2^{144} genotypes, whereas the RBN genotype was 336-bits long resulting in 2^{336} genotypes. A total of 1,024 independent random neutral walks of length 500 were performed for each mapping and the average number of phenotypes discovered is shown in Figure 4.13.



(a) CA



(b) RBN

Figure 4.13: Phenotype discovery on a series of 1,024 independent random neutral walks for 16-bit phenotype spaces.

The figure reveals that both mappings again allowed the discovery of large numbers of phenotypes through neutral drift and that the rate of phenotype discovery was relatively constant

throughout the walk. However in this case, the difference between the CA and RBN mappings was more pronounced. Approximately 450 phenotypes were discovered for the CA mapping whereas the number discovered for the RBN mapping was an order of magnitude greater at approximately 4500. This was partly due to the fact that the length of the walk was longer in this experiment and hence any difference between the two mappings was accentuated. However, it was again also due to the inherent biases of the experimental method. The RBN genotype was 192 bits longer than that of the CA and thus 192*500=96,000 more genotypes were assessed during a neutral walk for the RBN mapping.

A fairer comparison can again be made by calculating the probability of discovering new phenotypes by relating the total number of genotypes assessed to the number of phenotypes discovered. For the CA mapping this probability was approximately 450/(144*500) = 0.06 and the for the RBN mapping 4500/(336*500) = 0.03. Thus, despite the increased *possibility* of discovering new phenotypes with the RBN mapping the *probability* of doing so was less than for the CA mapping for these larger spaces.

4.4 Adaptive Fitness Walks

The previous section presented evidence showing that both the CA and RBN mappings allowed the discovery of many more phenotypes than were possible with a more traditional direct encoding. However, the discovery of new phenotypes alone is not sufficient to provide an advantage to an evolutionary algorithm. For this the mappings must allow access to new phenotypes of *higher fitness*. Their ability to do so is assessed in this section through use of similar walks to those of the previous section but with the distinction that moves to higher fitness phenotypes are taken whenever possible.

In order to determine which phenotypes are of a higher fitness, a fitness landscape must be defined. Two different landscapes were explored, the first of which assigned fitnesses to phenotypes according to a random distribution and the second according to the hierarchical if and only if (h-iff) fitness function proposed by Watson et al. [88-91]. In both cases 16-bit phenotypes were used resulting in a total of 65,536 phenotypes.

4.4.1 Random fitness landscape experiments

4.4.1.1 The fitness landscape

In this set of experiments each of the 65,536 phenotypes were randomly assigned a fitness value in the range [0, 1] with higher values denoting higher fitnesses. However, the random numbers were generated so as to create a landscape with many more low fitness values than high. This non-uniformity mirrors that of many real problems in which poor solutions are far more numerous than good solutions. It also increased the difficulty of the problem so that any

performance differences between the two mappings were accentuated. The fitness assignment was performed according to the following equation:

$$f = e^{100(r-1)} \qquad \textit{Equation 4.5}$$

Where r is a random number in the range [0, 1] drawn from a uniform distribution.

4.4.1.2 The adaptive fitness walk

The walk used in these experiments was very similar to the random neutral walk with the exception that steps were taken to a neighbouring genotype of higher fitness whenever possible. If a higher fitness neighbour did not exist, a neutral neighbour was chosen at random as before. The process is summarised below:

1. Randomly choose a genotype G_0 mapping onto a phenotype with fitness F_0
2. Generate all single-point mutants of G_0 and store in list M_0
3. Generate a list N_0 consisting of all members of M_0 that map onto phenotypes of fitnesses greater than or equal to F_0
4. Determine the genotype G_{max} within N_0 that has the highest fitness F_{max}
5. If no genotypes are of a higher fitness then randomly assign a member of N_0 to G_{max}
6. Make $G_0 = G_{max}$ and $F_0 = F_{max}$
7. Repeat from step 2 until a given number of steps have been performed.

4.4.1.3 Results

A total of 1,024 independent adaptive fitness walks of 500 steps were performed for a direct encoding, the CA mapping and the RBN mapping. The results, shown in Figure 4.14, reveal that adaptive fitness walks using the direct encoding quickly become trapped at local optima; no improvements in fitness were achieved following the first few steps. In contrast, both the CA and RBN mappings allowed the discovery of higher fitness phenotypes throughout the walk. The increased phenotypic accessibility afforded by the RBN mapping allowed high fitnesses to be readily achieved. However, the performance of the RBN mapping was again aided by the experimental methods. As with the random neutral walks, the entire neighbourhood of the current genotype was assessed at each step which favoured mappings that generated an increased possibility of phenotype discovery through the use of larger genotypes. As was highlighted previously, it is the probability of phenotype discovery rather than the theoretical possibility that is likely to be of greater importance in any real evolutionary algorithm. Nonetheless, these results demonstrate that both mappings not only allow access to a greater number of phenotypes but also access to phenotypes of higher fitness for this landscape.

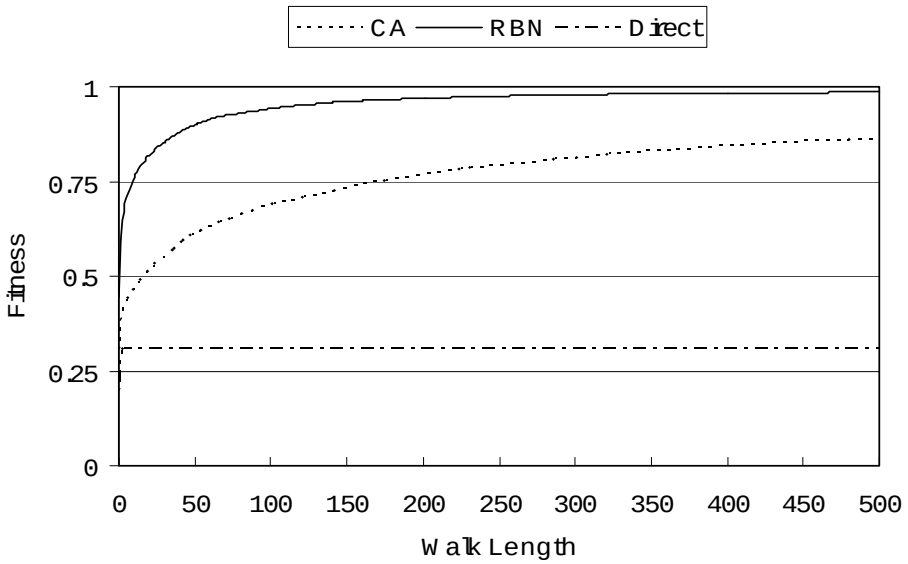


Figure 4.14: Adaptive fitness walks for the CA, RBN and direct mappings with a random fitness landscape. The direct encoding quickly becomes trapped at local optima and makes no further improvements during the walk. However, both the CA and RBN mappings allow higher fitnesses to be achieved. These results are averaged over 1,024 independent walks.

4.4.2 H-IFF fitness landscape experiments

4.4.2.1 The fitness landscape

The "hierarchical if and only if" fitness function was developed as an example of a fitness function that models hierarchical interdependency between a number of building blocks [88-91]. H-IFF decomposes a genotype into a number of layers which are composed of building blocks from lower layers within the hierarchy. This process is controlled by two parameters; *k* indicating the number of sub-blocks within each block and *p* indicating the number of layers of hierarchy. Each building block contributes to the overall fitness if all bits within it are of the same value. An example is shown in Figure 4.15.

It can be seen from the figure that while each block can maximise its fitness independently at a given layer of the hierarchy, its total contribution to fitness is dependent on the state of neighbouring blocks and ultimately the state of the whole genotype. This process is formalised in the recursive equation below where $n = k^p$:

$$f(b_1,\ldots,b_n) = \begin{cases} 1 & \text{if } (n=1) \\ n + f(b_1,\ldots,b_{n/2}) + f(b_{n/2+1},\ldots,b_n) & \text{if } (n>1) \text{ and } (\forall i{:}b_i = 0 \text{ or } \forall i{:}b_i = 1) \\ f(b_1,\ldots,b_{n/2}) + f(b_{n/2+1},\ldots,b_n) & \text{otherwise} \end{cases}$$

Equation 4.6

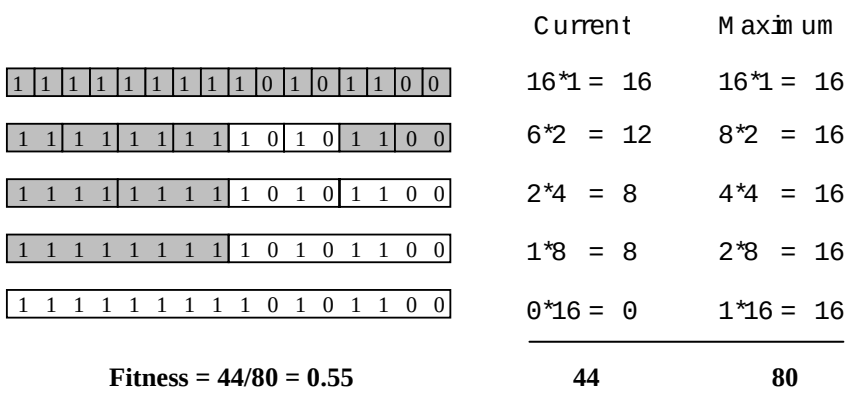


Figure 4.15: An example fitness calculation using the H-IFF fitness function. In this example the number of sub-blocks in each block, *k* is 2 and the number levels of hierarchy, *p* is 4. A fitness contribution is made when all values within a block are the same value.

The H-IFF fitness function results in two maximally distinct global optima in which all values are either 1 or 0. It also results in a number of local optima that correspond to situations in which adjacent sub-blocks are individually aligned but with opposite states. A feature of these local optima is that they are maximally distant from a higher fitness value. For example, the local optima with the highest fitness consist of the two genotypes with half-zeros and half-ones. Fully half of the genotype must be changed in order to reach the next highest fitness level, which in this case are the two global optima consisting of all-ones and all-zeroes. H-IFF is therefore very difficult to solve for a hill-climbing search process and is a good test of the neutral mappings. The nature of the local optima can be emphasised by taking a cut through the landscape as shown in Figure 4.16.

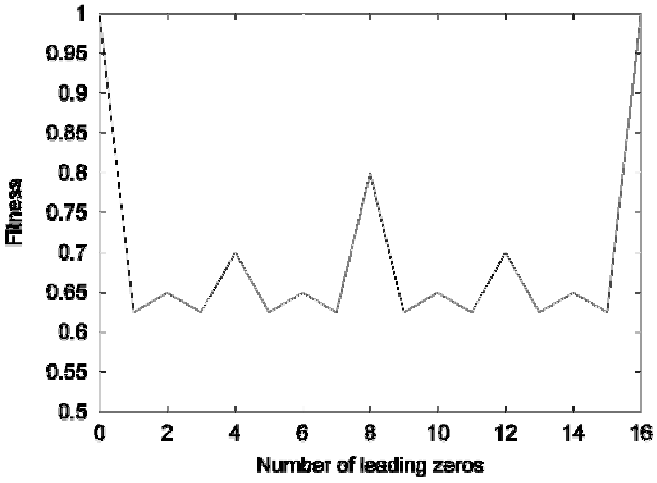


Figure 4.16: A cut through the H-IFF fitness landscape for a 16-bit genotype. A number of different genotypes are shown consisting of a series of leading zeros followed by ones. Both global optima are represented together with a number of local optima that are maximally distant from higher fitness values.

4.4.2.2 The adaptive fitness walk

The adaptive walk used for these experiments removed the inherent bias of the walks performed for the random landscapes which favoured larger genotypes. This was achieved by more closely mirroring a mutation-based evolutionary algorithm. Rather than exhaustively enumerating a genotype's neighbourhood and choosing the neighbour with the highest fitness, a single neighbour was chosen at random independently from its fitness i.e. a mutation was performed. This neighbour was retained if its fitness was greater than or equal to the original genotype. The process is outlined below:

1. Randomly choose a genotype G_0 mapping onto a phenotype with fitness F_0
2. Mutate G_0 to generate a single-point neighbour G_1 and calculate its fitness F_1
3. If F_1 is greater than or equal to F_0 then make $G_0 = G_1$
4. Repeat from step 2 until a given number of steps have been performed.

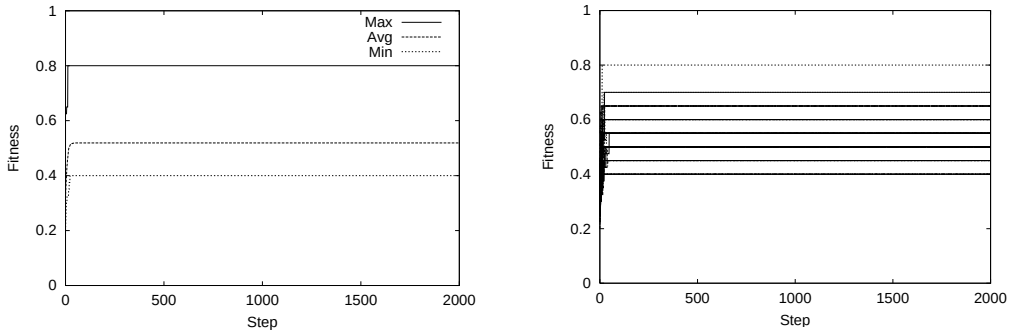
4.4.2.3 Results

A series of 100 adaptive fitness walks were performed for the CA, RBN and direct mappings using the H-IFF fitness function. Each of these walks consisted of 2000 steps. The results, shown in Figure 4.17, reveal that adaptive walks using a direct encoding quickly became trapped at local optima as for the random landscape. Small initial improvements were evident but progress quickly halted. However, both the CA and RBN mappings again allowed the attainment of higher fitness values and in many cases a global optimum was discovered. These results demonstrate therefore that the use of the CA and RBN mappings allowed a very difficult fitness landscape to be negotiated by a simple hill-climbing search algorithm.

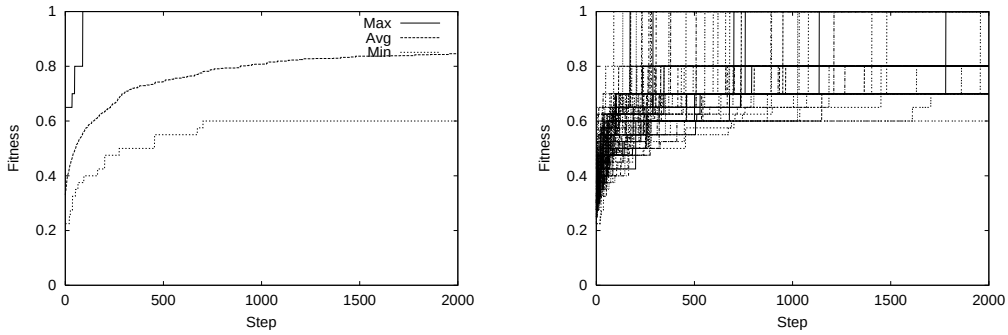
In previous experiments the RBN mapping seemingly outperformed the CA mapping, in this case however the reverse was true. This was a consequence of the revised adaptive fitness walk which removed any biases towards larger genotypes. During the random neutral walks it was discovered that although the RBN mapping increased the possibility of finding new phenotypes, the probability of doing so was less than that of the CA mapping. In the revised adaptive walk, a random neighbour was chosen at each step in the absence of complete knowledge of a genotype's neighbourhood. In this case therefore, the importance of the probability of phenotype discovery overrode the theoretical possibility and this was reflected in the results.

This highlights an important point when designing neutral mappings. Although sufficient neutrality is required to generate extensive and percolating neutral networks, too much neutrality may decrease the likelihood of discovering adaptive mutations. As the number of neutral neighbours increases, the likelihood of a mutation generating a different phenotype decreases i.e.

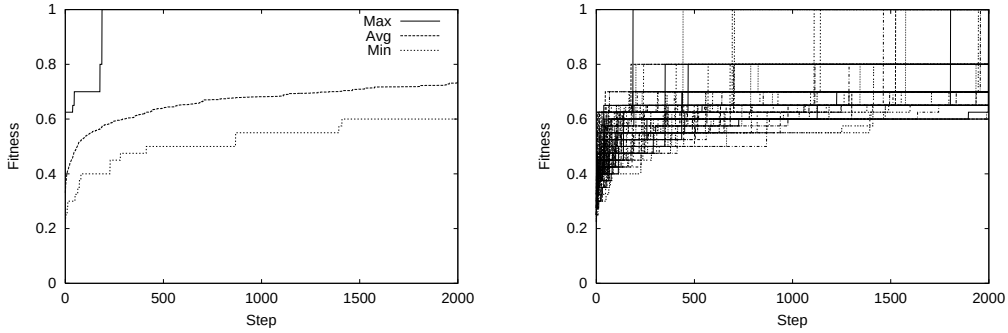
the neutrality produces mutational robustness. During these experiments it was found that 58% of neighbours were neutral for the RBN mapping on average whereas only 44% were neutral for the CA mapping. The latter produced a more effective balance between neutral and non-neutral mutations in this case.



(a) Direct



(b) CA



(c) RBN

Figure 4.17: Adaptive fitness walks for each mapping using the H-IFF fitness function. The left-hand graphs show the average, minimum and maximum fitnesses achieved over 100 independent walks and the right-hand side graphs show the performance of each individual walk.

4.5 Discussion

The results presented above show that use of the CA and RBN mappings can increase accessibility of phenotypes and hence allow higher fitnesses to be reached using a simple hill-climbing algorithm. While the use of the mappings improved the performance of the hill-climber on several abstract problems, this does not in itself show that the mappings would be of use on larger scale problems using a realistic evolutionary algorithm. One of the key simplifications made in the above analysis was the consideration of only single-point mutations. While these mutations are common in many evolutionary algorithms, multiple changes to the genotype are also typically possible. The allowance of greater movement in genotype space changes the neighbourhood relationships and may allow shallow local optima to be avoided. This effect may substantially improve the performance of the direct encoding.

In a recent paper, Knowles and Watson compared the RBN mapping with a direct encoding on various optimisation problems with varying mutation rates [46]. This work suggested that the RBN mapping did not significantly improve the performance of either a hill-climber or a mutation-based evolutionary algorithm when an appropriate mutation rate was used. One example that was investigated was the H-IFF fitness function with 32 and 64-bit phenotypes. It was found that while the performance of the RBN mapping was relatively consistent with different mutation rates its performance was not as good as a direct encoding using high mutation rates and elitism i.e. keeping the best solution found to date. In the extreme, increasing the mutation rate effectively reduces to random search. However, smaller increases in mutation rate may allow exploitation of local correlations and larger movement in genotype space when necessary to negotiate shallow local optima. The results presented by Knowles and Watson suggested that this was indeed the case. However, the relative performance of the two approaches may have also been influenced by the choice of mutation rates. Relatively large mutation rates are required to compensate for the mutational robustness afforded by the RBN mapping. Those chosen for these experiments may not have been high enough to fully exploit the capability of the RBN mapping. Nonetheless, there are deeper reasons for the relatively poor performance of the RBN mapping for these larger spaces which can be elucidated by returning to the original inspiration for the mappings; RNA folding.

The studies of RNA folding described in detail in chapter 2 revealed four properties of the associated genotype-phenotype mappings; large-scale neutrality, neutral networks, shape space covering and common phenotypes. It is clear from the above results that the proposed mappings generated large-scale neutrality and neutral networks; many genotypes mapped onto each phenotype and random neutral walks resulted in a constant innovation of phenotypes. In addition, these walks were not restricted to isolated areas of genotype space but traversed significant amounts of the space. For the 8-bit phenotype spaces, an average 44% of the CA genotype and 34% of the RBN genotype had changed by the end of the walk. The results also reveal that the

mappings exhibit shape space covering, which refers to the property that all phenotypes are accessible from any arbitrary location in genotype space. All phenotypes were discovered with a series of random samples and on the random neutral walks. However, these same statistics reveal that the mappings did not exhibit the final property of common phenotypes as all phenotypes were discovered with equal probability. This would be expected as the mappings were designed to remove any biases in genotype space. However, the lack of common phenotypes ultimately limits the usefulness of the mappings.

If all phenotypes were equally likely discoveries from any location in genotype space, the evolutionary process would reduce to random search with elitism or equivalently the use of a very high mutation rate with elitism. At each step along a neutral network, all other neutral networks would be equally probable discoveries. This may be beneficial for small spaces but as the size of the problem and hence the number of neutral networks increased, the likelihood of discovering a required neutral network would diminish. To be of benefit the mappings must exhibit *some* structure. The accessibility plots revealed that localised structure was apparent in genotype space; some phenotypes were more commonly discovered from a given neutral network. However, these common phenotypes were heavily influenced by the phenotypes that would have been accessible using a direct encoding and thus the structure in genotype space was still heavily influenced by the original neighbourhood relationships in phenotype space.

To gain full advantage of a neutral genotype-phenotype mapping it is likely that a more fundamental restructuring of genotype space is required which produces common phenotypes by introducing biases into genotype space. Such biases would make certain phenotypes more likely outcomes of evolution. Natural evolution has been able to build on whatever phenotypic biases are produced by the laws of physics and chemistry, which produce common RNA structures for example. However, in artificial mappings these biases must be introduced so that the common phenotypes tend to be those of relatively high fitness. This is likely to require *a priori* knowledge about the problem. The use of abstract neutral mappings for arbitrary problems is thus not likely to be a fruitful approach in the long run. The mappings will need to be designed in the context of a particular application. This is the subject of the remainder of this work.

4.6 Summary

In this chapter two abstract genotype-phenotype mappings have been explored that are based on generic models of natural self-organising processes; the cellular automaton and the random Boolean network. The key findings are highlighted below:

- Both the CA and RBN mappings exhibit large-scale neutrality and neutral networks.
- A random neutral walk on these neutral networks allows the discovery of many more phenotypes than would be possible using a direct encoding.

- The increased accessibility of phenotypes allows higher fitnesses to be reached in both a random fitness landscape and the “hierarchical if and only if” landscape.
- The mappings give equal emphasis to each phenotype and thus do not exhibit the property of common phenotypes observed in the RNA-folding landscapes described in chapter 2. There was no bias in the search space in favour of certain phenotypes.
- Such a property is necessary to gain full advantage from a neutral mapping and is likely to require the use of domain knowledge to encourage the common phenotypes to be of high fitness.

The following chapter builds on the knowledge gained in this chapter to develop a mapping for a telecommunications network design problem. This mapping uses domain knowledge with a view to biasing the search space in favour of high-fitness phenotypes. The resulting search space is extensively analysed to ascertain whether this was achieved.

Chapter 5

Growing Telecommunication Networks

5.1 Introduction

In the previous chapter it was concluded that to gain full advantage of neutrality, genotype-phenotype mappings must be designed in the context of a particular application. The use of domain knowledge opens up the possibility of producing common phenotypes that tend to be of high fitness. This would be highly advantageous as the common phenotypes are more likely outcomes of the evolutionary search process. In this chapter, domain knowledge is used to design a self-organising mapping for a specific real-world application with a view to generating such common phenotypes. This application involves the design of telecommunications networks.

