Artificial evolution with Binary Decision Diagrams: a study in evolvability in neutral spaces

by

RICHARD MARK DOWNING

A thesis submitted to

The University of Birmingham

for the degree of

DOCTOR OF PHILOSOPHY

School of Computer Science
The University of Birmingham
June 2008

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

Abstract

This thesis introduces a new approach to artificial evolution employing Binary Decision Diagrams as the genotypic representation, and uses it to study evolvability issues. The approach is referred to as *Evolving Binary Decision Diagrams using Inherent Neutrality* (EBDDIN). The aims are twofold. Firstly, to develop an evolutionary algorithm with a capability to address many of the issues facing the field of evolutionary computation today. Secondly, to develop a deep understanding of the concepts and mechanisms that facilitate within that context.

The issue of evolvability, loosely defined as the capacity to evolve, permeates the field of evolutionary computation. For reasons that are not yet fully understood, current approaches to artificial evolution fail to exhibit a pace and extent of evolutionary change so readily exhibited in nature. In order to resolve this discrepancy, the field of evolutionary computation must characterise, understand and apply evolvability to artificial evolution. If this can be achieved, systems of artificial evolution will become much more capable than they are presently.

The approach is developed with the primary practical and theoretical issues regarding evolvability in mind, exploiting inherent properties of the Binary Decision Diagram representation where possible. It is then used as a computational model for studying evolvability issues, giving particular emphasis to the role of neutrality, modularity, gradualism, robustness and population diversity, and the interplay between them. Carefully designed, controlled experiments elucidate the mechanisms and properties that facilitate evolvability and its evolution. The implications are then considered regarding the new understandings developed and the fidelity with the characteristics of biological evolution.

Pleiotropic patterns which bias the phenotypic effects of random mutation are found to emerge. These configurations represent the variation component of evolvability and are subject to indirect selection. Higher-level structural configurations (i.e. OBDD variable orderings) that better facilitate such patterns emerge as a logical consequence. Neutrality plays the crucial role of facilitating fitness-conserving exploration and completely alleviating local optima for the domain of Boolean functions. Population diversity allows evolvability traits to compete and evolve, ultimately facilitating the evolution of evolvability. The search is insensitive to the starting point and the absence of initial diversity, requiring only minimal diversity generated from gradual genotypic variation.

Gradual evolution in a search space that is free of local optima by way of neutrality can be a viable alternative to problematic evolution on multi-modal landscapes, exhibiting search characteristics that have greater fidelity to natural evolution. This is a fruitful direction for research that is directed at the problem of facilitating evolvability in artificial evolution, and it may lead to evolutionary systems that are open-ended.

Acknowledgements

I thank my former colleagues that offered me advice and support, particular in the challenging early days when I was finding my feet: Dave Parker, Hang Zhang, Vineet Khare, John Woodward, Felipe Orihuela-Espina, Dominique Chu, and many others. Thanks also to my supervisor, Ata, for questioning and providing feedback, and to the other members of my thesis group for listening to my ideas and reading my reports. I was very much inspired to do research by the professionalism, dedication and enthusiasm for research of my BSc and MSc supervisors, Marta Kwiatkowska and Julian Miller respectively; I am very grateful to them both. My research has also benefitted from contact with researchers from distant institutes who have kindly addressed my queries or provided valuable discussion, including Stefan Droste, Dan McShea, Marc Kirschner and Jeffrey Horn. Finally, I thank my family for putting up with me, and Amanda for her unwavering support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

According to Darwin, the wonderment that is life on earth has emerged as a result of little more than natural selection and heritable variation. Darwinism today is widely accepted. Many people strive to acquire a deeper understanding of the mechanisms and properties that underly evolution in nature. Some wish to harness its power to solve difficult problems or create artificial life: if man can harness the power that has produced man himself then he will truly become powerful, godlike, and his existence will be much the better for it. Today, and despite considerable research effort, attempts at artificial evolution fail to fulfill the promise. Addressing this shortcoming is the principal motivation for this thesis and the preoccupation of the field of evolutionary computation.

1.1 Aims & motivations

Darwinian evolution [23] is undoubtedly amongst the most profound of all scientific theories. One has only to look around at the great diversity, complexity, and beauty of life to appreciate its power and potential. The field of *evolutionary computation* (EC) seeks to mimic and exploit understandings of natural evolution for artificial evolution. The motives are usually to either simulate and better understand natural evolution, or to harness its potential to generate

novel solutions to challenging problems.

While it cannot be disputed that current approaches to artificial evolution often produce results that are novel, useful and competitive with human-engineered solutions, few in the field would dispute that what can presently be produced is woeful in comparison to nature's efforts. These artificial systems fail to scale to more challenging problems: they lack *evolvability*, the capacity to evolve, which prohibits the desirable pace and extent of evolution that is so readily exhibited in nature. Many reasons are postulated for this shortcoming and the issue of evolvability permeates through the field of EC. Yet, evolvability remains amongst the most contentious and poorly understood of evolutionary concepts. Even the definition of evolvability is not agreed upon within the field, even less are these evolutionary algorithms (EAs) that can exhibit it to the extent they ought. EAs are sometimes designed or augmented with features that contradict accepted evolutionary principles in an effort to overcome the limitations, but the problem persists. Only when a better understanding of evolvability is achieved will the gap between the actuality and the perceived potential of artificial evolution begin to evaporate.

The aim of this thesis is to make inroads towards a deeper understanding of the mechanisms and properties of a system of artificial evolution that can facilitate evolvability. For that purpose, this study introduces and exploits a EA which employs Binary Decision Diagrams (BDDs) [17] for the genotypic representation. This new approach to evolving BDDs is referred to as *Evolving BDDs with Inherent Neutrality* (EBDDIN). EBDDIN exhibits some interesting search characteristics that can have a close biological fidelity in some respects, and this makes it suitable for this study. Contrasts and comparisons with both natural evolution and other other systems of artificial evolution will be drawn in order to give context to the findings of the thesis.

1.2 Thesis questions

The following questions and discussion serve to clarify the objectives of the thesis. The questions are repeated in the concluding chapter along with summaries of the answers to those questions provided by the thesis.

The choice of representation is considered vital when designing an EA. BDDs are a state-of-the-art data structure for representing Boolean functions in some fields because they have some very attractive properties in terms of efficiency. Yet, BDDs are not a representation often considered in EC for Boolean or other functions encodable into BDD form. While some attempts at exploiting BDDs for synthesising functions have been made, they have not proved particularly successful.[1] A principal thesis question is, therefore: *Is there a better approach to exploiting the BDD data structure for artificial evolution?*.

Evolvability comes from the genetic operators transforming the representation in ways that leave intact adapted traits, but perturbs maladapted traits [2], but how? Wagner & Altenberg [122] suggest that *evolutionary modules* may be important here, in which there is a coupling between the effects of a gene group and phenotypic trait group. Hansen [49], however, suggests that other pleiotropic patterns may facilitate evolvability. The obscurity about what it is about a genotype that facilitates evolvability inhibits its study. A further thesis question is, therefore: *How might evolvability be represented within the genotype and what properties make one genotype more evolvable than another?*. Once this question is answered, an answer to the related question of how evolvability evolves is more easily addressed. In the absence of any intelligent design, evolvability in nature must have emerged under normal evolutionary forces: *What are the properties and mechanisms that facilitate the emergence and evolution of evolvability?*. A related question addresses the controversial role of neutrality: *What is the role of neutrality in evolutionary search?*.

These are challenging, open questions. This thesis brings a fresh perspective

[1] Please note that this thesis is concerned with synthesising functions in the BDD representation, and not in only minimising a given function's BDD representation.

to addressing them from the context of EBDDIN.

1.3 Research perspective

The focus of this thesis is to develop a new EA employing BDDs as the genotypic representation, and to develop a deep understanding of evolvability within that context. The best way to acquire an understanding of evolvability is to study a system that exhibits it to the desired extent. Biological systems exhibit such evolvability. However, biological systems are extremely complex, making their study difficult. EC systems, on the other hand, are simpler but typically fail to exhibit the evolvability that biological systems do: studying these systems, therefore, is easier but has limited potential for gleaning new insights. The EBDDIN approach introduced and developed in this thesis exhibits the desirable evolvability properties in limited circumstances, facilitating both a feasible and fruitful study of evolvability.