Evolutionary computation has been widely used within the telecommunications domain. The work dates back to the late 1980's and tackles numerous problems including frequency assignment, call allocation, routing, node location and topology design. A thorough review of this work can be found in [61,62]. The focus of this chapter is on node location, which involves discovering the best locations for nodes from a set of potential sites. A common approach in the application of evolutionary computation to the node location problem is to use an encoding in which the locations of the nodes are directly specified in the genotype. An example of this approach is the work of Routen who addressed the problem of concentrator location and assignment [122]. A concentrator accumulates traffic from a number of different sites so as to lessen the number of independent connections to the network. Routen employed a two-stage process in which one genotype defined the location of concentrators and another genotype encoded the allocation of sites to those concentrators. This approach is a classic example of a direct encoding, the network design can be interpreted directly from the genotype and no further information is required. Examples of similar approaches can be found in the work of Chardaire et al. [73], Celli et al. [24], Gondim [78] and Calégari et al. [72].

In this chapter, an alternative approach is developed in which the genotype does not directly specify the configuration of the network but rather instructions for creating that network. The final network design is the result of a self-organising developmental process that is tailored by the information encoded in the genotype rather than controlled by it. This approach is described more fully in the following section. By way of example, the approach is then applied to a simplified version of a real problem; the growth of the UK's data network. The mapping designed for this

application introduces significant neutrality in to the genotype-phenotype mapping as many different instructions produce the same network design. The impact of this neutrality is analysed in detail in the context of an example evolutionary run and through exhaustive enumeration of genotype space.

5.2 Coupling evolution and development for network design

In practice, real telecommunication networks are often built according to various planning rules that embody the accumulated experience of network designers. These planning rules typically take the form of a series of IF-THEN rules that define actions to be taken given certain conditions. For example, a rule may place a new node at a given site based on that site's level of demand as below:

IF (demand > x) THEN (add node to site)

Where x is a threshold at which the site's demand is deemed high enough to warrant the addition of a node. Designing a network would thus consist of applying this planning rule to each of the available sites. This example is obviously a gross simplification, any realistic design problem would require a number of different planning rules that take into account many conditions and define a number of possible actions. However the same principles apply, the state of the current network and its environment is assessed and the planning rules define the actions to be taken given these conditions.

It is possible to aid network designers in this process by automating the application of the design rules. In this approach, both the current network and its environment are simulated such that the relevant conditions can be assessed and actions taken. The network is automatically produced through application of the planning rules in the context of this simulation. The rules typically consider localised conditions and take localised actions and it is their combined action that produces the final network design. In effect, the planning rules define a self-organising developmental process. The network is "grown" under the influence of the self-organising dynamics rather than precisely specified, which has analogues to natural development as shown in Figure 5.1. A concrete example is given in the following section.

In order to evolve networks therefore, the parameters of the developmental process can be encoded in the genotype rather than the network design itself. For example in the rule shown above, the threshold parameter x may be genetically encoded. Evolution is able to influence the network design through tailoring the parameters and the resulting dynamics of the developmental process but must always work within the constraints that it imposes; it does not have a free hand to create arbitrary network designs. Good networks are produced using both the knowledge

gained during evolution in conjunction with that inherent to the growth process. This approach was developed by Shipman et al [17,96,97,100].

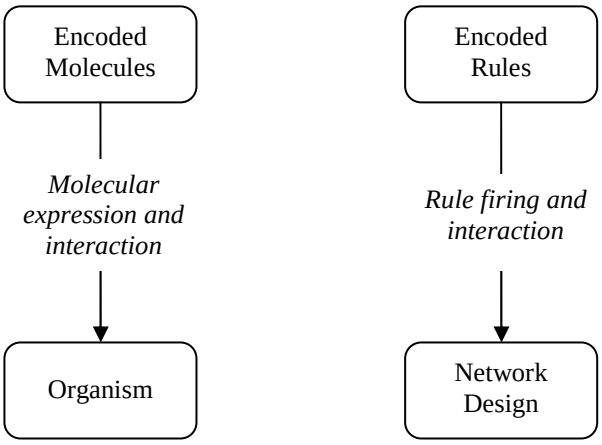


Figure 5.1: The natural analogy for this approach to evolutionary network design. Right, planning rules are encoded and interpreted in the context of a simulated environment to produce a network design. Left, the analogue in nature is the encoding of molecules that are expressed and interact during development to form the organism.

There are a number of advantages to this approach. Expert knowledge can easily be captured in the form of planning rules and used to seed the evolutionary process. The resulting rules are in a form familiar to network designers and can be used in conjunction with other network design tools. It is also possible to evolve the structure of the planning rules together with their parameters. This allows new planning rules to be evolved that were not previously known to the network designers.

In addition to these more pragmatic considerations, the introduction of a self-organising developmental process has a significant impact on the nature of the search space. The mapping from genotype (encoded rules) to phenotype (network design) typically contains large-scale neutrality as many different rule sets give rise to the same network design. It is this feature that is of primary concern in this work and is analysed in detail in the context of a specific application; the growth of the UK data network.

5.3 Growing the UK data network

As demand for telecommunication services increases, the networks designed to accommodate this demand must be grown. In particular, the explosive growth of the Internet has necessitated large-scale growth of the networks dedicated to data traffic. This is a challenging problem for network designers involving many competing constraints and the use of automated techniques to

aid in the process would be highly desirable. This problem thus provides a good test bed for the techniques proposed in the previous section. In order to apply these techniques to this data network problem, both a simulation of the network and a developmental process must be defined. These are the subject of the following two sections.

5.3.1 The network simulation

In order to design the real UK data network, a very complex network simulation would be required. However, the aim of this chapter is not to produce network designs that will actually be deployed but to analyse the nature of the search space created by the developmental process. For these purposes, a greatly simplified network simulation is sufficient.

Figure 5.2: The initial data network consisting of 3 core nodes and 20 potential sites for access nodes. The total demand (indicated by the fill of a box) is greater than can be handled by the existing network and thus it must be grown.

Figure 5.2 shows an early manifestation of the data network within the UK. This network consists of a fully meshed core of three high-capacity nodes situated in London, Manchester and Birmingham. All data traffic originates at exchange sites around the country and is transmitted via the standard public switched telephone network (PSTN) to one of these nodes. Twenty sites are shown in the figure with the level of data traffic originating at that site indicated by the fill of the box and detailed in Table 5.1. The amount of data traffic in the environment is more than can be handled by the existing core network and it must therefore be grown. The network design task consists of placing access nodes (concentrators) at appropriate sites in order to satisfy more demand and alleviate the strain on the PSTN. When an access node is placed at a site, it is connected directly to the nearest core node and hence the network topology is fixed.

Table 5.1: The demand for data traffic at the 23 sites of the sample application together with their co-ordinates on a 100x100 plane. All demands fall within the range 0-1000.

Site	Co-ordinate	Demand
Aberdeen	61, 25	100
Belfast	36, 49	600
Birmingham	65, 73	900
Brighton	75, 89	100
Bristol	63, 82	250
Cambridge	77, 78	100
Cardiff	57, 82	400
Edinburgh	56, 38	500
Exeter	53, 92	200
Glasgow	51, 39	400
Ipswich	81, 81	100
Leeds	67, 61	600
Liverpool	59, 63	500
London	77, 84	1000
Manchester	64, 64	900
Middlesbrough	67, 51	300
Newcastle	65, 47	400
Norwich	84, 72	200
Nottingham	70, 69	300
Plymouth	49, 94	100
Reading	71, 85	200
Sheffield	68, 65	600
Southampton	67, 90	200

The above environment allows both core and access nodes to be placed at appropriate sites and connected to form a network. The addition of this hardware results in costs being incurred. However, in addition to hardware costs the ability of the resulting network to handle the demand and alleviate PSTN strain must also be ascertained. In order to achieve this, a simple call-handling procedure was simulated and is detailed below:

1. All demand at a site without a collocated node is routed via the PSTN to the nearest node. This incurs a cost for the PSTN call.
2. Demand at a site with a collocated access node is handled by that node if there is available capacity.

3. Any remaining demand at a site with a collocated access node is routed to the nearest core node via the PSTN. This incurs a further cost for the PSTN call.
4. Any demand at sites with a collocated core node is handled by that node if there is available capacity.
5. Any remaining demand at a site with a collocated core node is dropped. This incurs a cost for each call that is dropped.

This call-handling procedure requires the capacities of the two types of node to be specified, the specific values used for thus simulation are given in Table 5.2.

Table 5.2: Parameters of the network nodes.

Node type	Capacity
Access	1000
Core	2000

5.3.2 The DevelopmentalProcess

The developmental process was created through use of a single planning rule that defines the conditions under which sites become candidates to house access nodes. These conditions take into account the demand at a site and its Cartesian distance away from an entry point into the data network, the following values are calculated for each site *i*:

$$\theta\_demand_i = a * (demand_i - b) \qquad \textit{Equation 5.1}$$

$$\theta\_distance_i = c * (distance_i - d) \qquad \textit{Equation 5.2}$$

The parameters *b* and *d* are threshold parameters i.e. the values at which the conditions are satisfied, parameters *a* and *c* are scaling parameters and allow the relative importance of the two conditions to be controlled, $demand_i$ is the demand at site *i* and $distance_i$ is the distance that site *i* is away from the nearest existing node. If both θ_demand_i and $\theta\_distance_i$ are above zero i.e. if both conditions are met then the site becomes a candidate to house an access node:

IF ($\theta\_demand_i > 0$) **AND** ($\theta\_distance_i > 0$)
THEN add site *i* to candidate list *Equation 5.3*

A "firing strength" is then calculated for each site in the candidate list:

$$FiringStrength_i = \theta\_demand_i + \theta\_distance_i \qquad \textit{Equation 5.4}$$

The site with the highest firing strength is chosen to house an access node. The candidate list is then recalculated and the process repeated until the candidate list is empty i.e. no more sites satisfy both planning rule conditions. This iterative process is necessary as the action of a rule affects the calculated conditions i.e. adding a node affects a site's distance to the nearest existing node and thus the order in which nodes are added is important. The scaling parameters *a* and *c* enable the relative importance of the two conditions to be controlled and hence the order in which nodes are added to be modified. If *a* was much higher than *c* then sites with high demand would fire more strongly than those a long way from the existing network and vice versa.

5.3.2.1 An example network development

Consider the following parameter settings for the developmental process defined above: *a*=1, *b*=400, *c*=1, *d*=20. These parameters scale both conditions evenly and make a site a candidate to house an access node when both the demand is greater than 400 and the Cartesian distance to the nearest current node is greater than 20. For this rule, the developmental process would proceed as follows:

1. θ_demand_i and $\theta\_distance_i$ are calculated for each vacant site.
2. A candidate list is built using Equation 5.3.
    - Belfast, Edinburgh, Leeds, Liverpool and Sheffield are the only vacant sites that satisfy the demand criterion as their demands are greater than 400.
    - Leeds, Liverpool and Sheffield are in close proximity to Manchester and therefore do not satisfy the distance criterion. However, Belfast and Edinburgh are a distance of 31.76 and 27.2 respectively away from the closest data node at Manchester. These two sites therefore also satisfy the distance criterion and are added to the candidate list.
3. The firing strength of the two sites in the candidate list is calculated:
    - Belfast FiringStrength = 1*(600-400) + 1*(31.76-20) = **211.76**.
    - Edinburgh FiringStrength = 1*(500-400) + 1*(27.2-20) = **107.2**.
4. Belfast has the highest firing strength and is thus assigned an access node which is connected to the nearest core node at Manchester. See Figure 5.3 (b).
5. The process is repeated and θ_demand_i and $\theta\_distance_i$ calculated for the remaining vacant sites.
6. A new candidate list is calculated.
    - Edinburgh, Leeds, Liverpool and Sheffield are now the only vacant sites that satisfy the demand criterion.

- Leeds, Liverpool and Sheffield are again ruled out due to their close proximity to Manchester.
- The addition of a node at Belfast makes this the closest node to Edinburgh at a distance of 22.84. Edinburgh thus satisfies both criteria and is the only candidate site.

7. Edinburgh is assigned an access node and connected to the nearest core node at Manchester. See Figure 5.3 (c).
8. The process is repeated and θ_demand_i and $\theta\_distance_i$ calculated for the remaining vacant sites.
9. No sites match both criteria and thus the candidate list is empty. The developmental process terminates.

The parameters specified above would therefore result in the addition of two access nodes at Belfast and Edinburgh as shown in Figure 5.3. Different sets of parameters would affect the dynamics of this developmental process and may generate different network designs.

Figure 5.3: The developmental process for the rule IF (demand > 400) AND (distance > 20) THEN (add access node). (a) the initial network (b) an intermediary network with an access node at Belfast (c) the final network design with a further access node at Edinburgh.

5.4 Evolving the Data Network

The environmental context and developmental process defined in the previous section allow networks to be produced given a set of parameters. In this section, an evolutionary algorithm is used to optimise these parameters with the aim of producing high quality network designs. The impact of neutrality on such an evolutionary process is analysed in detail. However, before an evolutionary algorithm can be used, a fitness function and a genetic encoding must first be defined.

5.4.1 The fitness function

In order to assess the quality of the final network design a number of criteria need to be considered that relate to the cost of the deployed network, the strain on the PSTN and a measure of the quality of service that the network provides. For the purposes of this work, the overall fitness was the sum of 3 individual costs:

1. *Hardware* – The cost of the deployed nodes and of linking them to the existing core network.
2. *PSTN call* – The cost of transmitting unsatisfied demand to a node in the data network via the PSTN.
3. *Quality of Service (QoS)* – The perceived quality of service, in this case the measure relates solely to the level of demand that could not be satisfied i.e. the number of dropped calls.

A good network design must strike a balance between these costs. Adding an access node to the network increases the hardware cost but may decrease the number of PSTN calls that are required and allow more demand to be satisfied.

Table 5.3: Costs incurred by a network design.

Type	Symbol	Cost
Access node	**An**	800
Link	**Lk**	25 per unit length
Dropped call	**QoS**	10
PSTN call	**PSTN**	5

The specific costs used in this case are specified in Table 5.3 and allow the fitness of a simulated network *f* to be calculated according to the following equation:

$$f = (n * An) + (l * Lk) + (dc * QoS) + (p * PSTN) \qquad \textit{Equation 5.5}$$

Where *n* is the number of access nodes that have been added to the network, *dc* is the total number of dropped calls and *p* is the total number of PSTN calls. The length of link is calculated as the Cartesian distance between an access node and its closest core node, and *l* calculated as follows:

$$l = \sum_{0}^{n} Cd(Access_n, Core_0) \qquad \textit{Equation 5.6}$$

Where $Access_n$ is the nth access node, $Core_0$ is its closest core node and Cd is the Cartesian distance between these two nodes.

5.4.2 The genetic encoding

A typical direct encoding for this problem would consist of a binary genotype of 20 bits. Each of these bits would represent one of the vacant sites and its value would indicate whether or not that site would house an access node in the final network design. In this case, however, it is the parameters of the developmental process that must be encoded rather than elements of the final network design. There are a number of possible ways in which to encode these parameters each with their own properties. However, for the purposes of this study a binary genotype of length 20 was chosen. The genotype was divided into four 5-bit values that encode the parameters of the developmental process; the demand scaling parameter (*a)*, the demand threshold (*b)*, the distance scaling parameter (*c*) and the distance threshold (*d)*. This encoding is shown in Figure 5.4.

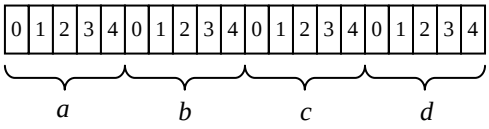


Figure 5.4: The genotype used to encode the parameters of the developmental process. Each of the 4 parameters *a, b, c* and *d* from Equations 5.1 and 5.2 are encoded as 5-bit binary values.

Use of 5-bit binary values allows $2^5 = 32$ distinct values for each parameter. These values can be used to quantise a defined range. The ranges used in this experiment are shown in Table 5.4.

Table 5.4: The ranges of the parameters of the developmental process.

Parameter	Description	Min	Max
a	Demand scaling	0	5
b	Demand threshold	0	1500
c	Distance scaling	0	5
d	Distance threshold	0	50

5.4.3 The evolutionary algorithm

A generational genetic algorithm was used to evolve these parameters in which an entirely new population was created at each generation [12]. A population of 20 genotypes was used and both per-bit mutation and single-point crossover employed to generate new individuals. The parents were chosen using roulette wheel selection, which selects parents with a probability that is dependent on their fitness. The parameters for this algorithm are given in Table 5.5.

Table 5.5: Parameters for the initial evolutionary algorithm

Parameter	Value
Population size	20
Mutation rate	0.01
Crossover rate	0.7

5.4.4 An example evolutionary run

Figure 5.5 shows a plot of the evolutionary process over 1180 generations. All encoded rule sets initially produced high cost networks as would be expected for a random set of parameters. However, better quality rules were quickly discovered and a relatively low cost network was produced within the first 76 generations. This network was not improved upon for around a thousand generations until two better quality networks were discovered in relatively close succession. The figure reveals a punctuated equilibrium dynamic with periods of no gain followed by rapid jumps to higher quality networks.

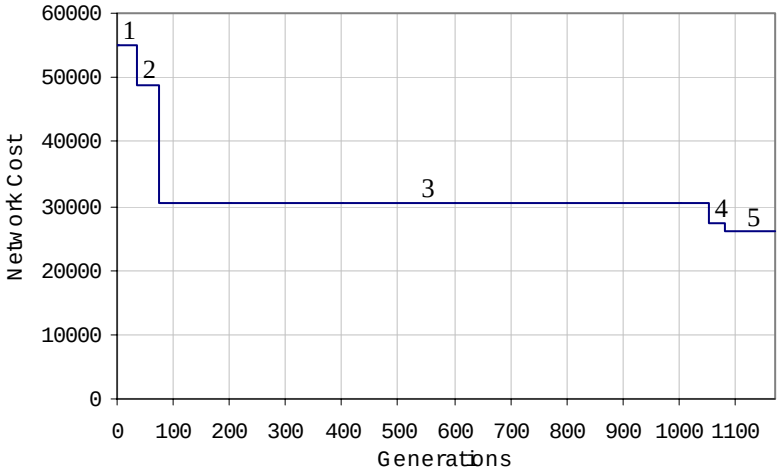


Figure 5.5: An example evolutionary run over 1180 generations. A series of 5 fitness plateaus are identifiable corresponding to a series of 5 different network designs, 1-5.

The final network design was the endpoint of a series of 5 networks that correspond to the plateaus in the evolutionary plot. These networks are visualised in Figure 5.6.

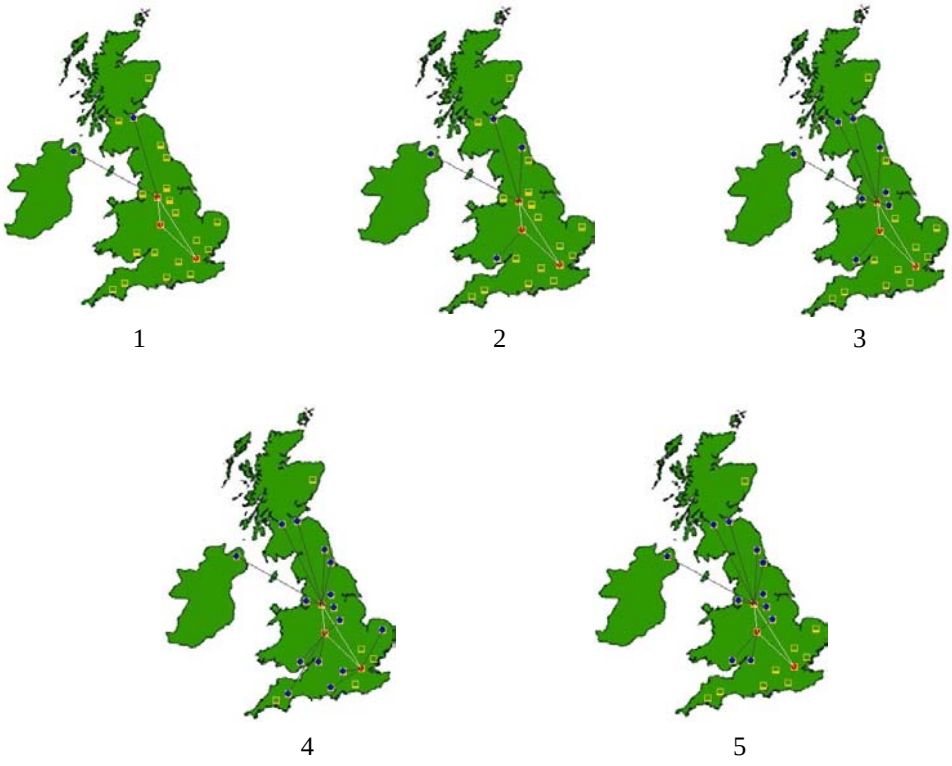


Figure 5.6: The series of best networks produced during the sample evolutionary run. The networks 1-5 correspond to the five fitness plateaus in Figure 5.5.

The parameters of the planning rules that created each of the five networks are shown in Table 5.6. These parameters relate to the first discovery of the network i.e. the start of each plateau.

Table 5.6: The rule parameters that generated each of the 5 networks shown in Figure 5.6. These parameters relate to the first network found on each plateau.

	a Demand Scaling	b Demand Threshold	c Distance Scaling	d Distance Threshold
1	**4.69**	**421.88**	**3.59**	**18.75**
2	**3.24**	**375**	**4.84**	**6.25**
3	**1.88**	**375**	**2.03**	**0**
4	**3.12**	**187.5**	**2.66**	**0**
5	**2.50**	**234.38**	**2.66**	**1.56**

It can be seen from these parameters that the best solution in the initial random population set relatively high thresholds for both the distance and demand conditions. Relatively few sites matched both these criteria and thus the first network contained only two access nodes at Belfast and Edinburgh. It was quickly discovered, however, that a smaller distance threshold allowed lower cost networks. Particular gains were made in the transition from network 2 to 3 as this

threshold was reduced to zero. This allowed the demand to be satisfied in the geographically concentrated population centers of northern England. Further smaller gains were achieved mainly through fine-tuning the demand threshold to find an optimum balance between hardware and quality of service costs. This fine-tuning initially increased the amount of hardware at network 4 until a network design was discovered that reduced the hardware costs by a greater degree than corresponding increases in the cost of PSTN and dropped call costs. Figure 5.7 shows the component costs of the final network, number 5. All demand is satisfied after 8 iterations of the growth process. However, further nodes are added to the network as the reduction in the PSTN cost more than offset the increase in hardware costs.

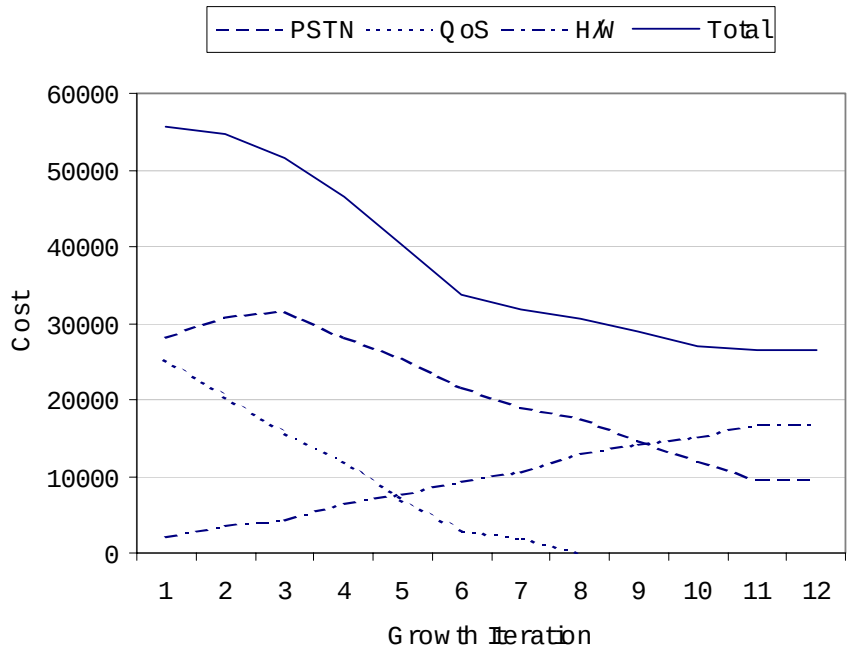


Figure 5.7: The cost of the best network produced by the evolutionary algorithm. In addition to the total cost, the diagram shows the three component costs Quality of Service (QoS), PSTN calls and hardware. The horizontal access shows iterations of the growth process. After the addition of 8 nodes all the demand is satisfied. A further 3 nodes are added, however, which reduced the PSTN cost to a greater degree than the increases in hardware costs.

5.4.5 Useful and useless neutral mutation

The encoding of the rule used to generate the above networks introduced a significant possibility for neutral mutations. For any given value of distance and demand, the only mutations to the threshold parameters that had the potential of changing the network design were those that resulted in a value becoming greater than a threshold it was previously below or vice versa. Many mutations did not have this effect and were thus neutral. In addition, many changes to the scaling parameters did not affect the final network design, as they did not modify the relative strength of rule firing by a great enough degree. However, care needs to be taken when introducing the

possibility for neutral mutation. Although such a possibility would always allow for neutral drift, it would not always allow for *beneficial* neutral drift. In order to decrease the likelihood of entrapment at local optima, neutral drift must allow access to phenotypes that would otherwise have been inaccessible.

A situation where this was not the case can be illustrated by considering the thresholds. In this example the maximum value of the demand threshold was set to 1500 and the maximum distance threshold to 50. However, the maximum possible demand at a site was 1000 and the maximum possible distance away from the existing data network was approximately 39. Thus, it was guaranteed that any modification to the demand threshold that maintained the value within the range 1000 to 1500 would not modify the network design. This is the case as demands greater than 1000 were not present at any site and thus changes to the threshold in this range could not change the set of sites which matched that criteria. Similarly, modifications to the distance threshold that maintained the value within the range 39-50 could not possibly modify the network design. Mutations that maintained the demand threshold in the range 1000-1500 or the distance threshold in the range 39-50 would therefore be neutral. However, these neutral mutations would not be of any benefit in reducing the likelihood of local optima, as they could *never contribute* to the fitness of the phenotype. For any given network design it would make no difference whether the threshold was 1000, 1500 or any value in between. A neutral mutation would only be useful if it produced a value that, given changes elsewhere on the genotype, played a role in the development of the phenotype i.e. a value that had the potential of loosing its neutrality. The neutrality created through increasing the maximum value of these thresholds could not improve evolutionary search but may simply slow it down through encouraging non-beneficial random drift.

5.4.6 Neutrality in the scaling parameters

The neutrality described in the previous section is at one end of a scale - guaranteed to never directly contribute to the fitness of the phenotype. However, neutrality could also be introduced that was potentially useful but very unlikely to be so. For example, the entire genotype could be duplicated and an extra bit employed that determined whether the original or its duplicate would be interpreted. Thus, at any one time an entire (original) genotype would be free to be mutated at will without affecting the current fitness. It is possible that neutral mutations could produce a high quality phenotype that could then be interpreted as the current phenotype through mutation of the "switch". However, this is *highly* unlikely - local optima do not exist in this search space but the transition points between phenotypes (i.e. areas of genotype space in which further increases in fitness could be made) are very sparse. This is akin to randomly drifting along an enormous neutral ridge looking for a single point that allows access to a different phenotype. It is very unlikely that the needle in this haystack would be discovered. To improve evolutionary search we need to encourage the presence of many needles.

An example of where this was not achieved is in the choice of scaling parameters. These two parameters were assigned the ranges 0 to 5 and as already discussed, scaled the contribution of the associated condition to the overall firing strength. However, their effect was likely to be very different for the two conditions. As stated in the previous section, the maximum distance a site could be from the existing network was 39 and the maximum demand at a site was 1000. Thus, the maximum amount the two values could exceed their thresholds was 1000 for the demand threshold but only 39 for the distance threshold. However, the range of their scaling was the same. There was thus an implicit bias in the encoding to place access nodes at sites with high demand over those that were a large distance away from the nearest node. This bias was exacerbated by the fact the placement of sites was such that their actual distance from the existing network was typically much less than the maximum possible distance of 39. It was still possible for these biases to be overcome (with a high distance scaling parameter and a low demand scaling parameter) but the bias made it less likely that mutations to the scaling parameters would have an effect on the overall network design. The encoding of the scaling parameters introduced a significant probability of neutral mutation. However, the bias shifted the balance between neutral mutations and non-neutral mutations too much in favour of the former. Neutral ridges had been formed but the accessibility between them had been reduced; there was too much hay and not enough needles.

The reduced effect of the scaling parameter is highlighted by the data presented in Table 5.6. All the transitions to networks of higher fitness were the result of mutations to the thresholds and not to the scaling parameters. The changes in the scaling parameters during periods of neutral drift between individual transitions were coincidental rather than necessary. However, that is not to say that neutral drift in the scaling parameters was unimportant. Consider the transitions from network 1 to 3 for example. Changes in the scaling parameter were not necessary to allow the transition from network 1 to 2 or the transition from 2 to 3. However, changes to the scaling parameter were necessary to allow the transition from network 1 *to 3*. That is, the scaling parameters used in the rule that generated network 1 could not have been used in the rule that produced network 3. Thus, although neutral drift in the scaling parameters was not important for individual transitions, it was important when considering the *series* of transitions.

5.4.7 Neutrality in the threshold parameters

The threshold parameters were the driving force behind the transitions to higher fitness networks. The transitions from network 2 to 3 and from network 4 to 5 were the result of single mutations. In the former case the mutation reduced the distance threshold from 6.25 to 0. In the latter case the demand threshold was increased from 187.5 to 234.38. Figure 5.5 reveals that these mutations were relatively quickly discovered. The transitions from network 1 to 2 and network 3 to 4 required two mutations. In the former case a single mutation was required to both the demand and

distance thresholds. In the latter case a dual mutation to the demand threshold was required. However, Figure 5.5 shows that they took very different lengths of time to discover. The time taken for the transition from network 1 to 2 was of the same order as that for a single mutation; the transition from 3 to 4 took a great deal longer.

Figure 5.8 shows the effect of all possible mutations on the rule producing network 1 at the start of the plateau. The figure shows that all possible single mutations are either deleterious or neutral. It is not possible to increase the fitness of the network from this point in genotype space. The figure also shows that the two mutations required to transition to network 2 are *individually* neutral - either can be made without affecting the current network. Thus, the scene can be set for the second required mutation, which does increase the fitness of the resulting network. This is a good example of useful neutrality. The individual mutation has no immediate effect on the phenotype but *given changes elsewhere on the genotype* comes to play an important role in its construction.

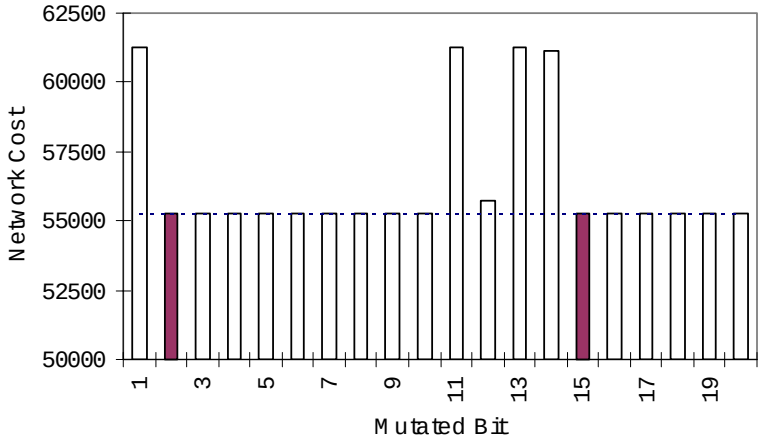


Figure 5.8: The effect on network cost of all possible mutations of the rule producing network 1. No mutations are immediately beneficial, however, the two mutations required to cause a transition to network 2 (shaded columns) are both individually neutral. They can thus be made without affecting the current network in order to "set the scene" for the second mutation. The dashed line shows the cost of network 1.

Figure 5.9 shows the effect of all possible mutations on the rule producing network 3 at the start of the plateau. Again all mutations are either deleterious or neutral. However, in this case both the mutations that are required to produce the transition from network 3 to 4 are individually deleterious. It is thus not possible for these mutations to be made without forgoing the current fitness value – the stage cannot be set for the second mutation. In this case both mutations must be made simultaneously or within short succession before being removed from the population through selection pressure. The probability of this occurrence is very much less than that of a

single mutation, which is reflected in the number of generations that were required for the dual-mutation to occur. In effect, a local optimum has been reached at network 3 and reliance is being made on improbable multiple mutation events to jump an individual over a valley. Note that the jump could potentially also be made via a beneficial crossover event. However, the functional parts of the genotypes in the population were typically highly converged during periods of neutral drift and therefore crossover had little effect.

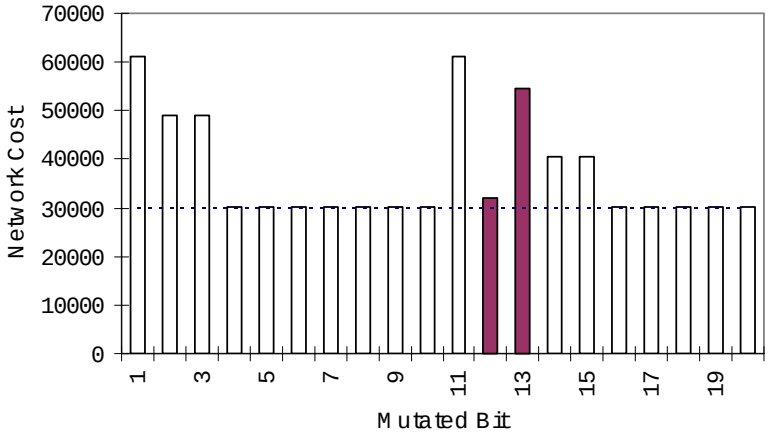


Figure 5.9: The effect on network cost of all possible mutations to the rule producing network 3. Again, no mutations are immediately beneficial. In this case both the mutations required to transition to network 4 (shaded columns) are individually deleterious and cannot be made without forgoing the current fitness value. The dashed line shows the cost of network 3.

The ability of neutral mutations to improve evolutionary search is evident from this example. In the cases were immediate improvement was not possible, the existence of beneficial neutral mutations allowed the search to quickly progress. Their absence caused it to drastically slow and even halt but for unlikely multiple mutation events.

5.5 Exhaustive Enumeration of Genotype Space

In the previous section, the benefit of the neutral mutations allowed for by the developmental process was highlighted in the context of an example evolutionary run. However, the same example also revealed that use of the developmental process had created a local optimum. A primary goal of a neutral mapping is to alleviate this problem and the results therefore suggest that this goal is not achieved by the mapping in its current form. In order to gain a greater insight into the nature of the search space created by the developmental process, a more thorough analysis is required. In this section, genotype space is exhaustively enumerated so that definitive statements can be made about its structure. The lessons learned from this analysis are then used to re-engineer the developmental process to create a more amenable search space that is free from local optima and contains the best possible network design.

5.5.1 Phenotypic constraints

The encoded planning rule resulted in a genotype of 20 bits and thus generated 2^{20} = 1,048,756 distinct genotypes. This relatively manageable number allowed the phenotypes (network designs) that resulted from the application of each of these encoded rules to be ascertained and thus the precise mapping between genotype space and phenotype space to be determined. There were also 20 vacant sites in the initial environment, each of which could be in one of two states in the final network design; vacant or housing an access node. There were thus 2^{20} potential phenotypes resulting in phenotype space being exactly the same size as genotype space. In theory therefore the mapping between genotype and phenotype space could be one-to-one in which each genotype produced a unique phenotype. In reality, however, there was large-scale neutrality in this mapping. Only 52 phenotypes were produced by all possible instances of the encoded planning rule and the vast majority of phenotypes (99.995%) were not possible; the developmental process imposed enormously tight constraints on evolution.

On average approximately 20,000 genotypes mapped on to each of the 52 possible phenotypes. However, the exact figure was highly variable ranging from 16 to over 750,000. The proportion of genotype space representing each of the possible phenotypes is shown in Figure 5.10. It can be seen that genotype space was dominated by a single phenotype that was produced by over 70% of genotypes. This phenotype represented the network design in which no access nodes had been added to the network i.e. when the developmental process had no effect. It was of a low fitness and it was thus not desirable for it to occupy so much of genotype space. Its dominance reduced the amount of genotype space occupied by other, higher fitness phenotypes and hence had the potential of isolating these phenotypes in genotype space. The reason for the dominance of this "null" phenotype is discussed in later sections.

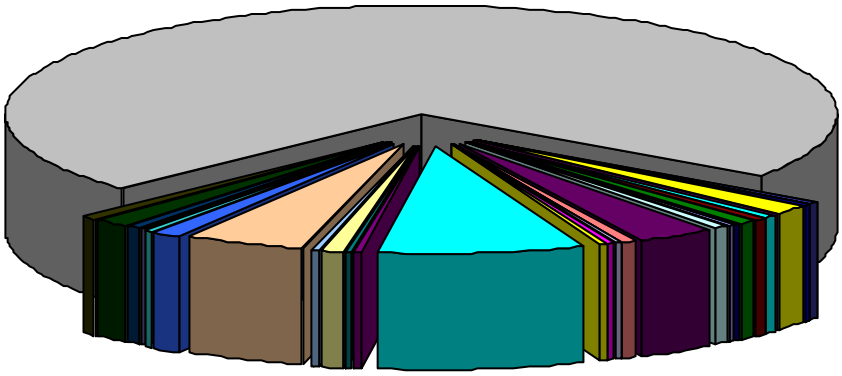


Figure 5.10: The proportion of genotype space covered by each of the 52 phenotypes generated by the encoded planning rule. Genotype space is dominated by a single phenotype that covers over 70% of the space.

5.5.2 Neutral networks in genotype space

The above analysis revealed that genotype space was sub-divided into 52 sets of genotypes each producing a specific phenotype i.e. there were 52 neutral sets. In order to allow a population to explore these neutral sets in search of areas of genotype space that allow further increases in fitness (when no immediate increase in fitness is possible), they must be connected by application of the genetic operator i.e. the neutral sets must form neutral *networks*. In order to ascertain whether this was the case, the connectedness of the genotypes within each neutral set was determined. However as for the analysis of the abstract mappings in the previous chapter, a measure of connectedness must be defined. The analysis of the abstract mappings was restricted to the simplest modification to the genotype; single-point mutation. This approach allowed a "base level" connectivity to be determined, which was only likely to be enhanced by considering higher mutation rates and more complex genetic operators. For the same reasons, the analysis performed in this section was also restricted to single-point mutations.

In order to determine which genotypes within a neutral set were connected into neutral networks by single-point mutations, the following process was used:

1. Move the first genotype in a neutral set, N_S, into a subset, S_{NS}.
2. For all remaining genotypes in N_S,
    a. If the genotype is a single mutation away from any member of S_{NS}, move the genotype from N_S to S_{NS}.
3. If new genotypes were added to S_{NS}, repeat from step 2.
4. If N_S is not empty, repeat from step 1 using a new subset.

If all the genotypes within a neutral set were connected into a single neutral network, this process would result in one subset containing all the genotypes in the neutral set. However if the neutral network was fragmented, a number of subsets would result. These subsets would form neutral sub-networks that represented the same phenotype but were disconnected from each other in genotype space. The results, shown in Figure 5.11, reveal that 23 of the neutral sets were connected into a single neutral network. However, the remaining 29 neutral sets were fragmented into a number of smaller sub-networks. The majority of these formed only 2 disconnected neutral networks, however, in some cases 3 and 4 were present. This fragmentation may be detrimental to the search process as it has the potential of reducing the ability of a population to explore genotype space through neutral drift; a population may be restricted to one of the sub-networks and hence isolated in genotype space. This would increase the likelihood of local optima.

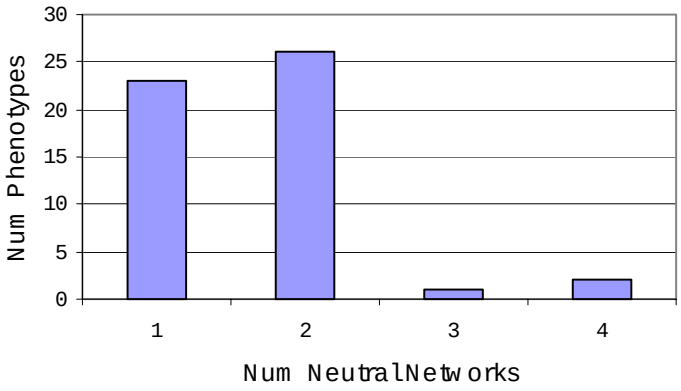


Figure 5.11: The neutral networks associated with each of the 52 phenotypes. 23 of the phenotypes have a fully connected neutral network associated with them. The remainder are represented by a number of neutral sub-networks that are disconnected from each other in genotype space.

5.5.3 Local optima

In order for genotype space to be free of local optima, all neutral networks with the exception of the global optimum must allow access to another neutral network of higher fitness. If this was not the case then the neutral network would effectively form a "local" optimum; the population may drift at the same fitness level but this neutral drift could not move the population to an area of genotype space that allowed continued improvements in fitness. In order to assess whether any such local optima were present in this case, the accessibility between each neutral network was assessed. The analysis again concentrated on single-point mutations and the following process used; all single-point mutants of each genotype on a neutral network were generated and the resulting phenotypes recorded. The fitness of each phenotype was then assessed and the number of unique higher-fitness phenotypes recorded. The results for each of the 86 neutral networks are shown in Figure 5.12.

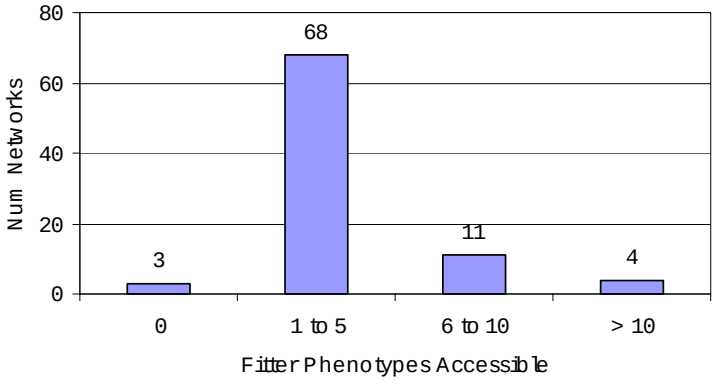


Figure 5.12: All neutral networks categorised according to the number of higher fitness phenotypes they allow access to. In 3 cases, higher-fitness transitions are not possible.

For a neutral network not to represent a local optimum, it must give access to at least one higher-fitness phenotype. However, the figure shows that 3 of the neutral networks did not allow access to higher fitness phenotypes via single-point mutations. One of these represented the global optimum but the remaining two represented local optima that could be detrimental to the search process. Indeed, one of these local optima was discovered in the example evolutionary run analysed earlier in this chapter. Its detrimental effect was very apparent in this case as evolutionary progress drastically slowed. Removal of these local optima would thus be highly desirable.

5.5.4 Removing local optima

Both the problematic neutral networks contained 3069 genotypes and were one sub-network of a pair. That is, two disconnected neutral networks were present in genotype space that represented the same phenotype. Furthermore, in both cases the other sub-network of the pair did allow access to higher fitness networks. Thus it was the fragmentation of the neutral networks that had resulted in local optima; if the neutral sets had formed a single connected neutral network they would not have arisen. The cause of the fragmentation was thus assessed in greater detail by determining the differences between the two sub-networks of a pair.

For each phenotype, it was crucial that certain sections of the genotype were at a precise value. These sections were thus constant for all genotypes within a given neutral network. However in the case of the local optima, the required state of certain constant sections was different for each sub-network. Thus, it was not possible to connect the two sub-networks through neutral mutations as this would require the entire constant section to be simultaneously changed from the state required for one sub-network to that for the other. For both local optima, the problem resulted from the encoded demand threshold. In one case, the required value for one sub-network was 7 and for the other 8. Although these values resulted in very similar demand thresholds and generated the same phenotype, in genotype space they were separated by a hamming distance of 4 i.e. four single-point mutations were required to change from one value to another. This effect was the root cause of the fragmentation as illustrated in Figure 5.13.

In order to address these difficulties it was necessary to modify the encoding to ensure that contiguous values of the parameters i.e. those that were a single quantised level from each other, were also close to each other in genotype space. This was achieved by adopting the familiar Gray encoding scheme [21]. In this scheme, contiguous values are guaranteed to be a hamming distance of 1 away from each other and hence the problem highlighted above would not arise. The above analysis was repeated for the Gray encoding scheme and revealed that the use of a Gray code had eliminated fragmentation of the neutral networks. Each of the 52 phenotypes was

represented by a single fully connected neutral network. All these networks allowed access to higher fitness phenotypes and thus the local optima generated by the binary encoding had been removed.

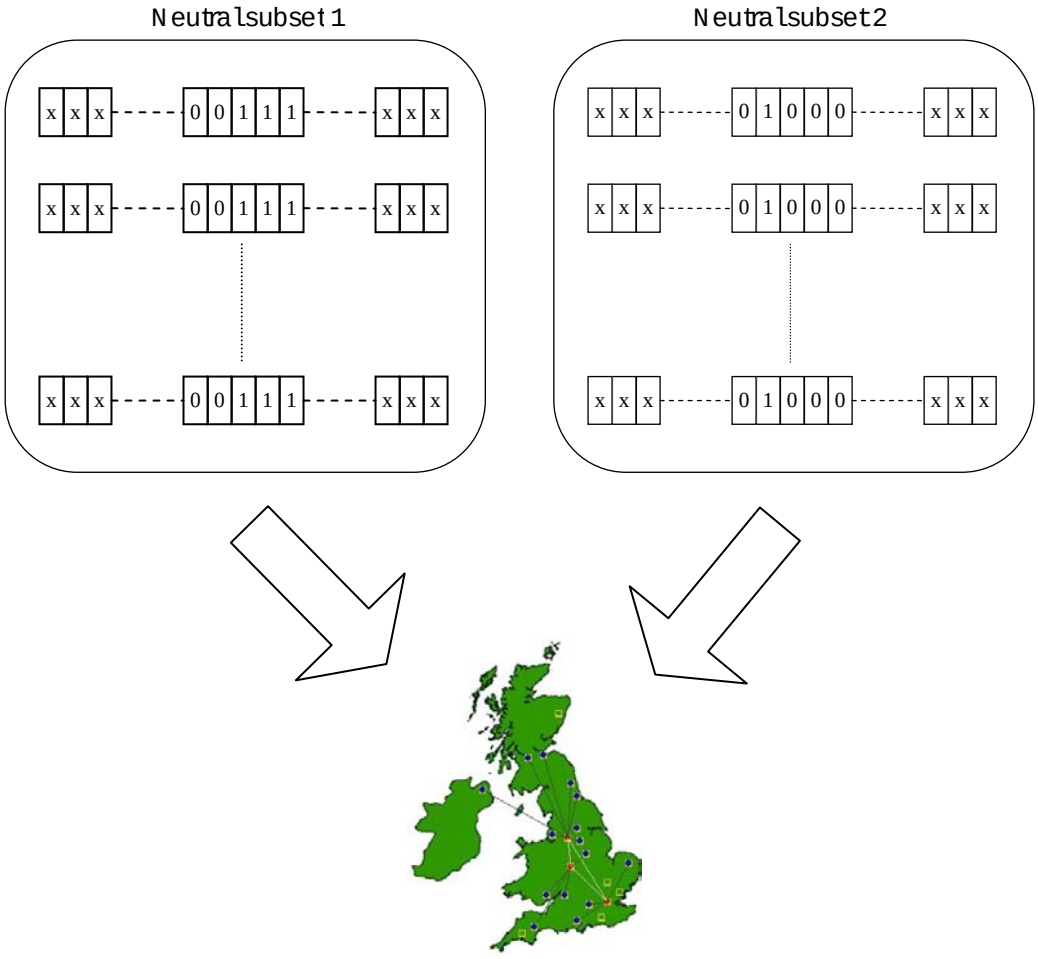


Figure 5.13: Illustration of neutral network fragmentation. The set of genotypes mapping onto a given phenotype is divided into two distinct neutral networks. Each sub-network requires one of the encoded values to be at an exact value. Although these values are very similar when decoded (7 in one case and 8 the other) and produce the same phenotype, they are very different in genotype space and are not neutral neighbours.