A holistic perspective is necessarily taken, appreciating all the practical and conceptual issues pertaining to evolvability in artificial evolution. The more practical issues are, for example, avoiding becoming trapped in local optima, managing bloat, choosing parameters, and the cost of fitness evaluation. The more conceptual issues considered are neutrality, modularity, robustness, gradualism, landscapes and the the evolution of evolvability. Only by taking a holistic perspective can the properties and mechanisms that facilitate evolvability be revealed and understood. Controlled experiments and careful analysis are used to elicit this understanding and enrich the development of the EBDDIN approach.

A limited number of scalable problem instances are chosen from the general class of Boolean functions. These instances are not chosen for their performance similarities under the EBDDIN approach, but mostly for their contrasts in performance. It is the understanding of these contrasts in performance combined with knowledge of the BDD representations of these functions that aids in being able to generalise the findings across the wider problem domain.

1.4 Contributions of the thesis

The thesis makes the following contributions, listed in no particular order:

1. **A new and improved approach to evolving BDDs that also facilitates dynamic variable reordering.**

   Previous approaches to synthesising functions in the BDD representation using EC have not proved particularly successful. The EBDDIN approach proves superior in the number of fitness evaluations required on a number of benchmark functions against the compared approaches [128, 103, 118]. EBDDIN also facilitates dynamic variable reordering.

2. **A new EA to be exploited by the EC community.** The field of EC is broad, exploiting many representations and approaches, some of which have become common place or standardised, but each of which comes with its own limitations and range of applicability. In EBDDIN, EC researchers and practitioners have another EA to draw upon.

3. **An investigation into how evolvability is represented within the genotype.**

   Chapter 6 investigates this within the context of EBDDIN and characterises evolvability as being represented by pleiotropic patterns which constrain the phenotypic effects of random genotypic mutation. Mutation is then able to perturb maladapted traits while leaving adapted traits relatively unperturbed. Modularity plays an important role by controlling the relative exposure of adapted traits to variation.

4. **A computational model demonstrating the evolution of evolvability within the context of a static environment.** Chapters 8 & 9 demonstrate that structural configurations (i.e. OBDD variable orderings) emerge as a logical consequence of better facilitating evolvability. This supports Dawkins [24] claim that evolvability is a selectable trait and is selected for in evolution indirectly. The impact on an evolutionary run is

shown to be dramatic, sometimes achieving a linear rate of fitness increase with a static fitness function for almost the duration of the run. Reisinger et al. [94] and Turney [117] have previously argued that a dynamic fitness function was necessary for the evolution of evolvability.

5. **An investigation into the role of neutrality in evolutionary search.**

   Neutrality is shown to be a crucial, and the most important, source of exploration within the context of EBDDIN and the problem domain employed. Neutrality is shown to completely alleviate local optima for the class of Boolean fitness functions (chapter 5). Chapter 7 shows that neutrality is a better source of exploration than increased mutation rate, and is cost-free under EBDDIN, allowing the trade-off between exploration and exploitation to be circumvented. Chapters 5, 6 & 9 also shows that neutral variation in evolvability traits is important for the evolution of evolvability.

6. **An investigation into the role of population diversity in a search space that is free of local optima by way of neutrality, which shows diversity to facilitate the evolution of evolvability.**

   Chapter 9 shows that although the search performance is tolerant of the absence of initial diversity, the diversity that results from gradual mutation is important and facilitates the evolution of evolvability by permitting the evolvability traits of individuals to compete via their offspring. Selection is thus seen as acting indirectly on evolvability traits, and by this, favourable evolvability traits are propagated. Barnett [10] had previously argued that population diversity was not beneficial in such a space, and that a minimally-sized population was optimal.

7. **An investigation into the the role of mutation rate in a search space that is free of local optima by way of neutrality.**

   Chapter 7 investigates this with EBDDIN, and the notion of search space *adequacy* is introduced. Gradual genotypic mutation is shown to provide

the best performance over *completely adequate* search spaces for the problems investigate. It is further argued that the balance typically assumed necessary between exploration and exploitation can be avoided by utilising cost-free exploration from explicitly neutral mutations. The most gradual mutation maximises heredity not only of fitness, but also of favourable evolvability traits.