5.6 Re-engineering the developmental process

The developmental process explored above generated large-scale neutrality in the genotype-phenotype mapping; a total of 2^{20} genotypes mapped onto only 52 of the 2^{20} possible phenotypes and thus the vast majority of phenotypes were not possible. While generating such phenotypic biases was an aim of this approach, a danger is that the very best networks could never be produced. If the possible phenotypes are of high enough quality, the loss of the actual global optimum may be an acceptable price to pay for the prospect of removing local optima. However,

it would be desirable if the global optimum could be created by the developmental process. In order to assess whether this was the case, the fitness of each of the 2^{20} network designs was ascertained. This revealed that there were indeed network designs that were of a higher fitness than the best of the 52 possible phenotypes. However, only two such network designs existed and these were of only slightly higher fitness.

The reason that the global optimum could not be produced in this case can be understood by examining the relationship between the planning rule and the fitness calculation. The planning rule defines two conditions that take into account the demand at a site and its distance away from an existing node. The former condition relates very well to the fitness calculation. Access nodes need to be placed where there is high demand as this reduces the necessity for routing calls via the PSTN and increases the demand that can be satisfied. However, the latter condition is less well matched to the fitness calculation. For a high distance threshold, sites that are far from the existing network would be preferentially selected to house access nodes. This reduces the requirement to transmit data over the PSTN and hence reduces the associated cost. However, the same cost reduction could be achieved if a node was placed at an equivalent site that was close to the current network as the PSTN costs are not dependent on distance. Preferentially selecting sites based on distance does not therefore reduce these costs. However, the distance condition does impact another part of the fitness calculation as the cost of the links required to connect an access node to the core network is greater for larger distances. In some cases, therefore, it may be beneficial to favour sites that are relatively close to the existing network so as to reduce the cost of the links. However, the distance condition does not allow for this as when the distance threshold is low enough to allow these sites to satisfy the distance condition it also allows all the sites that are at a greater distance to do the same.

The encoded planning rules are effectively heuristics that help bias the search space in favour of quality networks. The distance condition is not a good heuristic for designing this network and is thus detrimental to the search process rather than an aid to it. This is evidenced by the example evolutionary run highlighted earlier in this chapter. Table 5.6 shows that in order to create high quality networks, the distance threshold was set at a low level so that the condition was always true and this part of the planning rule made redundant. In fact, the best of the 52 possible phenotypes could have been achieved by simplifying the planning rule and encoding only the demand threshold. However, demand alone is not sufficient for generating the best possible network design. For this the development process must be re-engineered.

5.6.1 Modifying the planning rule

When a site is vacant, the call-handling procedure routes its demand to the nearest node via the PSTN. Thus, when a node is added to a site it is possible that additional demand will be generated from adjacent vacant sites. It may be important, therefore, for the planning rules to take

into account not only the demand at a site but also the demand at nearby sites. This requirement is captured by the following planning rule:

IF ($demand_i > j$) **OR** ($demand\_in\_radius_i(k) > l$)
THEN Add access node

Equation 5.7

Where j and l are threshold parameters, $demand_i$ is the demand at site i and $demand\_in\_radius_i(k)$ is the demand within a radius k from site i. This updated rule does not consider the currently deployed network and thus the action of adding a new node does not affect the conditions of the rule. For this reason, an iterative developmental process is not required. The network design can be generated through a single application of the rule to each of the vacant sites.

5.6.2 The genetic encoding

The 3 parameters j, k and l were represented using 5-bit Gray encoded values which resulted in a 15-bit genotype. This encoding again allowed for 2^5=32 distinct values of each parameter, which were used to quantise the ranges given in Table 5.7.

Table 5.7: Ranges of the parameters for the revised planning rule.

Parameter	Description	Min	Max
j	Demand threshold	0	1500
k	Radius	0	50
l	Radius threshold	0	1500

5.6.3 Phenotypic constraints

The phenotypes produced by each of the 2^{15} = 32,768 genotypes were generated and a total of 137 unique phenotypes were discovered. This again imposed very tight constraints on evolution; 99.987% of the potential phenotypes could not be generated by the modified planning rule. On average, approximately 240 genotypes mapped on to each of the 137 possible phenotypes. However, this number was again highly variable ranging from 2 to nearly 22,000. The proportion of genotype space occupied by each of these neutral sets is shown in Figure 5.14. As for the initial planning rule, genotype space was dominated by a single phenotype that occupied 66% of the space. This phenotype was of relatively low fitness and corresponded to the network design in which a node was added to every vacant site. Its dominance restricted the extent of the other neutral networks and thus there was again the potential of isolating these neutral networks in genotype space and creating local optima.

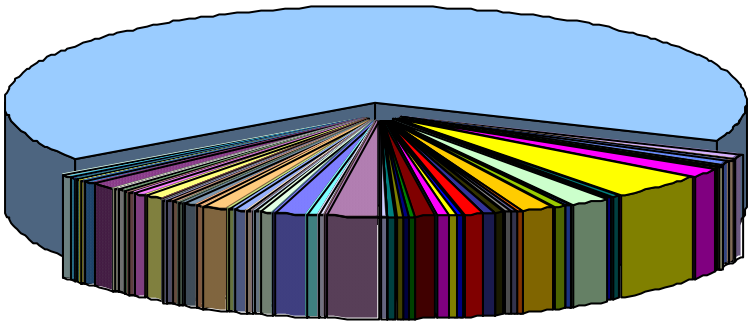


Figure 5.14: The proportion of genotype space occupied by the 137 phenotypes created by the new planning rule. Genotype space is again dominated by a single phenotype occupying 66% of the space.

Although the phenotypic constraints were again very strong for this planning rule, the set of 137 possible phenotypes included the best possible network design. Modification of the planning rule to embody more appropriate heuristics had thus created the equivalent of common phenotypes which included the global optimum.

5.6.4 Neutral networks

Although the global optimum was present in genotype space, it remained important for the neutral sets to be connected into networks that each allowed access to higher fitness phenotypes. In order to assess whether this was the case, the neutral sets were analysed using the process described in section 5.5.2. This analysis allowed the structure and any fragmentation of the neutral networks to be ascertained. The results, shown in Figure 5.15, reveal that a large majority of the phenotypes were represented by a single neutral network. However, even with the use of Gray coding some fragmentation was evident. The cause of this fragmentation will be discussed in a later section.

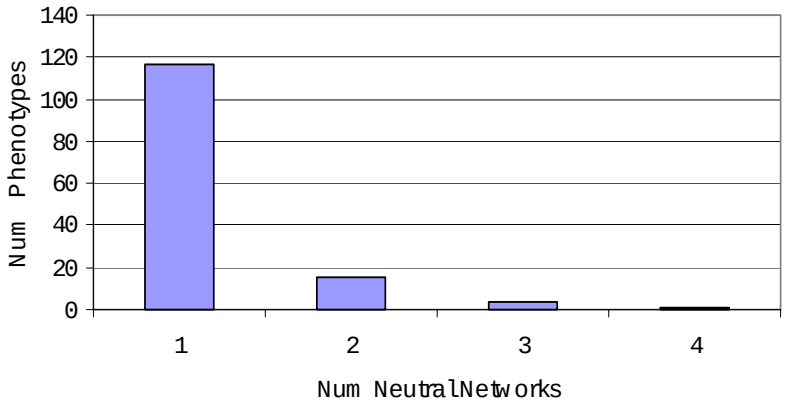


Figure 5.15: The 137 possible phenotypes categorised according to the number of neutral networks that represent them in genotype space. Limited fragmentation of the neutral networks is evident.

The fragmentation had split the 137 neutral sets into a total of 163 neutral networks. However, in this case it did not generate any local optima. A higher fitness phenotype was accessible from all neutral networks apart from the global optimum. The re-engineered developmental process thus resulted in a search space with highly desirable properties; it emphasised phenotypes of high quality that included the global optimum and did not contain any local optima.

5.6.5 The effect of parameter ranges

For both the original and modified planning rules, a single phenotype dominated genotype space and thus restricted the extent of the neutral networks associated with other phenotypes. Although this did not generate any local optima for the modified planning rule, it has the potential of doing so when the approach is applied to other problems as a population's movement in genotype space is restricted. These problems are likely to be exacerbated by the fragmentation of the neutral networks that was evident for the revised planning rule. Analysis revealed that both these difficulties were heavily influenced by a common cause; the ranges of the encoded parameters.

For the original planning rule, the demand threshold was allowed to vary between 0 and 1500; however, the maximum demand at any of the vacant sites was 600. Thus, all encoded values that were greater than 600 resulted in a phenotype in which no access nodes were added to the network as no site could possibly exceed the threshold and thus satisfy the rule condition. 20 of the 32 quantised levels of the demand threshold resulted in a value greater than 600 and thus 62.5% of the possible values resulted in this "null" phenotype. This strong bias was the overriding reason for the dominance of that phenotype and was caused by an inappropriate choice of parameter range. If the maximum value of the threshold mirrored the maximum demand at a site any biases in genotype space would have been greatly reduced.

A similar effect also resulted in the dominance of a single phenotype for the revised planning rule. However, in this case the dominant phenotype corresponded to the network design in which access nodes were added to every vacant site. The principal reason for this was not the demand threshold but the radius. This parameter controlled the size of the neighbourhood that was deemed relevant to a given site. If demand within this radius was greater than a threshold then an access node may be added to the site. When the radius was sufficiently large it included enough sites to allow the accumulated demand to exceed the majority of threshold values. This was the case even for relatively small radii and thus with a large maximum value of 50, a majority of parameter combinations resulted in every site satisfying the rule conditions and a fully populated network design. Again, the choice of the maximum value of a parameter was the overriding cause of the dominant phenotype.

The choice of parameter ranges also contributed to the fragmentation of the neutral networks. The parameters were encoded using 5 bits and thus the number of possible values was fixed at 2^5=32. If these 32 values quantised a relatively large range then the minimum change to the parameter was also relatively large. For example, the range of the demand threshold was 1500 and thus the minimum change to this parameter was 1500/32 = 46.875. Such large changes are less likely to be neutral with respect to the phenotype. As an example, consider the following parameter values for the revised planning rule.

Table 5.8: Example parameter ranges for two genotypes encoding the revised planning rule that map onto the same phenotype.

	a	*b*	*c*
Genotype 1	562	15	1218
Genotype 2	562	17	1312

The two genotypes shown in Table 5.8 map on to the same phenotype but are on two disconnected neutral sub-networks. Changes to both parameters *b* and *c* are required to move between these genotypes in genotype space. However, individual changes to both parameter *b* and *c* result in different phenotypes thus fragmenting the neutral network. Due to the wide range of the radius parameter *b*, a single mutation changes the value from approximately 15 to approximately 17. However, if the quantisation were finer allowing a mutation that resulted in a value of 16, the change would be neutral. This would then allow parameter *c* to be neutrally mutated from 1218 to 1312 followed by a second adaptive mutation to parameter *b* that would complete the transition from one genotype to the other. Thus, finer quantisation would increase the number of neutral neighbours and allow previously disconnected neutral sub-networks to merge into a single network with a greater extent in genotype space. While this finer quantisation could be achieved by increasing the number of bits that encode each parameter, it can also be achieved by reducing the ranges of the parameters. This would have the effect of reducing the minimum changes to a parameter and increase the likelihood of neutral neighbours.

5.6.6 Modifying the parameter ranges

In order to determine whether a more appropriate choice of parameter ranges could reduce fragmentation and alleviate the problem of overly dominant phenotypes, the ranges for the revised planning rule were reduced as shown in Table 5.9. These ranges more closely mapped on to the actual values that were likely to be present in the network.

Table 5.9: Revised parameter ranges for the new planning rule.

Parameter	Description	Min	Max
j	Demand threshold	100	600
k	Radius	0	15
l	Radius threshold	100	1500

It was discovered that the restricted parameter ranges had reduced the number of possible phenotypes to only 36. In effect, more precise knowledge about the problem was embedded into the planning rule heuristics which introduced more stringent constraints. This set of phenotypes again included the global optimum and only one of the associated neutral networks exhibited any fragmentation. The revised ranges had also removed the overly dominant phenotype. Figure 5.16 shows that genotype space is more evenly distributed among the phenotypes. The largest proportion of genotype space occupied by a single phenotype was only 21%. Thus, more careful consideration of parameter ranges had removed undesirable phenotypic dominance and greatly reduced the fragmentation of the neutral networks.

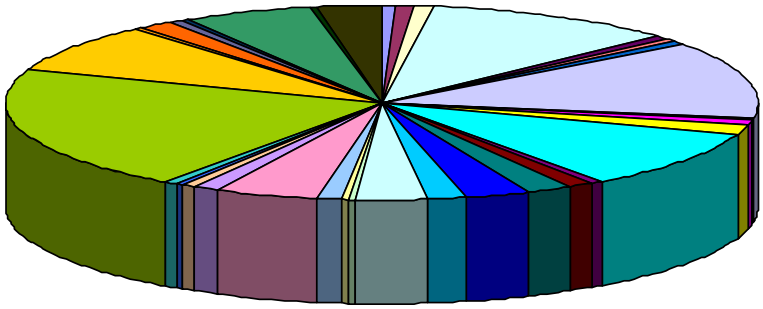


Figure 5.16: The proportion of genotype space occupied by the 36 phenotypes generated by the revised planning rule with reduced parameter ranges. Genotype space is no longer dominated by a single phenotype.

5.7 Discussion

This chapter has demonstrated that it is possible to bias the search space in favour of high-fitness phenotypes by embedding domain knowledge into a self-organising genotype-phenotype mapping. In addition, careful design of the mapping enabled each of these phenotypes to be represented by neutral networks that allowed access to higher fitness phenotypes. However in order to achieve this, the search space created by the developmental process required exhaustive enumeration such that the precise effect of each design decision could be ascertained. While this was possible on the small-scale problems investigated in this chapter, it is not possible for more realistic, larger-scale problems. Applying this approach to such problems may thus be problematic.

However, application of the approach does not rely on the ability to exhaustively enumerate the search space. Small-scale analyses allow lessons to be learned on manageable problems that may be equally applicable in other situations i.e. it allows the development of useful heuristics. There are two categories of heuristic that can be developed; generic and problem-specific. An example of the former was the effect of the binary encoding in the original planning rule. This encoding resulted in detrimental fragmentation of the neutral networks, which was removed through use of a Gray code. This would be a sensible option to encourage fully connected neutral networks for any problem in which continuous variables are represented in a binary genotype. This is thus an example of a generic heuristic that applies to a variety of problems regardless of their scale. Analyses of this nature could allow a number of such heuristics to be developed, which would be of significant advantage when developing neutral mappings for new problems.

The development of problem-specific heuristics can also be aided by these small-scale analyses. With an accurate network simulation, many of the lessons learned on small-scale design problems would be equally applicable for larger problems. For example, consider the design of the revised planning rule in this chapter. This rule was designed to map onto the specifics of the call-handling procedures of the network simulation. These procedures would remain the same for larger-scale problems and thus the same design principles would likely be equally fruitful. The structure of this planning rule is thus a problem-specific heuristic that can be used for related problems. This is exactly the approach taken in the following chapter in which larger-scale networks are designed using a similar network simulation and similar planning rules.

This chapter has focused on the effect of the developmental process on the structure of the search space. Although the ability to evolve network designs was also demonstrated, a more detailed analysis of the performance of an evolutionary algorithm using such a process is required to demonstrate its utility. In addition, the performance of the approach in comparison to a more traditional direct encoding must be ascertained. The small-scale problem considered in this chapter does not provide a suitable test-bed for such an analysis as the resulting search spaces are relatively trivial. However, the lessons learned in this chapter can be carried forward to larger-scale problems that would provide a more challenging test-bed and allow meaningful comparison to be made between the developmental process and a direct encoding. This is the subject of the following chapter.

5.8 Summary

In this chapter a new method of evolving telecommunications networks has been proposed, which genetically encodes the instructions for creating a network rather than the network itself. These instructions take the form of planning rules and their application in the context of the network's simulated environment produces the final network design. This process is loosely analogous to biological developmental and growth. Its analysis revealed the following key points:

- The developmental process biased the search space in favour of a small number of phenotypes and imposed enormously tight constraints on evolution.
- The process introduced neutrality into the search space that was shown to be potentially beneficial to evolutionary search; however it also generated local optima.
- The very best network design was not initially one of the possible phenotypes as poor heuristics were embedded into the planning rules.
- Careful design of the developmental process allowed a more amenable search space to be created that included the global optimum and did not contain any local optima.
- The lessons learned on small-scale problems such as those considered in this chapter allow heuristics to be developed that may be useful for related, larger-scale problems.

The following chapter furthers the development of this approach by applying it to more challenging network design tasks. The main theme of the chapter is to compare the performance of the developmental approach with a more traditional direct encoding.

Chapter 6

Growing networks versus direct encoding: a comparative study

6.1 Introduction

The exhaustive enumeration of genotype space performed in the previous chapter allowed precise details of the search space created by the developmental process to be ascertained. This enabled the design of both the developmental process and the associated genetic encoding to be modified to result in a search space with the desired properties; a bias towards a sub-set of phenotypes that included the global optimum and an absence of local optima. However, in order to demonstrate the utility of the approach, its effectiveness must be investigated on a more realistic scale of problem when complete information about the search space is not available. An aim of this chapter is to conduct such an investigation.

The main theme of the chapter is to compare the developmental approach with a typical direct encoding and to draw out the strengths and weaknesses of each. In order for this comparison to be meaningful, the scale and difficulty of the problem was significantly increased. However, it shared many of the characteristics with the problem explored in the previous chapter which allowed the knowledge gained to be put to use. The focus of the problem was again determining the best locations for network nodes from a set of potential sites. In this case, however many more sites were made available as potential locations for nodes. In addition, multi-level networks were introduced that required optimisation at several different levels of hierarchy. These modifications allowed the scalability of each approach to be assessed.

The problem considered is a “green field” network design in which no network infrastructure is currently deployed and thus design of the entire network was required. Details of this network are given in the following section together with the environment in which it must operate. This modified problem also required a modified developmental process consisting of several stages that produced each hierarchical layer of the network. Subsequent sections give details of this process and of the direct encoding used for comparison. A mutation-based evolutionary algorithm is used to evolve networks using both approaches and comparative results are presented and discussed.

6.2 Network simulation

The problem addressed in this chapter was the design of a data network from an initially clean slate. Such "green-field" problems may occur when a telecommunications operator extends its network to new countries or areas for example. Although there is no currently deployed hardware to consider, the environment places other constraints on the design such as a finite set of potential locations for network nodes. The environmental context used in this work is described in the following section followed by details of the network that is to operate in that environment.

6.2.1 The environment

The overriding aim of this chapter is to compare the developmental approach with a direct encoding. It is possible that a specific environmental configuration could favour one or other of these approaches and thus a range of environments were used that shared certain features but were otherwise randomly defined. Considering the average performance over a range of such generic environments reduces the risk of the results being a consequence of the particular characteristics of a specific environment. The environments consisted solely of a number of sites that had access to the PSTN and could potentially house a data network node. Each of these sites was randomly assigned a demand and a coordinate on a two-dimensional plane; the potential ranges of the parameters are given in Table 6.1.

Table 6.1: Parameters for the sites in the simulated environment.

Parameter	Minimum	Maximum
Demand	0	1000
x coordinate	0	100
y coordinate	0	100

A site's demand was chosen according to a uniformly random distribution over the specified range. However, in order to simulate geographical clustering of a population and hence demand, the locations of the sites were non-uniformly generated. A number of population centres were defined within the 100-by-100 grid and the majority of sites were clustered around these population centres according to a Gaussian distribution with a definable standard deviation. The remaining sites were uniformly randomly distributed in the grid independently from the location of the population centres. This process is summarised below for P_c population centres:

1. Divide the sites into (P_c+1) sets of equal size.
2. Assign a set of sites to each population centre P_c leaving one unassigned set.
3. Uniformly randomly generate x and y coordinates within the defined range for each of the sites in the unassigned set.

4. Uniformly randomly generate an x and y coordinate within the defined range for each of the P_c population centres.
5. For all sites assigned to each population centre
    a. Generate random numbers drawn from a Gaussian distribution for both x and y coordinates.
    b. Add these values to the x and y coordinates of the population centre to generate the sites coordinates.
6. Uniformly randomly generate a demand for each site with a defined maximum.

Experiments were performed with environments containing 125 sites, 250 sites and 500 sites. The number of population centres used within the environments depended on the number of sites. A single population centre was used for 125-site environments, 2 population centres for 250-site environments and 4 population centres for 500-site environments. An example result of this process for an environment consisting of 500 sites and 4 population centres is visualised in Figure 6.1.

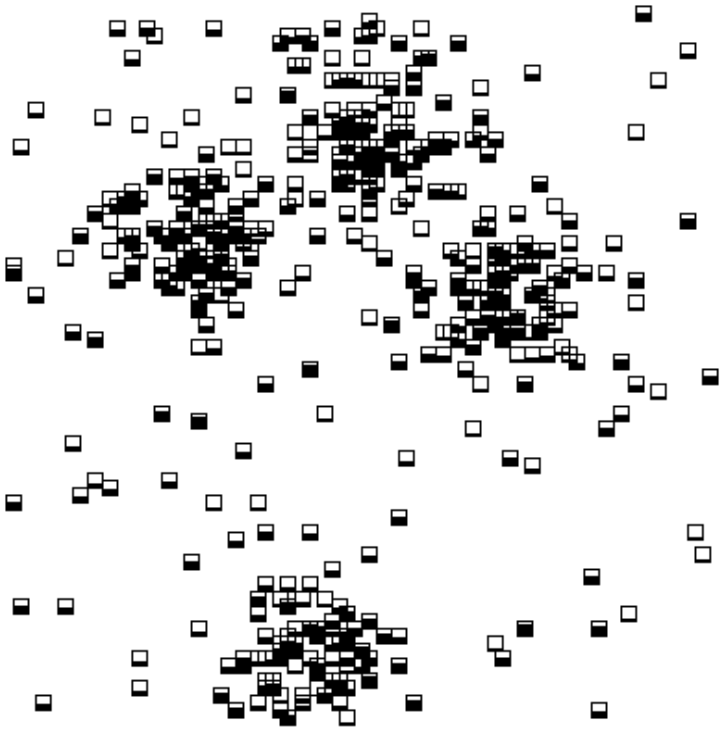


Figure 6.1: An example of a 500-site environment with 4 population centres. The majority of sites are clustered around one of the population centres. Demand at a site is illustrated by the fill of the box and is assigned randomly.