8. **Gradual evolution in a search space that is free of local optima by way of neutrality present as a viable and more plausible alternative to problematic evolution on multi-modal landscapes. This is demonstrated on Boolean functions within the context of EBDDIN.**

   The thesis adds to a growing body of evidence that a desirable pace and extent of evolution may require a search space structured differently to that envisaged by Wright [127]. The problems of evolution on Wright's landscapes have been well-studied, and no general evolutionary mechanism for successfully navigating them has been discovered. This thesis shows that gradual evolution in a search space that is free of local optima by way of neutrality can exhibit characteristics with a greater consistency with biological evolution.

   No claim is made that it is always possible to formulate a given problem to be free of local optima by way of neutrality to permit effective gradual evolution. The claim is only that it may be possible with the right choice of representation and operators and other components, and should be given serious consideration as an alternative to problematic evolution on multi-modal landscapes.

1.5 Limitations of the thesis

While the practicality of the EBDDIN approach is emphasised throughout, it is not developed towards, or applied to, real-world applications. The focus is on

understanding the fundamental issues pertaining towards evolvability using a limited number of benchmarks rather than applications to real-world problems. It is hoped and expected, however, that inspired and derived approaches towards real-world problems will follow as a result of this thesis.

The findings of the thesis are applicable only within the context of EBDDIN and the problem domain employed herein: any attempt to generalise the findings of this thesis will be explicitly stated. While EBDDIN and the problem domain are non-trivial and have a wide potential applicability, individual or subsets of the findings and contributions cannot easily be generalised to other EAs or other problem domains. This is why a holistic perspective has been emphasised for the thesis. It is the interactions between all components that are argued to facilitate evolvability and its evolution within the context of EBDDIN. Changing any one significant aspect will likely yield very different results. This is not to say the findings of this thesis cannot be generalised to different EAs and problem domains. Indeed, if evolvability is to be better understood in general, attempts must be made to generalise and build on the findings of this thesis. In doing so, however, the holistic perspective promoted herein should be adhered to.

1.6 Overview

The first four chapters introduce and provide background for the rest of the thesis. Chapters 5 through 9 represent the core of the thesis. The core chapters are reasonably self-contained, though a best appreciation will come from reading the thesis in its entirety.

Chapter 2 provides relevant background. The findings of natural evolution are reviewed, focussing on Darwinian evolution and the contemporary theories of *neutrality* and *facilitated variation* that have followed. This provides important background for relating the search characteristics of the approach developed herein to that of natural evolution.

Chapter 3 reviews some of the main issues in EC. The concept of evolvabil-

ity is first introduced and discussed. The nature of search spaces, how they are commonly depicted as multi-modal landscapes, and their relevance for hypothesising the problem and mechanisms of evolution are also discussed. The problem of bloat is reviewed. Finally, the postulated role of neutrality in EC is also discussed.

Chapter 4 reviews some of the other approaches to evolving BDDs.

Chapter 5 introduces EBDDIN. The role of neutrality inherent in the BDD representation is given particular emphasis, being the principal defining characteristic of the EA. The search space is shown to be free of local optima by way of neutrality for the class of Boolean functions. Methods to reduce the cost of fitness evaluation are detailed. Bloat management is discussed. The EBDDIN approach is then compared on a sample of benchmarks to other approaches to evolving BDDs. Finally, the benefits of the approach as a computational model for studying evolvability issues is argued.

Chapter 6 investigates the role of modularity (i.e. reuse of subfunction) in facilitating evolvability and the emergence of favourable pleiotropic configurations. Notions of modularity are first disambiguated. Modularity, in conjunction with neutrality, is then argued to facilitate evolvability by permitting an emergent bias in pleiotropic influence.

Chapter 7 looks at the role of mutation rate in facilitating evolvability. It is argued that the commonly perceived necessity for a trade-off between exploration and exploitation is avoidable where the cost of evaluating neutral offspring can be circumvented. Exploration through cost-free neutrality is argued to be preferable over exploration through increased mutation rate.

Chapter 8 extends the EBDDIN approach to include dynamic variable reordering, demonstrating good variable orderings as an emergent property. This is shown to be consistent with the evolution of evolvability.