6.2.2 The network

In order to satisfy the demand in the above environments, hierarchical networks were employed as in the previous chapter. These networks again contained both core and access layers. The former consisted of a number of high capacity nodes that were fully meshed i.e. each node was connected to every other. The latter consisted of lower capacity nodes that were directly connected to their nearest core node. In addition to these two-level networks, more complex network design tasks were also defined. This was achieved by extended the network to include a third level of hierarchy consisting of so-called minor nodes. These nodes connected directly to their nearest access node. As the topology of the network was fixed by these rules, the design problem consisted solely of node placement i.e. determining the number of nodes together with their location. An example of a three-level network topology is shown in Figure 6.2 and the node capacities used for these experiments in Table 6.2.

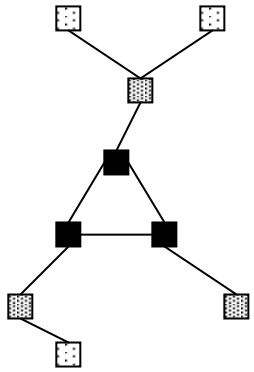


Figure 6.2: An example of the network topology created by the fixed rules used for these experiments. The network consists of three types of node, high-capacity core nodes (dark), lower-capacity access nodes (medium) and minor nodes (light).

Table 6.2: Parameters for the network nodes.

Node type	Capacity
Minor	125
Access	500
Core	2000

6.2.3 Call-handling procedure

Given the above environmental context and a candidate network design, the ability of the network to handle the demand at the various sites within the environment must be ascertained. For this reason, a call-handling procedure was defined. This procedure was almost identical to that used in the previous chapter and is repeated below:

1. All demand at a site without a collocated node is routed via the PSTN to the nearest node. This incurs a cost for the PSTN call.
2. Demand at a site with a collocated access or minor node is handled by that node if there is available capacity.
3. Any remaining demand at a site with a collocated access or minor node is routed to the nearest core node via the PSTN. This incurs a further cost for the PSTN call.
4. Any demand at sites with a collocated core node is handled by that node if there is available capacity.
6. Any remaining demand at a site with a collocated core node is dropped. This incurs a cost for each call that is dropped.

6.3 Developmental Process and Encoding

The hierarchical nature of the networks described above was mirrored in the developmental process through use of a multi-stage approach that progressively built the network layer by layer. Three stages were defined; the first was responsible for placing the core nodes, the second for placing the access nodes and the third the minor nodes. Similar planning rules controlled each of these stages however separate parameters were used in each case. The process is summarised below:

1. The following rule was applied to all sites:

$$\textbf{IF } (a_c < \text{demand}_i < b_c) \textbf{ OR } (c_c < \text{demand\_in\_radius}_i(d_c) < e_c) \quad \textit{Equation 6.1}$$

THEN Add core node

2. All core nodes were connected to one another.
3. The following rule was applied to all remaining vacant sites:

$$\textbf{IF } (a_a < \text{demand}_i < b_a) \textbf{ OR } (c_a < \text{demand\_in\_radius}_i(d_a) < e_a) \quad \textit{Equation 6.2}$$

THEN Add access node

4. All access nodes were connected to their nearest core node.
5. The following rule was applied to all remaining vacant sites:

$$\textbf{IF } (a_m < \text{demand}_i < b_m) \textbf{ OR } (c_m < \text{demand\_in\_radius}_i(d_m) < e_m) \quad \textit{Equation 6.3}$$

THEN Add minor node

6. All minor nodes were connected to their nearest access node.

Each of the planning rules considers both the demand at a site and the demand within a given radius of that site. This structure is very similar to the rule developed in the previous chapter. However, in this case two parameters were defined for each condition allowing a range to be defined rather just a threshold. These parameters were represented using 10-bit Gray-encoded numbers resulting in a possible 2^{10}=1024 values. These values were used to quantise the ranges shown in Table 6.3. The use of 10-bit values resulted in a finer quantisation than the encoding used in the previous chapter and was employed to discourage the neutral network fragmentation that was caused through overly coarse-grained quantisation.

Table 6.3: Parameter ranges for the three-stage developmental process.

Parameter	Min	Max
a_c, a_a, a_m, b_c, b_a, b_m	0	1000
c_c, e_c	0	10000
c_a, e_a, c_m, e_m	0	2500
d_c, d_a	0	20
d_m	0	10

The full developmental process required 15 parameters and thus a genotype of length 150 bits. However for two-level networks experiments only steps 1 to 4 of the developmental process were required, which resulted in 10 parameters and a genotype of length 100 bits.

6.4 DirectEncoding

The direct encoding specified a single gene for each of the sites in the environment, the state of which determined the state of the corresponding site. Each gene was assigned a number of alleles allowing the specification of all possible states of a site; for three-level networks 4 alleles were required whereas only 3 alleles were required for two-level networks. This approach generated no neutrality in the genotype-phenotype mapping, each genotype mapped on to a single unique phenotype as shown in Figure 6.3.

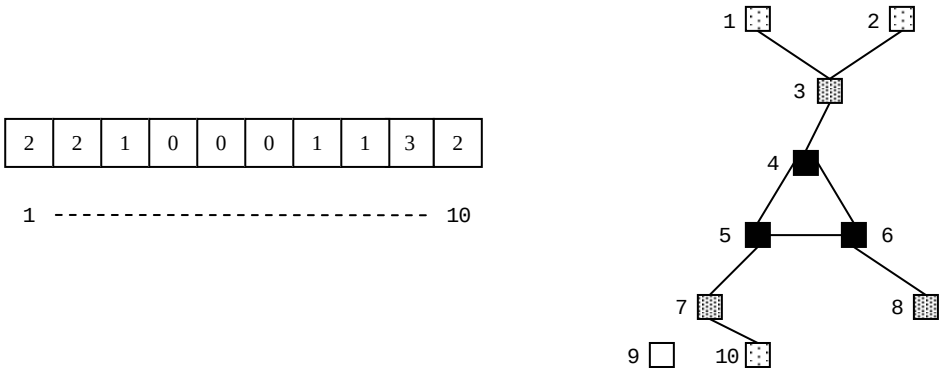


Figure 6.3: The direct encoding for a 3-layer network and 10-site environment. The genotype contains a single gene for each site that can adopt one of four values indicating whether the associated site will house a core node (0), access node (1), minor node (2) or be vacant (3).

6.5 Fitness function

The fitness of a network was a function of three component costs; hardware, PSTN calls and quality of service (QoS). The hardware cost was the sum of the cost of each network node together with the links required to connect the nodes into a network The PSTN cost resulted from the call-handling procedure being unable to satisfy demand at a given site and "routing" that demand to another node. The QoS cost resulted from demand that could not be satisfied by the network and was dropped by the call-handling procedure. These costs are quantified in Table 6.4.

Table 6.4: Costs for the extended network design problem.

Type	Symbol	Cost
Minor node	**Mn**	200
Minor Link	**Mlk**	12 per unit length
Access node	**An**	800
Access Link	**Alk**	50 per unit length
Core node	**Cn**	3200
Core link	**Clk**	200 per unit length
Dropped call	**QoS**	10
PSTN call	**PSTN**	1

The fitness function was a simple summation of these various costs and is shown in Equations 6.4 and 6.5 for two-level and three-level networks respectively.

$$f = (n_a * An + l_a * Alk) + (n_c * Cn + l_c * Clk) + (dc * QoS) + (p * PSTN)$$

Equation 6.4

$$f = (n_m * Mn + l_m * Mlk) + (n_a * An + l_a * Alk) + (n_c * Cn + l_c * Clk) + (dc * QoS) + (p * PSTN)$$

Equation 6.5

Where n_x is the total number of nodes of type x and l_x is the total length of the links required to link nodes of type x according to the defined topological rules. The subscripts m, a and c indicating minor, access and core nodes respectively, dc is the number of dropped calls and p the total number of PSTN calls.

6.6 Evolutionary Algorithm

A mutation-based evolutionary algorithm was chosen to allow comparison between the two approaches. Only a single individual was used in this algorithm and thus the effects of both population size and recombination were not considered. While larger population sizes and recombination are likely to have some impact on the results, their absence does not obviously favour either approach but allows the number of free variables required to control the algorithm to be reduced. The algorithm is summarised below:

1. Randomly generate a genotype and calculate its fitness.
2. Mutate the genotype and calculate the mutant's fitness.
3. If the mutant fitness is greater than or equal to the original fitness discard the original genotype and keep the mutant.
4. Repeat from step 2 until a given number of generations has been reached.

Various mutation rates were used for each encoding that were specified as a function of the length of the associated genotype L. A mutation event for the developmental encoding consisted of flipping a randomly chosen genotypic bit whereas for the direct encoding an alternative allele was randomly chosen for a randomly chosen gene. Four different mutation rates were considered in order to allow the most appropriate mutation rates to be ascertained for each approach. These rates defined the probability of mutating each gene and were set at *1/L*, *2/L*, *4/L* and *8/L*. The parameters of the evolutionary algorithm are summarised in Table 6.5.

Table 6.5: Parameters for the evolutionary algorithm used to compare the direct encoding and the developmental approach.

Parameter	Value(s)
Population Size	1
Generations	5000
Mutation Rates	1/L, 2/L, 4/L, 8/L

6.7 The Effect of Mutation Rate

The mutation rates that were used in the experiments gave a probability of mutating each gene. Thus when averaged over the whole genotype, the mutation rates 1/L, 2/L, 4/L and 8/L gave an expected number of mutations per genotype of 1, 2, 4 and 8 respectively. For the direct encoding it was guaranteed that each of these mutations would change the network configuration, which would almost certainly change the corresponding fitness value as the phenotype-fitness mapping was found to exhibit little or no neutrality. Thus, each mutation allowed new phenotypes and hence fitnesses to be explored. However, the neutrality inherent to the developmental encoding produced some buffering to the effect of mutation. A substantial number of mutations were neutral for the developmental encoding and thus did not change the network configuration. In effect, the neutrality reduced the exploration of new phenotypes for a given mutation rate. It would be expected, therefore, that the most appropriate mutation rate for the developmental encoding would be higher than that for the direct encoding to allow sufficient exploration to occur. The effect of mutation rate using the developmental encoding for 125-site environments with 2-levels of hierarchy is shown in Figure 6.4. These results are averaged over 10 independent runs.

These results support the intuition that higher mutation rates are more appropriate when using the developmental encoding. Progressive improvements are apparent as the mutation rate is increased; the cost of the final networks produced using an 8/L mutation rate are on average 0.8%, 1.2% and 3.1% lower than those produced using mutation rates of 4/L, 2/L and 1/L respectively. In contrast, Figure 6.5 shows that higher mutation rates perform poorly when using the direct encoding. Improvements in performance are apparent as the mutation rate is reduced from 8/L to 4/L and 2/L. However, if the mutation rate is reduced further to 1/L the performance decreases indicating that the exploration rate is not high enough. The cost of the networks produced using a mutation rate of 2/L are 2.4%, 3.5% and 14.4% lower than those produced using mutation rates of 1/L, 4/L and 8/L respectively.

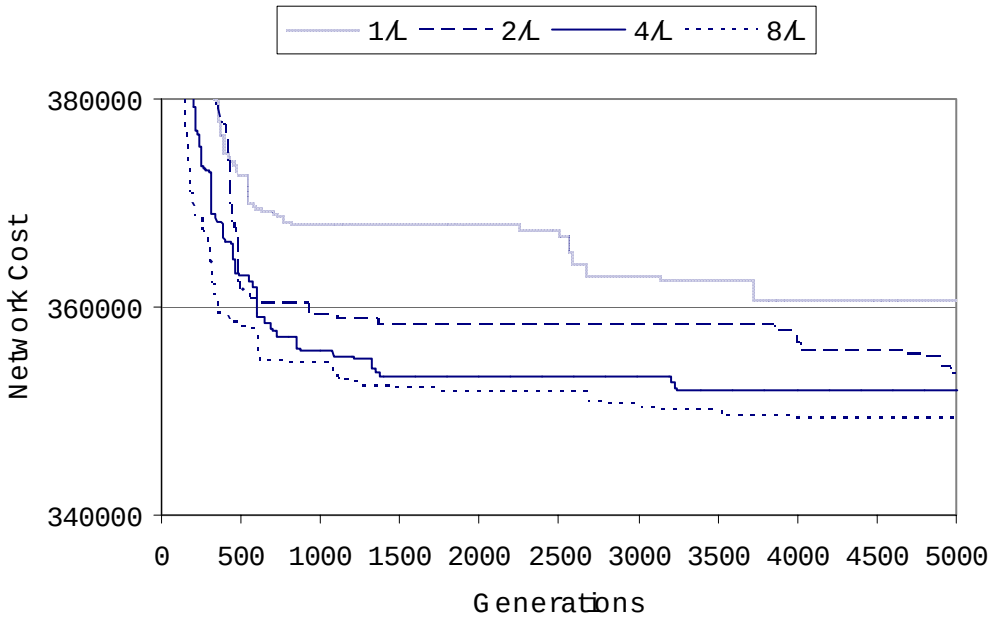


Figure 6.4: The effect of mutation rate using the developmental process for 125-site environments. The results are averaged over 10 independent runs.

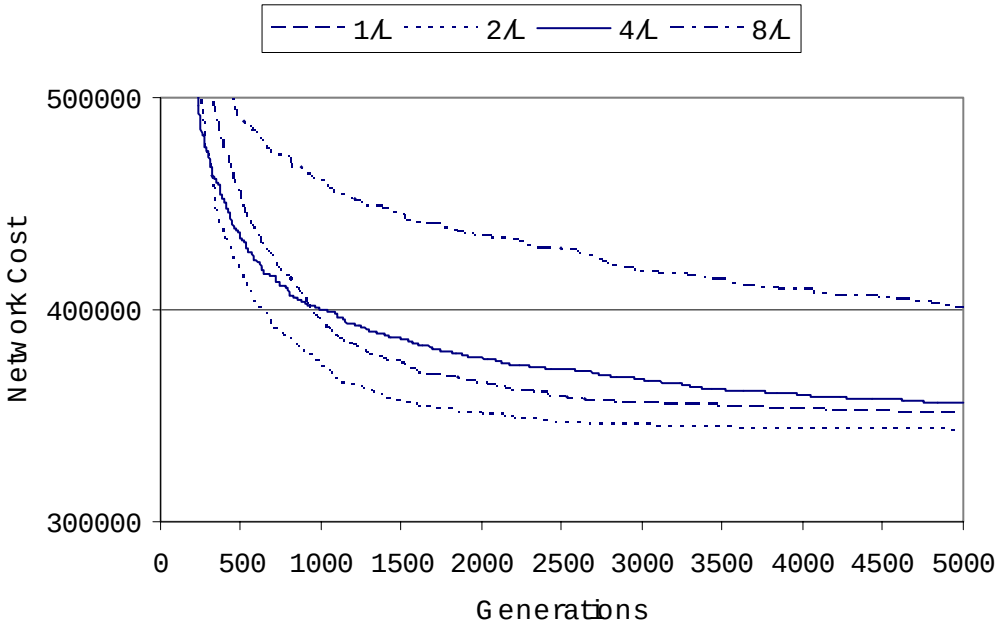


Figure 6.5: The effect of mutation rate using the direct encoding for a 125-site environment. The results are averaged over 10 independent runs.

Although the differences are relatively small in some cases and could potentially be the result of statistical fluctuations, this effect was consistent across the different environments that were explored in these experiments. As the number of sites was increased and an extra level of

hierarchy added to the network, 2/L consistently emerged as the most appropriate mutation rate for the direct encoding and 8/L for the developmental encoding. The results using these mutation rates alone are therefore presented in subsequent comparison of the two approaches.

6.8 Two-level Networks

The first comparative analysis was performed for networks consisting of two levels of hierarchy. Both the direct encoding and the developmental approach were used to evolve networks for a range of 125, 250 and 500 site environments and the results are presented below.

6.8.1 125-site environments

The network design task for 125-site environments and 2-level networks was the most simple of the problems explored in this chapter. These networks required only core and access nodes and hence minor nodes could be disregarded. Thus, only steps 1 to 4 of the developmental process highlighted in section 6.4 were used and the direct encoding required only three alleles. A series of 10 independent experiments were performed and the average performance of the two approaches is shown in Figure 6.6.

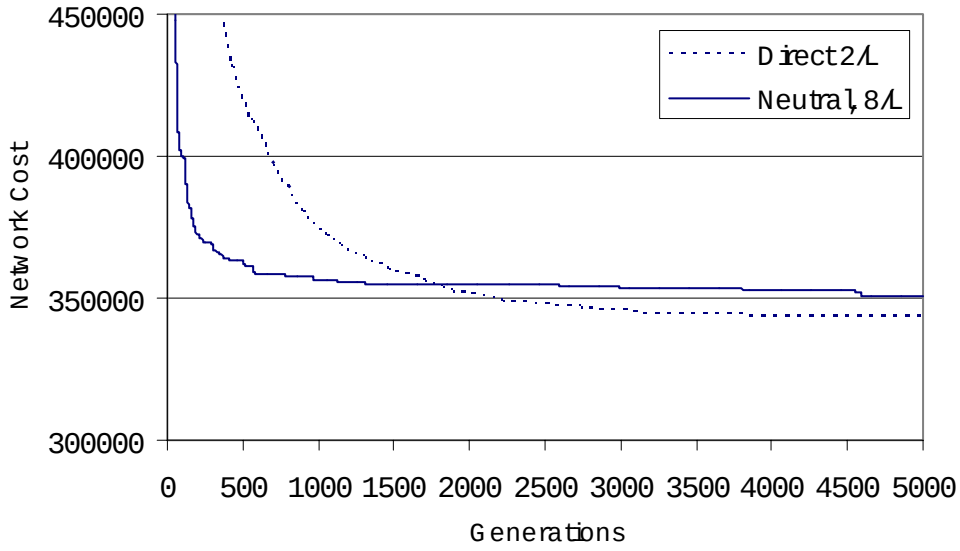


Figure 6.6: Comparison of direct and developmental encodings for 125 site environments and two-level networks. Results are averaged over 10 independent runs.

The figure shows that the developmental approach allowed the rapid discovery of relatively low-cost networks. As was seen in the previous chapter, this approach places tight constraints on the phenotypes that are available to evolution and emphasises a relatively small set of phenotypes.

The results suggest that the set of possible phenotypes included a number of relatively low-cost networks that were quickly discovered. These networks were then further refined during the evolutionary process. However, the results also suggest that this set of phenotypes may not include the best networks. The direct encoding initially produced higher cost networks but these were gradually improved upon until they were of a lower cost than those produced by the developmental approach. It is possible that further generations would have allowed the developmental approach to produce equally good networks but this does not seem likely given the small improvements that were evident in the later part of the process.

It is instructive to determine how the network design was modified during the evolutionary process. Figure 6.7 shows the number of nodes making up the network throughout the 5000 generations using the direct encoding. The initial state of the genotype was generated uniformly randomly and thus the number of access nodes, core nodes and vacant sites would be expected to be roughly equal at the beginning of the run. This is supported by the figure showing that the number of both access and core nodes in the initial network design was just over 40 i.e. around one third of the available sites. The number of core nodes was then quickly reduced and held relatively constant throughout the process. Modification of the location of a core node would likely have a significant impact on the network cost as it not only affects the cost of the core network but also impacts the access network. All access nodes are connected to the nearest core node and thus changing the location of a core node changes the cost of the links for all associated access nodes. Such large changes are likely to be detrimental as the quality of the networks increases and thus the design of the core network essentially becomes fixed at an early stage in the evolutionary process.

Changes to the network design and hence cost, are subsequently dominated by the access nodes which are gradually added to the network. As the number of connections to a core node is not considered in the fitness calculation, there is no penalty for connecting a large number of nodes to a given core node. Evolution is thus able to fine-tune the access network as decisions about node placement can be made relatively independently from other access nodes. The major factor in that decision is whether the additional hardware costs are warranted by a reduction in PSTN and dropped call costs. This is influenced by whether neighbouring sites contain access nodes as a site without a co-located node routes its demand to the nearest node and thus the presence of an access node may change the number of calls that are routed between neighbouring sites. However, these interactions are much weaker than those for core nodes and thus the access network can be more flexibly modified.

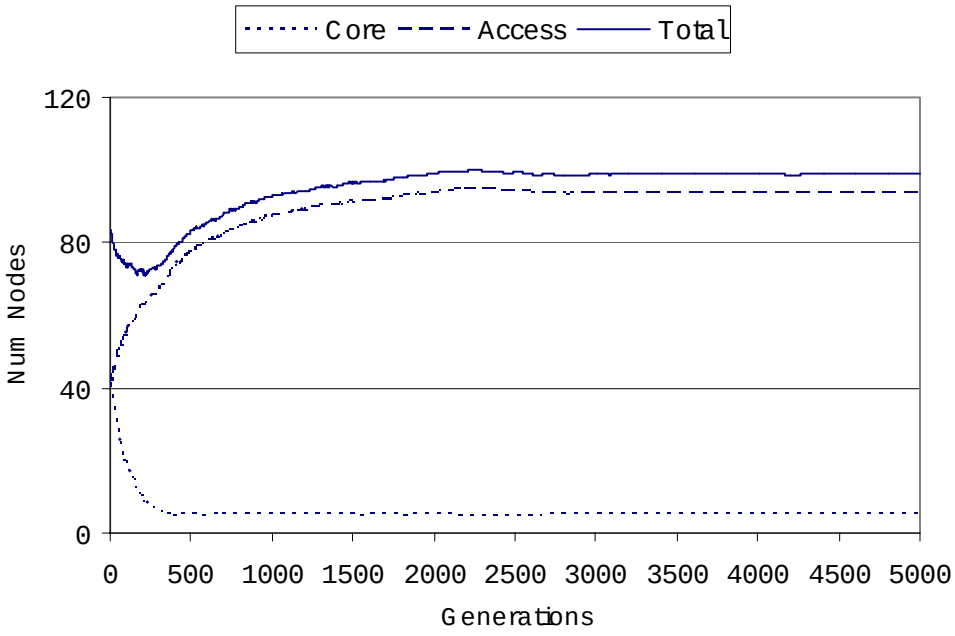


Figure 6.7: The number of core and access nodes making up a two-level network design for the direct encoding. Results are average over 10 independent runs.

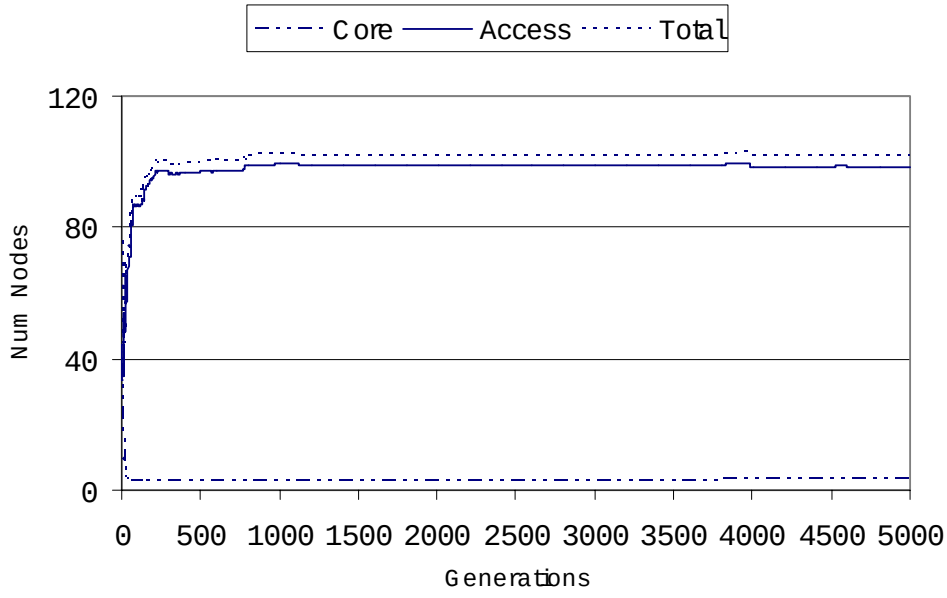


Figure 6.8: The number of core and access nodes making up a two-level network design for the developmental encoding. Results are average over 10 independent runs.

Figure 6.8 reveals a similar pattern for the developmental encoding, the core network quickly became established and the access network was subsequently modified. However, in this case this process occurs in far fewer generations. This accounts for the rapid discovery of high-fitness

networks when using the developmental process. However, the figure also reveals that the same degree of access network fine-tuning did not occur. The access network becomes largely established early in the process and the number of access nodes is relatively consistent after around 700 generations. In contrast to the direct encoding, evolution did not have the flexibility to control the addition of individual access nodes. It is this reduced flexibility that accounted for the fact that the direct encoding eventually produced better quality networks. Thus in this case, the constraints imposed by the developmental process are both advantageous in that they allow relatively high quality networks to be quickly discovered but ultimately detrimental in that they limit fine-tuning of the network.

6.8.2 250-site environments

In order to test the scalability of each approach, experiments were carried out on larger scale environments. That is, environments containing a greater number of sites at which nodes could be housed. Figure 6.9 shows the average performance of each approach for 10 independent experiments carried out on environments in which the number of sites was doubled to 250.

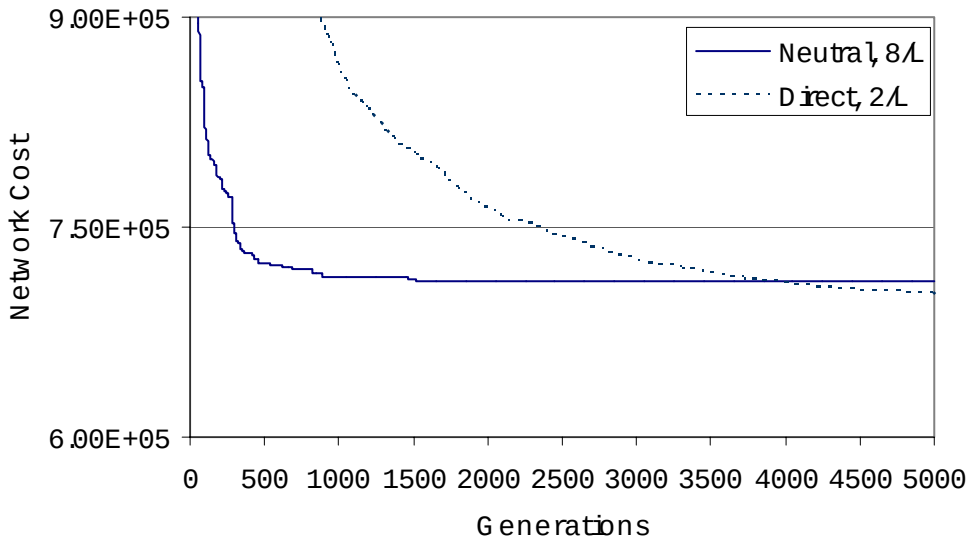


Figure 6.9: Comparison of direct and developmental encodings for 250-site environments and two-layer networks. Results are averaged over 10 independent runs.

The pattern of these results is similar to that for the 125-site environments. Relatively low-cost networks are quickly discovered using the developmental encoding but the direct encoding eventually produces lower cost networks. However, in this case the direct encoding takes many more generations to outperform the developmental approach. Lower cost networks are not discovered until around 4000 generations.

This evidence suggests that the developmental process may scale better than the direct encoding. When using the direct encoding, increasing the number of sites also increases the size of the search space as more genes are required to specify the state of the additional sites. Thus moving from 125 to 250 sites, increases the size of the search space from 3^{125} to 3^{250} genotypes. It is this effect that causes a corresponding decrease in the speed in which good solutions are found. In contrast, increasing the number of sites has no effect on the size of the search space generated by the developmental encoding. The same rules are applied to the additional sites as for the original sites and the heuristics built into the planning rules bias the search space in a similar way.

6.8.3 500-site environments

Figure 6.10 shows the results for environments in which the number of sites was again doubled to 500. It can be seen that progress when using the direct encoding was further slowed using these larger environments. In this case, the best solutions found using the direct encoding after 5000 generations were equivalent to those discovered after only several hundred generations using the developmental encoding. However, extrapolation of the direct encoding graph suggests that better solutions would eventually be discovered in this case also.

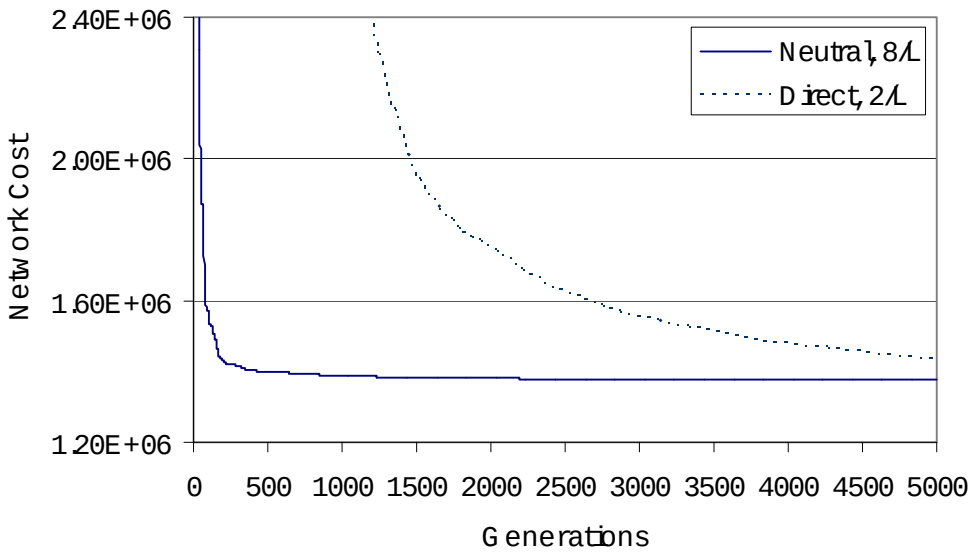


Figure 6.10: Comparison of direct and developmental encodings for 500-site environments and two-layer networks. Results are averaged over 10 independent runs.

As already discussed, increasing the number of sites in the environment increases the size of the direct encoding search space. However, it does not fundamentally alter its nature. The ruggedness and corresponding difficulty of a search space is heavily influenced by the epistatic interactions

between genes. If the contribution each gene makes to fitness is independent of the state of any other genes, then there are no epistatic interactions and the search space is entirely smooth containing no rugged peaks that lead to local optima. A direct encoding would be expected to perform very well on such a landscape, given enough time. However, in most real problems this is not the case and the fitness contribution of a gene is typically dependent on the state of a number of other genes. In this case there are relatively strong epistatic interactions as the fitness contribution of a node is always dependent on the location of other nodes in the network. Increasing the scale of the environment does not affect the fundamental nature of these interactions and hence does not fundamentally change the nature of the associated search space. It would be expected, therefore, that increasing the number of sites would slow down progress when using the direct encoding as a larger space must be navigated but would not change the fact that better networks would be discovered given enough generations on average. Nonetheless, the results for these larger scale environments begin to suggest that the advantages of the constraints imposed by the developmental process (quick discovery of relatively good networks) may begin to outweigh the disadvantages (reduced flexibility) as larger problems are considered.

6.9 Three-level Networks

In the previous section, it was suggested that the constraints imposed on evolution through use of a developmental process may be more advantageous as the size of the problem was increased. In this section, the difficulty of the problem was also increased by adding minor nodes and hence a new level of hierarchy to the network. These experiments required all 6 stages of the developmental process to be used and the number of alleles required for the direct encoding to be increased to 4 to allow specification of the 3 types of node together with a vacant site. Experiments were again performed using a series of 125, 250 and 500 site environments.

6.9.1 125-site environments

A series of 10 independent experiments were performed for both the direct encoding and the developmental approach using 3-level networks and 125-site environments. The results, shown in Figure 6.11, reveal that the developmental encoding again allowed the rapid discovery of relatively low-cost networks. However, in this case the direct encoding no longer allowed the discovery of lower cost networks as the evolutionary process unfolded. In addition, the direct encoding graph appears to have reached an asymptote and there is thus no indication that lower cost networks would eventually be discovered given more generations.

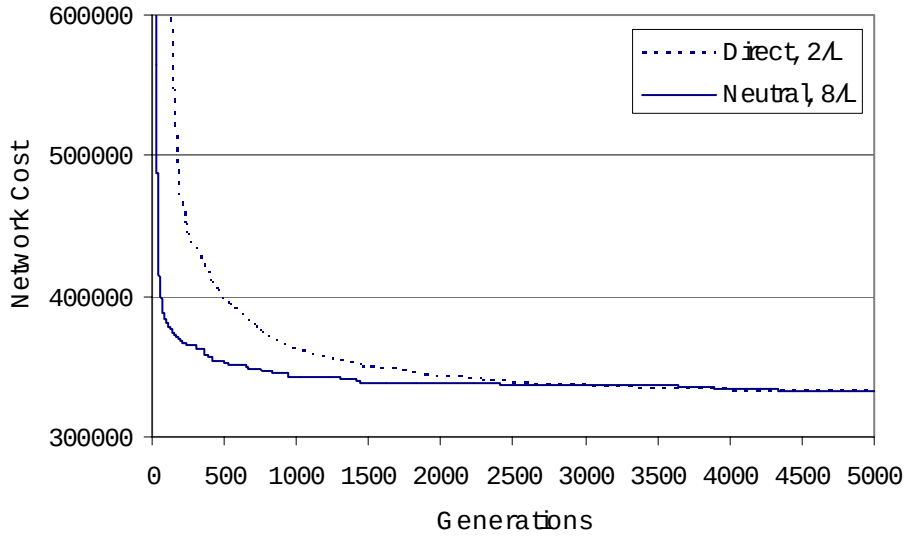


Figure 6.11: Comparison of direct and developmental encodings for 125-site environments and three-layer networks. Results are averaged over 10 independent runs.

In the absence of minor nodes, the access layer could be relatively flexibly modified once the basic structure of the core network was established. However, the inclusion of minor nodes reduced this flexibility as changing the location of an access node could now also impact the minor node layer. The effect of adding a new level of hierarchy to the network was thus to increase the strength of the epistatic interactions. Such effects created a more challenging search space for the direct encoding, which is reflected in the results.

It is again instructive to determine the make-up of the network during the evolutionary process. Figure 6.12 shows the average number of core, access and minor nodes when using the direct encoding. The pattern of core and access node usage was very similar to that for the two-level networks. The number of core nodes was quickly reduced and then held relatively constant, access nodes were then added to the network. The flexibility with which the access nodes were added may be surprising given the presence of the minor nodes. It may have been expected for the access network to become relatively fixed earlier in the process in a similar way to the core network due to the presence of minor nodes. However, the number of minor nodes was low throughout the process. The majority of high demand sites were serviced by either a core or an access node and thus the number of sites with a high enough demand to warrant the hardware costs of a minor node was relatively small. This in turn meant that the average number of minor nodes connected to each access node was also low. In the later stages this average was only approximately 0.4, which reduced the impact of an access node change on other network nodes and hence reduced the strength of the epistatic interactions. The effect on the search space remained strong enough however to be detrimental to the direct encodings comparative performance.

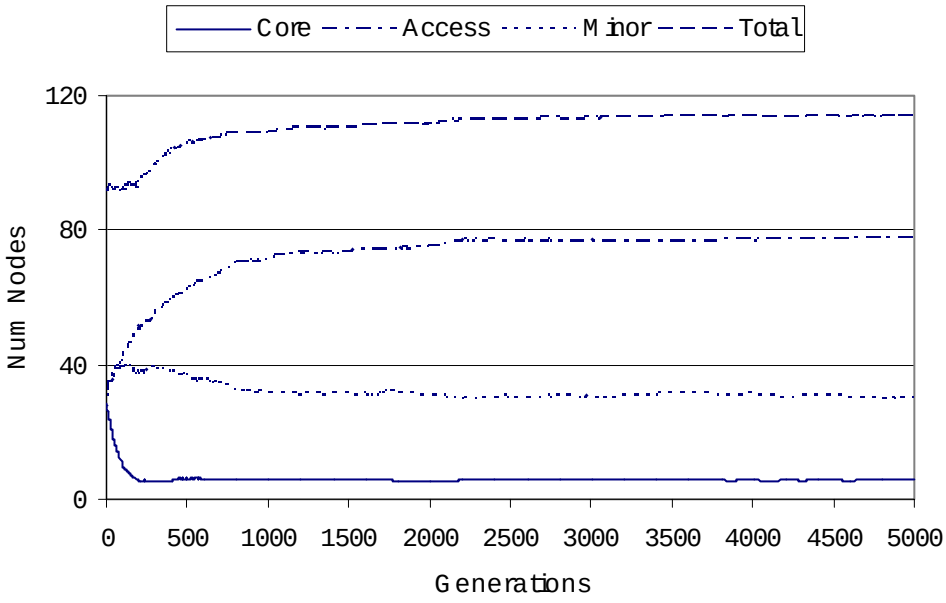


Figure 6.12: The number of core, access and minor nodes making up a three-level network design for the direct encoding. Results are averaged over 10 independent runs.

The developmental process achieved a similar balance between the three types of nodes but in far fewer generations due to the heuristics embedded in the planning rules. These results are shown in Figure 6.13.

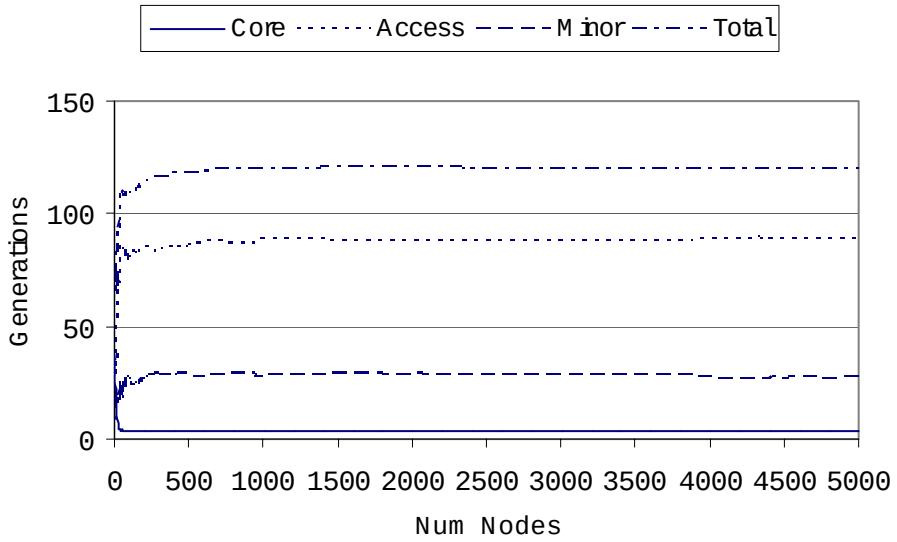


Figure 6.13: The number of core, access and minor nodes making up a three-level network design for the developmental encoding. Results are average over 10 independent runs.

6.9.2 250-site environments

The scalability of both approaches was again tested by increasing the number of sites in the environment to 250. The results averaged over 10 independent runs are shown in Figure 6.14.

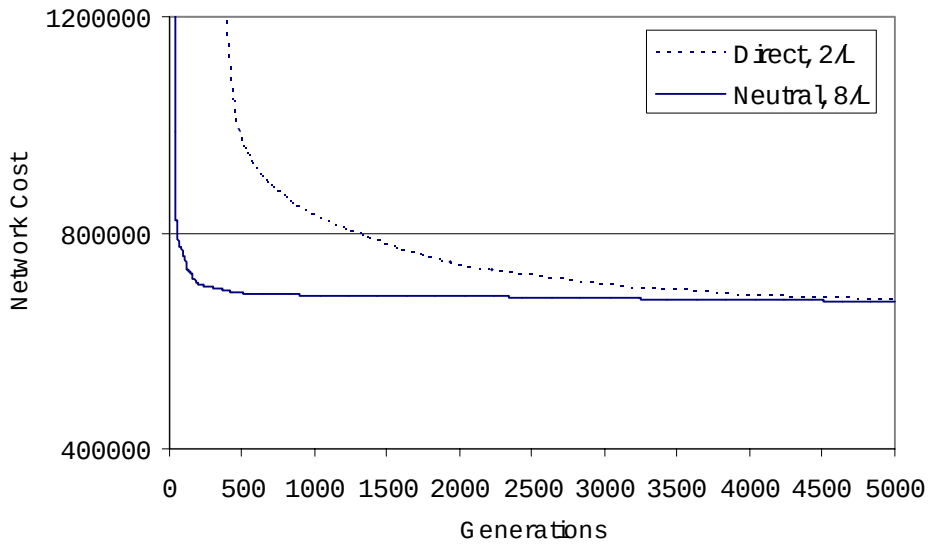


Figure 6.14: Comparison of direct and developmental encodings for 250-site environments and three-layer networks. Results are averaged over 10 independent runs.

The figure reveals a similar pattern as for the 125-site environments. Increasing the number of sites to 250 increased the size of the direct encoding search space from 4^{125} to 4^{250}, which again slowed evolutionary progress. After 5000 generations the quality of the networks produced using the direct encoding was equivalent to those produced by the developmental process. However, the latter discovered such networks in a very much smaller number of generations.

6.9.3 500-site environments

Increasing the number of sites to 500 further decreased the speed at which good solutions were found using the direct encoding, which was not able to match the performance of the developmental encoding over 5000 generations as shown in Figure 6.15. However, extrapolation again suggests that similar quality networks would eventually be discovered. These results support those from the two-level network experiments in suggesting that the direct encoding does not scale to larger scale problems as well as the developmental approach.

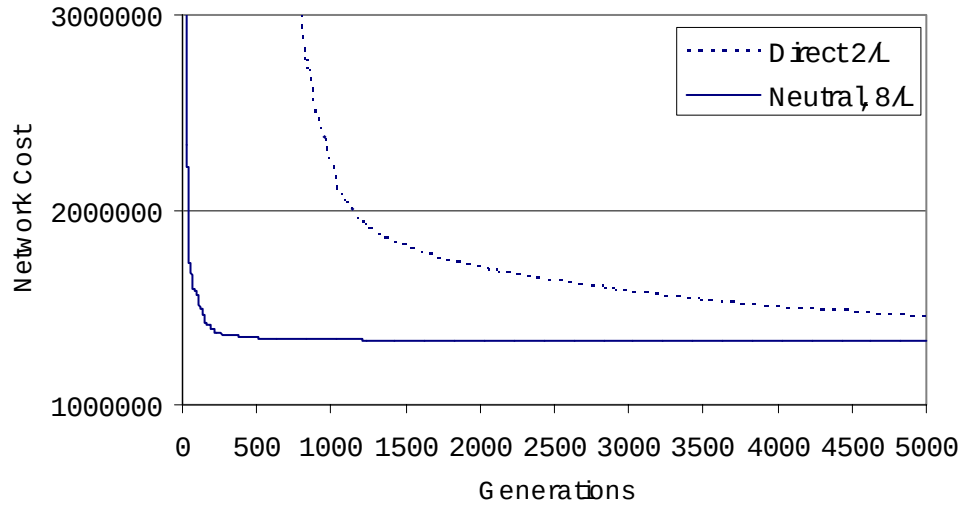


Figure 6.15: Comparison of direct and developmental encodings for 500-site environments and three-layer networks. Results are averaged over 10 independent runs.

6.10 Discussion

The results presented above do not and could not offer conclusive proof of the superiority of either the direct encoding or the developmental approach. Each has their own strengths and weaknesses and is appropriate in different situations. If a relatively smooth landscape can be created through use of a direct encoding then it would be sensible to adopt such an approach. The emphasis of certain phenotypes produced by the developmental approach may allow relatively good phenotypes to be quickly discovered but the rewards are not likely to justify the risk of disallowing the best phenotypes given that the direct encoding is likely to produce very good solutions given enough time. However, as both the scale and complexity of the problem increases then the direct encoding may no longer be the most prudent option.

The evidence presented above suggested that the relative performance of the direct encoding could not be maintained as the scale of the problem was increased. As larger problems were tackled, progress using the direct encoding slowed. For very large problems, the size of the search space may become too large to be successfully negotiated in any reasonable time scale. As the fundamental complexity of the problem and hence search space increases, the difficulties are exacerbated and the use of some form of heuristic is likely to become a necessity. The approach explored in this work is a very promising method of using such heuristics. Embedding planning rules within the developmental process allowed good phenotypes to be emphasised in the search space and made them likely outcomes of evolution rather than isolated solutions in vast search spaces.

The utility of the developmental approach is dependent on the design of the planning rules. It was shown in the previous chapter how inappropriate choice of both the planning rules and the ranges of the associated parameters had a detrimental effect on the nature of the search space. However, very good solutions were still easily accessible and were of only slightly lower fitness than the very best solutions possible. It is possible that the planning rules used in this chapter may not have allowed the generation of the global optimum. However, they allowed phenotypes to be very quickly produced that were comparable to or better than those produced by the direct encoding after a far larger number of generations. In realistic problems, specialised knowledge would be available in the form of existing and proven planning rules that would likely further enhance the developmental approach.

Addition of a third-level of hierarchy to the network increased the difficulty of the problem by increasing the strength of the epistatic interactions when using the direct encoding; the location of access nodes could no longer be modified without the likelihood of affecting the fitness of the minor node layer. The fact that the number of minor nodes was typically very low reduced this effect but did not remove it. The relative performance of the direct encoding for 3-level networks was reduced such that it could no longer produce designs of a higher quality than those produced after only several hundred generations by the developmental approach. For more realistic problems, many more issues would need to be considered in a network design. For example, the performance of the network as it routes data packets, the physical space available at various sites, the proximity to other networks etc. These factors would increase the difficulty of the problem and would likely cause additional difficulties for the standard direct encoding. However, the developmental approach would allow further heuristics to be captured in the planning rules with a view to fundamentally biasing the search space in favour of phenotypes that effectively deal with the various issues.