Chapter 9 demonstrates that population diversity facilitates the evolution of evolvability. The effects of the loss of the initial diversity in the local-optima-free space is shown to be negligible, highlighting that search is highly-independent

of the starting configuration and that periodic diversity loss is benign. A larger population is shown to better maintain evolvability traits than a smaller population. The combination of neutrality and diversity is shown to be particularly effective for performance.

Chapter 10 concludes the thesis, lists the contributions and discusses the implications. Suggestions for future directions are also given.

1.7 Publications associated with this thesis

Richard M. Downing "Evolvability via modularity-induced mutational focussing", *Proc. of Eleventh European conference on Genetic Programming*, Naples, Italy, 26th–28th March 2008.

Richard M. Downing "On population size and neutrality: facilitating the evolution of evolvability", *Proc. of Tenth European conference on Genetic Programming*, Valencia, Spain, 11th–13th April 2007.

Richard M. Downing "Evolving Binary Decision Diagrams with emergent variable orderings", *Proc. of Parallel Problem Solving from Nature IX (PPSN IX)*, Reykjavic, Iceland, September 9th–13th 2006, Springer LNCS.

Richard M. Downing "Neutrality and gradualism: encouraging exploration and exploitation simultaneously with Binary Decision Diagrams", *Proc. of the 2006 IEEE World Congress on Computational Intelligence (WCCI 2006)*, Vancouver, Canada, July 16th–21st 2006, IEEE Press.

Richard M. Downing "Evolving Binary Decision Diagrams using Implicit Neutrality" *Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, Edinburgh, UK, September 2nd–5th 2005, IEEE Press, 2107–2113.

Chapter 2

Background

This chapter provides some necessary background for the rest of the thesis. The foundations of biological evolution are introduced, including Darwinism, the neutral theory and the theory of facilitated variation. This is followed by an overview of the field of EC. Finally, an overview of the BDD data structure is given.

2.1 Biological evolution

Darwinism is the fundamental theory of evolution in nature. Since Darwin, evolutionary biology has made considerable advances stemming from the discovery of DNA as the hereditary material. The neutral theory provides insights to the nature of evolutionary change at the molecular level, complimenting Darwin's theory pertaining to evolution at the phenotypic level. The new theory of facilitated variation claims to fill a gap in Darwin's theory by explaining how random mutation can generate directed phenotypic variation. The following subsections provide an overview of these three theories, and this will be useful in relating the findings of this thesis to what is understood about evolution in nature.

2.1.1 Darwinism

Darwinian evolution [23] explains the diversity of life on earth. It postulates that all species have descended from common ancestors in a branching fashion. Darwin argues that differences in morphology between species are due to gradual modification in successive generations as populations diverge and adapt to their particular environments. The means of this, Darwin claims, is *natural selection*. Mayr [77] interprets Darwinism as consisting of five component theories. Although the founding of these five theories are not all attributed to Darwin, they all aid in understanding Darwinism as a whole. A summary of these five theories follows.

The nonconstancy of species Species change: they are nonconstant over time. Darwin was not the first to postulate this; other, both post-Darwinian and pre-Darwinian, evolutionary theories postulated the nonconstancy of species. However, it was Darwin that made this view popular.

Common descent All species descend from common ancestors in a branching manner. Darwin observed, from the geological (or *fossil*) record, that the more ancient a species, the more it appeared to differ from living species. Darwin also observed that an ancient species often resembles some intermediate between other extinct groups. Thus, Darwin concluded that all species are essentially part of the same family tree, or *grand system* as he called it. The theory of common descent is attributed entirely to Darwin.

Gradualism Builds on the theory of common descent by postulating that the morphological differences between species arise gradually, in small steps through a large number of intermediate forms. Evidence for gradualness posed some difficulty for Darwin: the living world and geological records did not exhibit a continuum of transitional forms. Furthermore, complex structures, such as the eye, did not lend themselves well to the notion of intermediacy. However, Darwin anticipated these and other difficulties, and successfully defended

gradualism against them [23, ch. 6].

Population thinking Or *population speciation*, identifies the population as the unit of evolution. A species may, however, be represented by several local populations spread over a wide and possibly discontinuous area. Moreover, this theory claims that a population possesses great variation and consists of unique individuals. It is the uniqueness of individuals within a population that allows natural selection to do its work, gradually changing the average composition of the population towards an average of individuals with favourable hereditary characteristics. Population thinking is attributed to Darwin.