In order to achieve this, a more suitable balance may need to be struck between the constraints imposed on evolution by the developmental process and the flexibility that evolution has to create new solutions. In this chapter, the structure of the planning rules was precisely specified such that only the parameters of the rules were under evolutionary control. This imposed very tight constraints on the phenotypes that were available to evolution. In the previous chapter it was shown that over 99.9% of feasible phenotypes could not be generated by the developmental process. It is this effect that allowed the rapid discovery of very good network designs. However, such tight constraints place a lot of pressure on the design of the planning rules. If these rules do not allow the production of the required phenotypes then evolution can never generate them. This lack of flexibility allowed the direct encoding to eventually produce better solutions for two-level networks. It may be desirable therefore to loosen the constraints imposed by the developmental process. This could be achieved by allowing the structure of certain rules to be evolved along with the associated parameters. A combination of fixed rules that capture the domain expertise of

the network designers and evolved rule structures may provide a better balance between phenotypic constraint and flexibility. While this remains a topic for future work, the evidence to date suggests that the developmental approach compares very favourably to a direct encoding. It expedites evolutionary progress and scales well to larger and more complex problems.

6.11 Summary

This chapter has compared the developmental approach with a typical direct encoding on network design problems of varying scale and difficulty. The key findings are highlighted below:

- The developmental approach consistently allowed the discovery of high quality networks very much more quickly than the direct encoding.
- For the simplest problems explored in this chapter involving two-level hierarchical networks and small-scale environments, the direct encoding eventually discovered higher quality networks indicating that the very best designs may not have been possible using the developmental approach.
- As the scale of the problem was increased evolutionary progress using the direct encoding significantly slowed whereas the developmental approach again allowed very rapid discovery of high quality networks.
- As the difficulty of the problem was increased to consider three-level hierarchical networks, the relative performance of the direct encoding was significantly impaired. The direct encoding no longer allowed better quality networks to be discovered.
- The developmental approach was better able to scale to larger and more complex problems. The benefits of this approach may thus outweigh the risks of loosing the very best phenotypes.

The following chapter summarises the results from all the previous chapters and presents the overall conclusions together with suggestions for future work.

Chapter 7

Conclusions

This thesis has investigated the use of self-organising genotype-phenotype mappings in artificial evolutionary systems. Several abstract genotype-phenotype mappings were developed and the impact of the inherent neutrality was analysed in depth. A novel approach to the evolutionary design of telecommunications was also developed that employed both evolution and self-organisation. This approach was applied to a simplified version of a real network design problem. The resulting search space was exhaustively enumerated and the impact of the design choices ascertained. The approach was then applied to a more realistic scale of problem and comparison made to a typical encoding in which the network design was directly encoded in the genotype. In the following section, the results of this work are summarised and discussed with a view to evaluating whether the objectives highlighted in the Introduction were achieved.

7.1 Summary and discussion

The primary objective of this work was to determine whether the efficacy of an evolutionary algorithm could be enhanced through the coupling of self-organisation and evolution. This broad objective was composed of two more specific sub-objectives. The first of which was to determine whether self-organising genotype-phenotype mappings could be developed that resulted in search spaces that shared the desirable properties found in natural search spaces. The second sub-objective was to determine whether the use of such mappings in an evolutionary algorithm could provide any advantage over a typical direct encoding.

The studies of RNA molecular self-organisation, highlighted in chapter 2, were an important influence for this work. These studies revealed four main properties of the resulting search spaces that provide a useful guide for interpreting the results presented in this thesis. The four properties were large-scale neutrality, neutral networks, common structures and shape-space covering. The first two properties indicate that many genotypes mapped onto the same phenotype and were largely connected by single-point mutations. The latter two properties indicate that only a relatively small sub-set of potential structures or phenotypes were commonly discovered and the neutral networks associated with these common structures covered the search space i.e. they were within a relatively small radius from any arbitrary location in the space.

These spaces were shown to be very amenable to evolutionary search. The expansive neutral networks associated with each of the common structures percolated throughout the search space and drift on these neutral networks allowed the constant discovery of new phenotypes. This

reduced the possibility of an evolving population becoming isolated in sub-optimal regions of the landscape. In addition, the property of shape-space covering increased the probability of neutral drift allowing the rapid discovery of higher fitness phenotypes. Assessing whether similarly amenable artificial evolutionary search spaces were created in this work forms the basis of this discussion.

7.1.1 Abstract genotype-phenotype mappings

In chapter 4, self-organising genotype-phenotype mappings based on both a cellular automaton and a random Boolean network were introduced. The philosophy behind these mappings was to encourage the desirable properties described above into artificial evolutionary search spaces through use of computationally efficient models of natural processes. Both the mappings succeeded in introducing large-scale neutrality into the search space, a very large set of genotypes produced each of the possible phenotypes. In addition, these sets were connected into expansive neutral networks through single-point mutations. This was demonstrated by performing random walks on the neutral networks in order to simulate neutral drift. Extensive movement through genotype space was possible on such walks indicating that the associated neutral networks percolated throughout much of the space.

In order to be beneficial to evolutionary search, neutral drift must allow access to phenotypes that would otherwise not have been possible. The statistics collected during the random neutral walks indicated that phenotypic accessibility was dramatically increased through use of the mappings. As for the RNA-folding landscapes, neutral drift allowed constant discovery of new phenotypes. Thus, the probability of evolutionary search becoming isolated in regions of genotype space from which higher fitness transitions were not possible was greatly decreased. This conjecture was supported by performing adaptive fitness walks which extended the random neutral walks to include steps to higher fitness phenotypes whenever possible. Several fitness functions were employed that created challenging search spaces containing many local optima. Use of the mappings allowed much higher fitnesses to be achieved than through use of a direct encoding for which the adaptive walks readily became trapped at local optima.

These results suggest therefore that the primary objectives were achieved by these mappings. Neutral networks were introduced into genotype space that allowed the constant discovery of new phenotypes. As a result the efficacy of a simple evolutionary algorithm was enhanced in comparison to a direct encoding. However a key simplification was made for this analysis - the consideration of only single-point mutations. Thus, only small movement was possible within genotype space at any given reproductive event. It was argued that this restriction was necessary to make the analysis computationally tractable and that higher mutation rates or more complex genetic operators would only enhance the extent of the neutral networks. However, realistic evolutionary algorithms typically allow greater movement in genotype space which may allow shallow local optima to be negotiated when using a direct encoding. Thus, consideration of

single-point mutations alone may have artificially hampered evolutionary progress when using a direct encoding. Work was reviewed that compared the random Boolean network mapping with a direct encoding using varying mutation rates and larger search spaces [46]. This work suggested that performance using the direct encoding was at least as good as that for the random Boolean network mapping when higher mutation rates were used. Use of the mappings did not therefore appear to increase the efficacy of a more realistic evolutionary algorithm in comparison to a direct encoding.

The reasons for the relative lack of performance of the RBN mapping in this case were elucidated in chapter 4 by returning to the four identified properties of the RNA-folding landscapes. Two of these properties, large-scale neutrality and neutral networks, were successfully introduced into the landscapes as described above. The analysis also revealed that the mappings successfully introduced the property of shape-space covering. All the neutral networks were readily discovered both during the random neutral walks and when the global features of the search space were probed using a series of random samples. However it was also found that each of the neutral networks was *equally likely* to be discovered, use of the mappings had not therefore introduced any large-scale biases into the search space. Thus, only three of the four identified properties of RNA self-organisation were generated through use of these mappings.

It was argued that it was this lack of common structures or phenotypes that limited the efficacy of the RBN mapping. In the absence of any bias or structure in genotype space, each of the neutral networks is equally likely to be discovered from any location within the space. The evolutionary process thus reduces to random search with elitism and as the size of the search space is increased the performance would quickly become unacceptable. To increase the efficacy of an evolutionary algorithm in large spaces therefore, the mappings must introduce *some* structure into genotype space. It was shown that localised structure was apparent but that this structure was heavily influenced by the direct encoding i.e. the more probable phenotype discoveries were those that would have been allowed for through use of a standard direct encoding. It was argued that a more fundamental restructuring of the space was required that emphasised certain phenotypes and thus generated large-scale biases in the search space.

The cellular automaton and random Boolean network have the ability to generate such biases. They are both discrete dynamical systems that settle into attractors as they are iterated. Not all final states are therefore possible as the system naturally tends towards one of a relatively small number of final behavioural patterns. However, this ability was effectively removed during the design of the mappings. Both models were updated for a fixed number of iterations and the resulting state was interpreted as the phenotype. This approach alleviated the difficulties in identifying the attractors and assigning each attractor to a suitable phenotype. However, modifying the underlying “physics” of the models in this way resulted in removal of the key property of common phenotypes and ultimately reduced the efficacy of the mappings.

This highlights a fundamental difficulty in developing abstract genotype-phenotype mappings. It may be possible to emphasise certain phenotypes, however doing so arbitrarily is not likely to be a fruitful approach. Any biases must be introduced with due consideration to the problem at hand so that the common phenotypes tend to be of high fitness. Design of effective self-organising genotype-phenotype mappings is thus likely to require the use of domain knowledge. This is the subject of the following section.

7.1.2 Growing telecommunication networks

In chapter 5 the work was extended to address a real-world application, the evolutionary design of telecommunications networks. A self-organising mapping was introduced that made use of domain knowledge in the form of planning rules that are used by network designers. These rules were used to iteratively modify a network design based on the locally perceived condition of the current network and its simulated environment. In effect, the rules allowed a network to be "grown" using instructions that were encoded in the genotype. The parameters of the rules were encoded into a binary genotype and it was shown that many encoded rules produced the same network design i.e. neutrality was introduced into the search space. Exhaustive enumeration of this space revealed that neutral networks had also been created. However, the extent of these networks was restricted due to fragmentation which limited neutral drift and ultimately created local optima. The analysis revealed that the root cause of this fragmentation was the choice of a binary encoding which caused similar parametric values to be large distances apart in genotype space. Adoption of a Gray encoding was shown to remove this fragmentation and the associated local optima.

In contrast to the abstract mappings, this approach introduced very strong biases into the search space. The vast majority, over 99.9%, of feasible network designs could not be generated through use of the growth process. The relatively small-scale of the initial problem allowed the fitness of every feasible network design to be calculated. This analysis revealed that the domain knowledge inherent to the mapping ensured that those that could be generated included very high quality network designs. They did not, however, include the very best network designs. It was shown that this was due to the design of certain features of the planning rules at the heart of the self-organising process. In effect, misleading domain knowledge was encoded into the mapping which had a detrimental effect on the search space and resulted in the global optimum being impossible to generate. When the planning rules were re-designed such that the inherent domain knowledge more closely reflected the properties required of high quality networks, the resulting growth process was able to generate the global optimum. This demonstrated an important point when designing genotype-phenotype mappings of this nature. Domain knowledge must be used with great care in order to ensure that the resulting biases in the search space are in favour of phenotypes of high fitness.

The above analysis demonstrated that amenable evolutionary search spaces could be created through use of a self-organising growth process. However, in order to demonstrate the efficacy of the approach it was also necessary to shown that it compared favourably with more traditional approaches on challenging problems. Chapter 6 presented results of such a comparison. Larger scale and more complex network design problems were tackled using both the self-organising approach and an encoding in which the network structure was directly represented in the genotype. It was discovered that the biases created by the self-organising process allowed evolution to very rapidly discover high quality network designs. In contrast, evolutionary progress was dramatically slowed through use of the direct encoding. On relatively small scale and low complexity problems, the direct encoding eventually allowed the discovery of higher quality network designs. However, this was no longer the case as the scale and complexity of the problem was increased.

For the most challenging problems considered in this work, the fitness of the network designs discovered through the coupling of evolution and the self-organising genotype-phenotype mapping was higher than those discovered through use of evolution alone. It is possible that comparable quality designs would eventually have been discovered using the latter approach given enough time. However, the relative performance of evolution using the direct encoding progressively decreased as the scale and complexity of the problem was increased and it is likely that this trend would continue with further such increases. The advantages of the self-organising approach would thus be further emphasised. This conjecture can be supported by examining the nature of the search space created through use of a direct encoding. As the problem is directly represented in the genotype, an increase in the scale of the problem produces a corresponding increase in the size of the genotype and hence that of the search space. This has a detrimental effect on the speed of evolution as evidenced by the dramatic slowing of evolutionary progress as the scale of the problem was increased in this work. In addition, increasing the complexity of the problem typically results in increasing the strength and number of competing constraints. As discussed in chapter 6, this typically increases the epistatic interactions between the genes of a direct encoding and increases the ruggedness of the associated search space. The end result is to exacerbate the problem of local optima and thus reduce the effectiveness of the direct encoding.

In contrast, the self-organising approach need not suffer from either of these fundamental difficulties. The information encoded in the genotype does not directly represent the problem and therefore its size is not directly proportional to the size of the problem. As the scale of the problem was increased in this work, the genotype encoding the planning rules was unchanged and thus the size of the search space remained constant. In addition, use of the self-organising approach opens up the possibility of more effectively handling the competing constraints inherent to more complex problems. The knowledge of how to do so can be represented in the planning rules which are subsequently encoded in the genotype. The competing constraints are thus

effectively handled by the growth process and any detrimental effects on the search space are reduced. The self-organising approach thus compares very favourably with the direct encoding. It expedites evolutionary progress and scales well to larger and more complex problems.

These results suggest therefore that the primary objectives were achieved by this approach. For the initial small-scale problem, exhaustive enumeration revealed that the self-organising genotype-phenotype mapping generated amenable evolutionary search spaces with strong biases towards high-fitness phenotypes. These spaces contained large sets of genotypes mapping onto each phenotype that were connected into neutral networks. These networks allowed access to higher fitness phenotypes and local optima were removed from the search space. In addition, consideration of more complex problems revealed that the approach resulted in numerous advantages over a typical direct encoding. These results are very promising and suggest that further investigation of the approach would be warranted. A natural next step for such investigations is presented in the following section.

7.2 Future Work

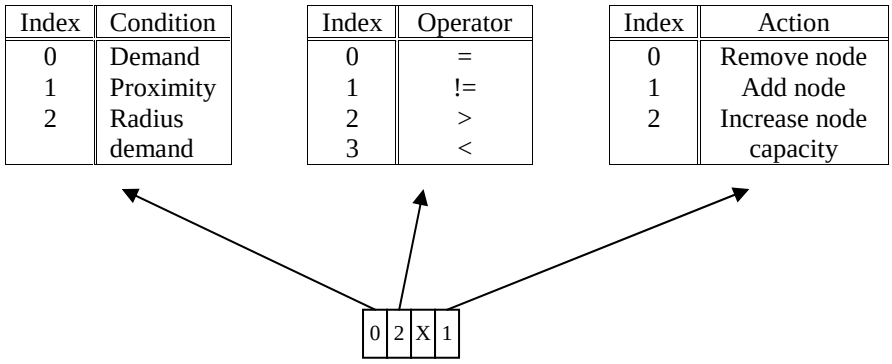
Self-organisation's constraining effect on evolution was particularly striking in the genotype-phenotype mapping created for the evolutionary design of telecommunication networks. Less than 0.1% of feasible phenotypes were possible outcomes of the mapping. These constraints were several orders of magnitude stronger than were found in the RNA-folding landscapes. In addition, some of the phenotypes that could be generated by the mapping were represented by relatively few genotypes and thus could not be considered as common phenotypes. This suggests that the genotype-phenotype mapping created for this work may have imposed too tight a constraint on evolution. This lack of flexibility was evident in the results presented in chapter 6 in which the direct encoding was eventually able to discover better quality networks for the simpler problems. Use of the self-organising process allowed evolution to very quickly discover high quality networks but then did not allow it the flexibility to perform the necessary fine-tuning of these networks. This relative lack of flexibility is inherent to the approach as it essentially employs heuristics to disallow many network designs. However, increasing evolutionary flexibility by relaxing some of the constraints imposed by the self-organising process has the potential to further increase the efficacy of the approach. A method of achieving this is presented in the following section.

7.2.1 Evolving planning rule structure

The approach introduced in chapter 5 and further explored in chapter 6 used a small number of planning rules with a fixed structure that was not under evolutionary control. Thus, only the parameters of these rules were encoded in the genotype. For the initial problem explored in chapter 5 only three parameters were eventually encoded and modification of these parameters allowed evolution to tune the growth process. However with such a small number of parameters

the tuning was very limited and thus the range of possible phenotypes was reduced. More complex problems would require a larger number of planning rules and would inevitably result in a larger number of encoded parameters. A total of 15 parameters were encoded for the most complex problems explored in chapter 6. However, the options available to evolution were still severely restricted. In order to increase the flexibility with which evolution could tune the growth process and hence increase the number of possible phenotypes, the structure of certain planning rules could be placed under evolutionary control along with the associated parameters.

One method of achieving this would be to maintain a set of observable conditions about the network and its environment together with a set of possible actions that can be carried out in that environment. In addition, a set of operators would be required that enabled these conditions and actions to be combined to produce the planning rules. The genetic encoding would thus consist of a number of indexes into these sets which would together define the encoded planning rule. Figure 7.1 shows a very simple example of such an encoded rule. The table that the indexes refer to would be defined by their placement within the genotype i.e. a prescribed sequence of encoded parameters would be defined. More complex planning rules would also be possible by defining a fourth table that included logic operators such as AND and OR that allowed a number of conditions to be combined into a single rule. The genotype would thus be composed of a number of sections that each defined a planning rule with a given number of conditions. Encoding the section type would enable the number of conditions and the sequence of associated parameters to be ascertained by the growth process. It would also be necessary to define the number of encoded rules of each type. One method of doing this would be to fix the number of each type and allow rules to be enabled or disabled through use of a genetic switch. In addition, gene duplication or deletion events could be employed to allow the number of rules of each type to be placed more fully under evolutionary control.



IF (Demand > X) ***THEN*** (Add node)

Figure 7.1: An example of a genetically encoded rule structure. The genotype encodes indexes into tables of observable conditions, operators and actions together with the associated parameters to form a planning rule.

This approach would significantly increase the options open to evolution and would undoubtedly allow a greater number of phenotypes to be produced by this approach. However, a balance needs to be struck between flexibility and constraint. Removing all planning rule structure would effectively remove much of the domain knowledge that allowed very good networks to be rapidly discovered by evolution. If high quality and proven planning rules exist for a particular scenario then it would be prudent to make good use of this knowledge. The most fruitful approach would likely therefore be a combination of rules with fixed structure and evolved rules. This would also open up the possibility of tuning the balance between flexibility and constraint by modifying the number of each type.

This method is one of a number of ways that the structure of the planning rules could be evolved. Another possibility is classifier systems that provide much of the machinery required to evolve rule sets for a given task [105]. In this approach, a population of rules is maintained with a number of conditions that are matched to the current environmental context. The rules that match this context are triggered and become candidates for execution. The strongest candidate is typically chosen for such a fate and its associated action is carried out. The classifier system also includes a system for rewarding rules that resulted in beneficial actions and the better rules tend to produce offspring that take the place of poorer quality rules within the population. In this way the structure of the rule set becomes better matched to the task at hand. Incorporation of techniques such as these would create a sophisticated method of evolving network designs with the potential to yield significant advantages over more traditional approaches.

7.3 Concluding remarks

The coupling of evolution and self-organisation has enormous consequences both for theories of biological evolution and the design of artificial evolutionary systems. The introduction of neutrality into the search space is but one such consequence, however, this alone fundamentally challenges our views of evolutionary adaptation. The use of these concepts in the design of genotype-phenotype mappings for evolutionary network design has been shown to be a very promising approach that generates significant advantages. It is likely that the efficacy of evolution for many other problems within the communications domain and beyond can be significantly enhanced through due consideration of self-organisation and neutrality. The first steps along this path have been taken and the route leads to a more complete emulation of the creative power of biological evolution within our artificial systems – an enticing goal indeed to fill us with energy for the journey ahead.

References

1. A. W. Burks, *Essays on Cellular Automata*. Urbana: University of Illinois Press, 1970.
2. Aderonke Babajide, Ivo L. Hofacker, Manfred J. Sippl, and Peter F. Stadler, *Neutral Networks in Protein Space: A Computational Study Based on Knowledge-Based Potentials of Mean Force*, Santa Fe Institute Working Paper, 96-12-085, 1996.
3. Aderonke Babajide, Robert Farber, Ivo L. Hofacker, Jeff Inman, and Alan S. Lapedes, *Exploring Protein Sequence Space Using Knowledge-Based Potentials*, Santa Fe Institute Working Paper, 98-11-103, 1998.
4. Adrian Thompson, "An Evolved Circuit, Intrinsic in Silicon, Entwined in Physics," in *Proceedings of the First International Conference in Evolvable Systems*, T Higuchi, M Iwata, and W Liu, Eds. Berlin: Springer, 1996.
5. Alan Turing, "The Chemical Basis of Morphogenesis," *Philosophical Transactions of the Royal Society*, vol. B 237, pp. 37-72, 1952.
6. Brian Goodwin, *How The Leopard Changed Its Spots: The Evolution of Complexity*. Great Britain: Phoenix, 1997.
7. Brian Goodwin and C Briere, "A Mathematical Model of Cytoskeletal Dynamics and Morphogenesis in *Acetabularia*," in *The Cytoskeleton of the Algae*, D Menzel, Ed. Boca Raton, FL: CRC Press, 1992, pp. 219-38.
8. Charles Darwin, *The Origin of Species : By Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life*. Bantam Classic, 1999.
9. Christian Reidys, Peter F. Stadler, and Peter Schuster, *Generic Properties of Combinatory Maps: Neutral Networks of RNA Secondary Structures*, Santa Fe Institute Working Paper, 95-07-058, 1995.
10. Christian V. Forst, Christian Reidys, and Jacqueline Weber, *Evolutionary Dynamics and Optimization: Neutral Networks as Model Landscapes for RNA Secondary-Structure Folding-Landscapes*, Santa Fe Institute Working Paper, 95-10-094, 1995.
11. D'Arcy Thompson, *On Growth and Form. Revised Edition.* Dover, New York.: Cambridge University Press, 1992.
12. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
13. Erik van Nimwegen and James P. Crutchfield, *Optimizing epochal evolutionary search: population-size dependent theory*, Santa Fe Institute Working Paper, 98-06-046, 1998.
14. Erik van Nimwegen and James P. Crutchfield, *Optimizing epochal evolutionary search: population-size independent theory*, Santa Fe Institue Working Paper, 98-06-046, 1998.
15. Erik van Nimwegen and James P. Crutchfield, *Neutral Evolution of Mutational Robustness*, Santa Fe Institute Working Paper, 99-03-021, 1999.

16. Erik van Nimwegen, James P. Crutchfield, and Melanie Mitchell, *Statistical Dynamics of the Royal Road Genetic Algorithm*, Santa Fe Institute Working Paper, 97-04-035, 1997.
17. Erwin Bonsma, Mark Shackleton, and Rob Shipman, "Eos - an Evolutionary and Ecosystem Research Platform," *BT Technology Journal*, vol. 18, no. 4, October, 2000.
18. *Evolutionary Computation: a Fossil Record*, David Fogel, Ed. John Wiley & Sons, 1998.
19. *Evolutionary Design by Computers*, Peter J. Bentley, Ed. Morgan Kaufmann, 1999.
20. F Jacob and M Monod, "Genetic Regulatory Mechanisms in the Synthesis of Proteins," *Journal of Molecular Biology*, vol. 3, no. 318, 1961.
21. F. Gray, 2 632 058, Pulse code communication, 1953.
22. F. J. Dyson, "A Model for the Origin of Life," *Journal of Molecular Evolution*, vol. 18, pp. 344-350, 1982.
23. G Nicolis and I Prigogine, *Self-Organization in Nonequilibrium Systems*. New York: Wiley-Interscience, 1977.
24. G. Celli, E. Costamagan, and A. Fanni, "Genetic Algorithms for Telecommunication Network Optimization," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1995, Vol. 2, pp. 1227-1232.
25. G. Cocho, G. Perez-Pascual, J. Ruis, and F. Soto, "Discrete Systems, Cell-Cell Interactions and the Color Patterns of Animals," *Journal of Theoretical Biology*, vol. 125, pp. 437, 1987.
26. G. Struhl, P. Johnston, and P. A. Lawrence, "Control of *Drosophila* Body Pattern by the Hunchback Morphogen Gradient," *Cell*, vol. 69, pp. 237-249, 1992.
27. Gregor Mendel, *Experiments in Plant Hybridization*, J.H. Bennett, Ed. London: Oliver and Boyd, 1965.
28. H. P. Schwefel, *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. New York: John Wiley & Sons, 1995.
29. I. Dayan, D. Stauffer, and S. Havlin, "Cellular Automata Generalization of the Weisbuch-Atlan Model for Immune Response," *J. Phys. A, Math. Gen.*, vol. 21, no. 10, pp. 2473, 1988.
30. I. Rechenberg, *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien der Biologischen Evolution*. Stuttgart: Frommann-Holzboog Verlag, 1994.
31. Inman Harvey and Adrian Thompson, "Through the Labyrinth Evolution Finds a Way: A Silicon Ridge," in *Proceedings of the First International Conference in Evolvable Systems*, T Higuchi, M Iwata, and W Liu, Eds. Berlin: Springer-Verlag, 1996, pp. 406-422.
32. Ivo L. Hofacker, Martijn A. Huynen, Peter F. Stadler, and Paul E. Storlorz, *RNA Folding on Parallel Computers*, Santa Fe Institute Working Paper, 95-10-089, 1995.
33. Ivo L. Hofacker, Peter Schuster, and Peter F. Stadler, "Combinatorics of RNA Secondary Structures," *SIAM Journal of Discrete Mathematics*, 1996.

34. Ivo L. Hofacker, Walter Fontana, Peter F. Stadler, L. Sebastian Bonhoeffer, and Manfred Tacker, *Fast Folding and Comparison of RNA Secondary Structures*, Santa Fe Institute Working Paper, 93-07-044, 1993.

35. J. D. Murray, "How the Leopard Got Its Spots," *Scientific American*, vol. 258(3), pp. 62, 1988.

36. J. D. Schaffer and A. Morishima, "An Adaptive Crossover Distribution Mechanism for Genetic Algorithms," in *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, J. J/ Grefenstette, Ed. Lawrence Erlbaum Associates, 1987, pp. 36-40.

37. J. D. Watson and F. H. C. Crick, "Molecular Structure of Nucleic Acids," *Nature*, vol. 171, pp. 737-738, 1953.

38. J. F. Miller and P. Thomson, "Cartesian Genetic Programming," in *Proceedings of the Third European Conference on Genetic Programming. Lecture Notes in Computer Science*. Berlin: Springer-Verlag, 2000, Vol. 1802, pp. 121-132.

39. J. H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor: University of Michigan Press, 1975.

40. J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

41. Jean Pierre Boon, "Lattice Gas Automata: Theory, Simulation, Implementation," *Journal of Statistical Physics*, vol. 68(3/4), 92.

42. John Maynard Smith and Eors Szathmary, *The Origins of Life: From the Birth of Life to the Origins of Language*. Oxford University Press, 1999.

43. John Maynard-Smith, "Natural Selection and the Concept of a Protein Space," *Nature*, vol. 225, pp. 563-564, 1970.

44. John R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

45. John Von Neumann and A. W. Burks, *Theory of self-reproducing automata*. Urbana: University of Illinois Press, 1966.

46. Joshua D. Knowles and Richard A. Watson, "On the Utility of Redundant Encodings in Mutation-Based Evolutionary Search," in *Proceedings of The Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII): LNCS 2439*, Springer Verlag, Ed., 2002.

47. L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*. New York: Wiley, 1966.

48. Lewis Wolpert, Rosa Beddington, Thomas Jessel, Peter Lawrence, Elliot Meyerowitz, and Jim Smith, *Principles of Development*. Oxford University Press, 2002.

49. Lionel Barnett, "Tangled Webs: Evolutionary Dynamics on Fitness Landscapes with Neutrality," MSc dissertation. School of Cognitive and Computing Science: University of Sussex, 1997.

50. Lionel Barnett, "Ruggedness and Neutrality - the NKp Family of Fitness Landscapes," in *Proceedings of Sixth International Conference on Artificial Life - Alife VI*, C. Adami, R. K. Belew, H. Kitano, and C. Taylor, Eds. Cambridge, MA: MIT Press, 1998, pp. 18-27.

51. Lionel Barnett, "Netcrawling - Optimal Evolutionary Search with Neutral Networks," in *Proceedings of the Congress on Evolutionary Computation: CEC'01*. IEEE Press, 2001.

52. M Eigen and J McCaskill, "Molecular Quasispecies," *Journal of Physical Chemistry*, vol. 92, pp. 6881-6891, 1988.

53. M Kimura, *The Neutral Theory of Molecular Evolution*. Cambridge University Press, 1983.

54. M. Gerhardt and H. Schuster, "A Cellular Automaton Describing the Formation of Spatially Ordered Structures in Chemical Systems," *Physica D*, vol. 36, pp. 209-221, 1989.

55. M. Newman and R. Engelhardt, "Effect of Neutral Selection on the Evolution of Molecular Species," *Proceedings of the Royal Society of London B*, vol. 256, pp. 1333-1338, 1998.

56. M. Van Dyke, *An Album of Fluid Motion*. The Parabolic Press, 1982.

57. Manfred Tacker, Peter F. Stadler, Erich G. Bornberg-Bauer, Ivo L. Hofacker, and Peter Schuster, *Algorithm Independence Properties and RNA Secondary Structure Predictions*, Santa Fe Institute Working Paper, 96-04-016, 1996.

58. Marc Ebner, Mark Shackleton, and Rob Shipman, "How Neutral Networks Influence Evolvability," *Complexity*, 2001.

59. Marc Ebner, Patrick Langguth, Juergen Albert, Mark Shackleton, and Rob Shipman, "On Neutral Networks and Evolvability," in *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001, Vol. 1, pp. 1-8.

60. Marc Toussaint and Christian Igel, "Neutrality: A Necessity for Self-Adaptation," in *Proceedings of the Congress on Evolutionary Computation: CEC'02*. IEEE Press, 2002.

61. Mark C. Sinclair, "Evolutionary Telecommunications: A Summary," in *Evolutionary Telecommunications: Past, Present and Future*. 1999 Genetic and Evolutionary Computation Conference. Orlando, Florida, July, 1999. (Bird-of-a-feather Workshop).

62. Mark C. Sinclair, "Evolutionary Algorithms for Optical Network Design: A Genetic-Algorithm/Heuristic Hybrid Approach," PhD thesis. Department of Electronic Systems Engineering: University of Essex, 2001.

63. Mark Shackleton, Rob Shipman, and Marc Ebner, "An Investigation of Redundant Genotype-Phenotype Mappings an Their Role in Evolutionary Search," in *Proceedings of the 2000 Congress on Evolutionary Computation*, 2000, Vol. 1, pp. 493-500.

64. Martijn A. Huynen, Peter F. Stadler, and Walter Fontana, "Smoothness Within Ruggedness: The Role of Neutrality in Adaptation," *Proceedings of the National Academy of Science*, vol. 93, pp. 397-401, 1995.

65. Martijn Huynen, *Exploring Phenotype Space Through Neutral Evolution*, Santa Fe Institute Working Paper, 95-10-100, 1995.

66. Martijn Huynen, Peter F. Stadler, and Walter Fontana, *Evolutionary Dynamics of RNA and the Neutral Theory*, Santa Fe Institute Working Paper, 95-01-006, 1995.

67. Moshe Sipper, *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*. Berlin: Springer-Verlag, 1997.

68. Nicholas Geard, Janet Wiles, Jennifer Hallinan, Bradley Tonkes, and Ben Skellett, "A Comparison of Neutral Landscapes - NK, NKp and NKq," in *Proceedings of the Congress on Evolutionary Computation: CEC'02*. IEEE Press, 2002.

69. Nick Jakobi, "Encoding Scheme Issues for Open-Ended Artificial Evolution," in *Proceedings of Parallel Problem Solving From Nature: PPSN IV. Lecture Notes in Computer Science*. Berlin: Springer, 1996, Vol. 1141.

70. O. Miglino, S. Nolfi, and D. Parisi, "Discontinuity in Evolution: How Different Levels of Organization Imply Pre-Adaptation," in *Adaptive Individuals in Evolving Populations*, R.K. Belew and M. Mitchell, Eds. Reading, MA: Addison-Wesley, 1996, pp. 399-415.

71. P Erdos and A Renyi, *On the Evolution of Random Graphs*. Institute of Mathematics, Hungarian Academy of Sciences, 1960.

72. P. Calégari, F. Guidec, P. Kuonen, and D. Kobler, "Parallel Island-Based Genetic Algorithm for Radio Network Design," *Journal of Parallel and Distributed Computing*, vol. 47, no. 1, pp. 86-90, 1997.

73. P. Chardaire, A. Kapsalis, J.W. Mann, V.J. Rayward-Smith, and G.D. Smith, "Applications of Genetic Algorithms in Telecommunications," in *Proceedings of the 2nd International Workshop on Applications of Neural Networks to Telecommunications (IWANNT'95)*. Stockholm, Sweden, 1995, pp. 290-299.

74. P. Hogeweg, "Cellular Automata as a Paradigm for Ecological Modelling," *Applied Mathematical Computing*, vol. 27, pp. 81-100, 1988.

75. P. Husbands, T. M. Smith, N. Jakobi, and M. O'Shea, "Better Living Through Chemistry: Evolving GasNets for Robot Control," *Connection Science*, vol. 10, no. 3-4, pp. 185-210, 1998.

76. P. J. Angeline, "Adaptive and Self-Adaptive Evolutionary Computations," in *Computational Intelligence: A Dynamic Systems Perspective*, M. Palaniswami, Y. Attikiouzel, R. Marks, D. B. Fogel, and T. Fukuda, Eds. IEEE Press, 1995, pp. 152-163.

77. P. J. Angeline, "Two Self-Adaptive Crossover Operations for Genetic Programming," in *Advances in Genetic Programming*. Cambridge, MA: MIT Press, 1996, Vol. 2.

78. P.R.L. Gondim, "Genetic Algorithms and the Location Area Partitioning Problem in Cellular Networks," in *Proceedings of the IEEE 46th Vehicular Technology Conference*. Atlanata, Georgia, USA, 1996, Vol. 3, pp. 1835-1838.

79. Peter F. Stadler, *Fitness Landscapes Arising from the Sequence-Structure Map of Biopolymers*, Santa Fe Institute Working Paper, 97-11-082, 1997.
80. Peter Schuster, *Landscapes and Molecular Evolution*, Santa Fe Institute Working Paper, 96-07-047, 1996.
81. Peter Schuster, *Genotypes with Phenotypes: Adventures in an RNA Toy World*, Santa Fe Institute Working Paper, 97-04-036, 1997.
82. Peter Schuster and Peter F. Stadler, *Sequence Redundancy in Biopolymers: A Study on RNA and Protein Structures*, Santa Fe Institute Working Paper, 97-07-067, 1997.
83. Peter Schuster and Walter Fontana, *Chance and Necessity in Evolution: Lessons from RNA*, Santa Fe Institute Working Paper, 98-11-107, 1998.
84. Peter Schuster, Walter Fontana, Peter F. Stadler, and Ivo L. Hofacker, *From Sequences to Shapes and Back: A Case Study in RNA Secondary Structures*, Santa Fe Institute Working Paper, 93-07-045, 1993.
85. Peter Schuster, Walter Fontana, Peter F. Stadler, and Ivo L. Hofacker, "From Sequences to Shapes and Back: A Case Study in RNA Secondary Structures," *Proceedings of the Royal Society of London B*, vol. 255, pp. 279-284, 1994.
86. Philip Ball, *The Self-Made Tapestry: Pattern Formation in Nature*. Oxford University Press, 1999.
87. Q. Ouyang and H. L. Swinney, "Transition from a Uniform State to Hexagonal and Striped Turing Patterns," *Nature*, vol. 352, pp. 610, 1991.
88. R. A. Watson, G. S. Hornby, and J. B. Pollack, "Modelling Building-Block Interdependency," in *Proceedings of Parallel Problem Solving from Nature: PPSN V*. Berlin: Springer-Verlag, 1998, pp. 97-106.
89. R. A. Watson and J. B. Pollack, "Hierarchically Consistent Test Problems for Genetic Algorithms," in *Proceedings of the 1999 Congress on Evolutionary Computation*. Washington D.C., USA: IEEE Press, 1999, pp. 1406-1413.
90. R. A. Watson and J. B. Pollack, "Hierarchically Consistent Test Problems for Genetic Algorithms: Summary and Additional Results," in *GECCO-99 Late Breaking Papers*, 1999, pp. 292-297.
91. R. A. Watson and J. B. Pollack, "Recombination Without Respect: Schema Combination and Disruption in Genetic Algorithm Crossover," in *Proceedings of the 2000 Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2000.
92. Richard Dawkins, *The Selfish Gene*. Oxford: Oxford University Press, 1976.
93. Richard Dawkins, *The Blind Watchmaker*. Harlow: Longman, 1986.
94. Richard Sole and Brian Goodwin, *Signs of Life: How Complexity Pervades Biology*. New York: Basic Books, 2000.
95. Rob Shipman, "Genetic Redundancy: Desirable of Problematic for Evolutionary Adaptation?" in *Proceedings of the 4th International Conference on Artificial Neural Networks and Genetic Algorithms*. Springer-Verlag, 1999.

96. Rob Shipman and Mark Shackleton, "Issues in Designing a Neutral Genotype-Phenotype Mapping," in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*. IEEE Press, 2002, pp. 1360-1365.

97. Rob Shipman, Mark Shackleton, and Inman Harvey, "The Use of Neutral Genotype-Phenotype Mappings for Improved Evolutionary Search," *BT Technology Journal*, vol. 18, no. 4, October, 2000.

98. Rob Shipman, Mark Shackleton, Marc Ebner, and Richard Watson, "Neutral Search Spaces for Artificial Evolution: A Lesson from Life," in *Proceedings of the Seventh International Conference on Artificial Life*, Mark A. Bedau, John S. McCaskill, Norman H. Packard, and Steen Rasmussen, Eds. MIT Press, 2000, pp. 162-169.

99. Rob Shipman, Mark Shackleton, Marc Ebner, and Richard Watson, "Neutral Search Spaces for Artificial Evolution: A Lesson from Life," in *Proceedings of the Artificial Life 7 Workshops*, Carlo C. Maley and Ellis Boudreau, Eds., 2000.

100. Rob Shipman, Paul Botham, and Paul Coker, "Coupling Developmental Rules and Evolution to Aid in Planning Network Growth," *BT Technology Journal*, vol. 18, no. 4, October, 2000.

101. S. F. Elena, V. S. Cooper, and R. E. Lenski, "Punctuated Evolution Caused by Selection of Rare Beneficial Mutations," *Science*, vol. 272, pp. 1802-1804, 1996.

102. S. J. Gould and , "Exaptation: A Crucial Tool for an Evolutionary Psychology," *Journal of Social Issues*, vol. 3, pp. 43-65, 1991.

103. S. Kirkpatrick, C. D. Gelatt, and H. P. Vecci, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 671, 1983.

104. S. Kondo and R. Asai, "A Reaction-Diffusion Wave on the Skin of the Marine Angelfish *Pomacanthus*," *Nature*, vol. 376, pp. 765, 1995.

105. S. W. Wilson, "ZCS: A Zeroth Level Classifier System," *Evolutionary Computation*, vol. 2, no. 1, pp. 1-18, 1994.

106. S.J. Gould and N. Eldredge, "Punctuated Equilibria: The Temp and Mode of Evolution Reconsidered," *Paleobiology*, vol. 3, pp. 115-151, 1977.

107. Seth Bullock, "Smooth Operator? Understanding and Visualising Mutation Bias," in *Proceedings of the Sixth European Conference on Artificial Life (ECAL 2001)*, J. Keleman and P. Sosik, Eds. Springer-Verlag, 2001, pp. 602-612.

108. Sewall Wright, "The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution," in *International Proceedings of the Sixth International Congress on Genetics, volume 1*, D.F. Jones, Ed., 1932, pp. 356-366.

109. Stephen Wolfram, "Cellular Automata as Models of Complexity," *Nature*, vol. 311, no. 4, pp. 419-424, 1984.

110. Stephen Wolfram, "Universality and Complexity in Cellular Automata," *Physica D*, vol. 10, pp. 1-35, 1984.

111. Stuart A. Kauffman, *The Origins Of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

112. Stuart A. Kauffman, *At Home in the Universe: The Search for Laws of Complexity*. Oxford University Press, 1995.

113. T Hoshino and M Tsuchida, "Manifestation of Neutral Genes in Evolving Robot Navigation," in *Proceedings of the Fifth International Conference on Artificial Life*, 1996.

114. T Ohta, "Population Size and Rate of Evolution," *Journal of Molecular Evolution*, vol. 1, pp. 305-314, 1972.

115. T. Bäck, "Self-Adaptation in Genetic Algorithms," in *Proceedings of the First European Conference on Artificial Life*, F. J. Varela and P. Bourgine, Eds. MIT Press, 1992, pp. 263-271.

116. T. M. Smith and A. Philippides, "Nitric Oxide Signalling in Real and Artificial Neural Networks," *BT Technology Journal*, vol. 18, no. 4, pp. 140-149, 2000.

117. T. M. Smith, A. Philippides, P. Husbands, and M. O'Shea, "Neutrality and Ruggedness in Robot Landscapes," in *Proceedings of the Congress on Evolutionary Computation: CEC'2002*. IEEE Press, pp. 1348-1353.

118. T. M. Smith, P. Husbands, and M. O'Shea, "*Not* Measuring Evolvability: Initial Investigations of an Evolutionary Robotics Search Space," in *Proceedings of the Congress on Evolutionary Computation: CEC 2001*. IEEE Press, 2001, pp. 9-16.

119. T. M. Smith, P. Husbands, and M. O'Shea, "Neutral Networks and Evolvabillity with Complex Genotype-Phenotype Mapping," in *Proceedings of the European Conference on Artificial Life: ECAL 2001*. Springer, 2001, pp. 272-281.

120. T. M. Smith, P. Husbands, and M. O'Shea, "Neutral Networks in an Evolutionary Robotics Search Space," in *Proceedings of the Congress on Evolutionary Computation: CEC'2001*. IEEE Press, 2001, pp. 136-145.

121. T. M. Smith, P. Husbands, P Layzell, and M. O'Shea, "Fitness Landscapes and Evolvability," *Evolutionary Computation*, vol. 10, no. 1, pp. 1-34, 2001.

122. T. Routen, "Genetic Algorithm and Neural Network Approaches to Local Access Network Design," in *Proceedings of the 2nd International Symposium on Modelling, Analysis & Simulation of Computer & Telecommunication Systems (MASCOTS'94)*. Durham, N. Carolina, USA, 1994, pp. 239-243.

123. Tina Yu and Julian F. Miller, "Neutrality and the Evolvability of Boolean Function Landscape," in *Proceedings of the 4th European Conference on Genetic Programming. Lecture Notes in Computer Science*. Springer-Verlag, 2001, Vol. 2038, p. 204 ff.

124. Tina Yu and Peter Bentley, "Methods to Evolve Legal Phenotypes," in *Proceedings of the Fifth International Conference on Parallel Problem Solving From Nature. Lecture Notes in Computer Science.* Berlin: Springer-Verlag, 1998, Vol. 1498, pp. 280-291.

125. Vesselin Vasseliv, Dominic Job, and Julian F. Miller, "Scalability Problems of Digital Circuit Evolution," in *Proceedings of the 2nd NASA/DOD Workshop on Evolvable Hardware*. Los Alamitos, CA: IEEE Computer Society, 2000, pp. 55-64.

126. Vesselin Vasseliv, Dominic Job, and Julian F. Miller, "Towards the Automatic Design of More Efficient Digital Circuits," in *Proceedings of the 2nd NASA/DOD Workshop on Evolvable Hardware*. Los Alamitos, CA: IEEE Computer Society, 2000, pp. 151-160.

127. Vesselin Vasseliv and Julian F. Miller, "The Advantages of Landscape Neutrality in Digital Circuit Evolution," in *Lecture Notes in Computer Science*, Julian F. Miller, Adrian Thompson, P Thomson, and Terry C. Fogarty, Eds. Third International Conference on Evolvable Systems: From Biology to Hardware. Springer-Verlag, 2000, pp. 252-263.

128. Vesselin Vasseliv and Julian F. Miller, "Embedding Landscape Neutrality to Build a *Bridge* from the Conventional to a More Efficient Three-Bit Multiplier Circuit," in *Proceedings of the 2nd Genetic and Evolutionary Computation Conference*. San Francisco, CA: Morgan Kaufmann, 2000, p. 539.

129. Vesselin Vasseliv, Julian F. Miller, and Terry C. Fogarty, "On the Nature of Two-Bit Multiplier Landscapes," in *Proceedings of the First NASA/DOD Workshop on Evolvable Hardware*. Piscataway, NJ: IEEE Computer Society Press, 1998, pp. 36-45.

130. Vesselin Vasseliv, Julian F. Miller, and Terry C. Fogarty, "Digital Circuit Evolution and Fitness Landscapes," in *Proceedings of the Congress on Evolutionary Computation (CEC'99)*. Piscataway, NJ: IEEE Press, 1999, Vol. 2, pp. 1299-1306.

131. Vesselin Vasseliv, Julian F. Miller, and Terry C. Fogarty, "Digital Circuit Evolution: The Ruggedness and Neutrality of Two-Bit Multiplier Landscapes," in *Evolutionary Hardware Systems*. Savoy Place, London: IEE Press, 1999, pp. 6/1-6/4.

132. Walter Fontana, Peter F. Stadler, Erich G. Bornberg-Bauer, Thomas Griesmacher, and Ivo L. Hofacker, *RNA Folding and Combinatory Landscapes*, Santa Fe Institute Working Paper, 92-10-051, 1992.

133. Walter Fontana and Peter Schuster, *Shaping Space: The Possible and the Attainable in RNA Genotype-Phenotype Mapping*, Santa Fe Institute Working Paper, 97-11-081, 1997.

134. Walter Fontana and Peter Schuster, *Continuity in Evolution: On the Nature of Transition*, Santa Fe Institute Working Paper, 98-04-030, 1998.

135. Walter Grüner, Robert Giegerich, Dirk Storthmann, Christian Reidys, Jacqueline Weber, Ivo L. Hofacker, Peter F. Stadler, and Peter Schuster, *Analysis of RNA Sequence Structure Maps by Exhaustive Enumeration*, Santa Fe Institue Working Paper, 95-10-099, 1995.

136. Wolfgang Banzhaf, "Genotype-Phenotype-Mapping and Neutral Variation - A Case Study in Genetic Programming," in *Proceedings of Parallel Problem Solving from Nature - PPSN III*, Y. Davidor, H. P. Schwefel, and R. Manner, Eds. Berlin: Springer-Verlag, 1994, pp. 322-332.