Natural selection Postulates the means of differential reproduction of individuals in a population due to heritable variation. Within any population more individuals are born than survive to reproduce. Population thinking states that populations are made up of unique individuals, and it is reasonable to conclude that some variants will exhibit favourable characteristics in the ability to reproduce. Such individuals are said to have higher *fitness*. Moreover, these favourable characteristics will, at least in part, be heritable. Thus, reproductively favourable characteristics will be selected by nature to be passed on to future generations, each successive generation exhibiting a slightly increased proportion of individuals with the favourable characteristics. Conversely, variants exhibiting less favourable characteristics (lower fitness) will be less lightly to breed and pass on their traits. This process is natural selection, and a population will become better adapted to its environment as a result.

Mayr describes natural selection as a two step process [77, p.132] in which probability plays a significant role. Firstly, random variation is produced in progeny through random mating and changes in the genetic material. The second step sees reproductively superior variants prosper at the expense of the less fit. This second step is not entirely random and favours fitter variants, but it does have a random component also.

It is often good to think about natural selection as a process of *elimination*

instead of active selection [77, p. 130-131]. *Selection* implies that some particular variant is chosen as best as if evolution had some particular ideal in mind. Where, in reality, a whole range of differing variants might have similarly high reproductive success. This is important point as it promotes greater variety in the population, and facilitates greater exploration. Therefore, rather than selecting a best variant, natural selection eliminates lesser variants leaving a possible plethora of highly fit variants.

2.1.2 The neutral theory

Kimura [59] first proposed his *neutral theory of molecular evolution* in 1968. Sometimes called simply *neutral theory*, it claims that most evolutionary change at the molecular level is caused by *random genetic drift* rather than natural selection. Neutral theory in no way contradicts, or seeks to lessen the significance or validity of, Darwin's concept of natural selection: Darwin's natural selection remains the means of guiding adaptive phenotypic change. Rather, neutral theory offers understanding of the mechanism of evolutionary change at the molecular level.

Mutations in DNA occur randomly at a fairly low, but constant rate. Some of these mutations will be beneficial, contributing further to the adaptedness of the phenotype and thus being the subject of positive natural selection. Still more of these mutations will be deleterious and therefore eliminated by natural selection. However, the vast majority of mutations will be selectively neutral, or nearly neutral as they produce no phenotypic change, or exhibit slight or non-selective phenotypic change. These neutral mutations may propagate through a population; this is called *random genetic drift*.

Random genetic drift is responsible for large amounts of intraspecific genetic variation. However, neutral mutations resulting from random drift has latent potential for producing selectable phenotypic variation. At any time, previously neutral genetic material can become active. This is of great significance. If there was little intraspecific variation – that is, all genotypes of a species were very

similar – the possible range of genotypic variation across the species due to mutation would be limited due to low mutation rate. Progeny would effectively be exploring a highly restricted space of possible genotypes, limiting phenotypic diversity and the capacity for adaptedness of the species by natural selection. However, with large amounts of intraspecific genotypic variation, a similarly low mutation rate can explore a much more diverse genotype space, potentially producing much greater diversity of phenotypes for natural selection to operate on. Furthermore, due to random genetic drift, the genotype space being explored from generation to generation is forever changing.

2.1.3 Facilitated variation

Darwin explains the fate of phenotypic variation using his theory of natural selection. However, Darwin says nothing about the origin of phenotypic variation. Phenotypic variation is often assumed to arise purely randomly. Yet, phenotypic variation cannot be entirely random. Mammalian progeny, humans for example, are highly viable, typically varying in non-lethal ways. Clearly, if variation was truly random then a high rate of infant mortality and disability would result due to morphological defects: the body is not tolerant of even minor morphological disruption. Thus, variation appears to be directed towards non-lethality and selective conditions and is not entirely random.

Kirschner & Gerhart's theory of *facilitated variation* [60] seeks to explain the origin of phenotypic variation and fills a gap in evolutionary theory left by Darwin. The individual organism is elevated from being a passive target of natural selection to being fundamental in directing phenotypic variation from random mutation. Random genetic mutation does not produce random phenotypic mutation, but highly constrained phenotypic variation which is directed toward non-lethality and selective conditions.

Evidence for the theory draws heavily on *evolutionary developmental biology*, evo-devo. Evo-devo suggests that it is not the genes that determine the organism, but the relationship between the genes via the control of their ex-

pression in development. This, in turn, is determined by regulatory regions of DNA. This helps to explain the fact that few genes are required to make such complex organisms, about 22,500 for humans. By way of analogy, there may be only a few different types of Leggo pieces, but the variety of things that can be made from those types is unlimited. So, it is the multitude of ways in which few genes can be expressed that make possible the immense variety of the living world, and more besides.

At the heart of the theory of facilitated variation are the *conserved core processes*. Their function is to generate the phenotype from the genotype. Regulatory regions of DNA influence the deployment of the core processes to generate new phenotypes. The core processes have remained highly unchanged over time, emerging in a few intermittent waves of evolutionary innovation. It is changes to the deployment of the core processes that have been responsible for most evolutionary change.

The core processes are built in special ways so that they are easily linked together in new combinations to generate new, viable phenotypes. *Weak linkage* implies that the interactions between the core processes are not tightly-coupled, as is a car's engine parts, for example. The core processes exhibit a preconditioned and self-inhibited response to mutation in producing phenotypic variation. *Exploratory behaviour* ensures that new, viable phenotypic variation is achieved without the need for simultaneous change to many systems. For example, the deployment of the vascular system is highly adaptive in development, and will effectively adapt to oxygenate any new morphological variation that arises. *Compartmentation* involves specialising the behaviours of different genes and processes, facilitating independent evolution of different regions of the organism.

Thus, facilitated variation postulates that the phenotype is both constrained and deconstrained by the organism itself in how it varies in response to random mutation in DNA. It is constrained in that variation will be less lethal or damaging, and deconstrained where variation may be more appropriate to selective

conditions. The result is an increase in phenotypic diversity for Darwin's natural selection to act upon, and the rapidity of evolution is enhanced as a result.

2.2 Evolutionary computation

Evolutionary computation exploits some of the principles of natural evolution to automatically generate solutions to problems that are represented inside a computer. These problems may be of function optimisation, design, or computer programming, etc. An EA consists of the following:

- A *population* of candidate solutions, usually called *individuals*.
- A genotypic *representation* for individuals, much as DNA is to organic life.
- A set of variation operators for generating the offspring from the parent(s), i.e. mutation or recombination operators.
- A *fitness function* which evaluates candidate solutions and assigns each a fitness value.
- A *selection* mechanism for choosing which individuals from the population, based on their fitness value, will be parents, and which individuals will die off.

Together, the fitness function, representation and variation operators determine the structure of the search space, or *landscape*, as it is usually termed. Both the pace and extent of evolution is dependent on the structure of the search space and other parameters. However, the interactions of the components are complex and often unpredictable, making a good choice difficult. These and other issues in EC are discussed in chapter 3. The rest of this section discusses some of the traditional approaches to EC. Contemporary EC draws on all of the traditional approaches and contemporary evolutionary biology, and the field has moved towards a *uniformed approach* [58] in which the traditional and other

approaches to EC are considered specific instances of a more general class of evolutionary algorithms.

2.2.1 Evolutionary Programming

Evolutionary programming (EP) was developed by Fogel [42] as an alternative, evolutionary approach to artificial intelligence (AI). Earlier approaches to AI were primarily based on developing and utilising extensive knowledge bases; so-called *expert systems*. Expert systems try to mimic intelligent behaviour by observing and capturing it, so it can be reproduced later when similar circumstances arise: a *monkey see, monkey do* approach to AI. Of course, it is not possible to capture every possible circumstance and response to a situation, so the expert system seeks to develop heuristics that *generalise* to all situations by isolating the contributing factors in the decision making process. However, expert systems do not lend themselves well to generalisation, and prove brittle when unfamiliar circumstances arise.

Fogel took the view that trying to create artificial intelligence by modeling human intelligence was not the best approach. He argued that modeling the process by which human intelligence had emerged was a much better approach: modeling evolution itself. This would provide an alternative, possibly superior, way of creating artificial intelligence and also provide greater insight into the main properties of intellect by observing how it emerges. Fogel [42, p.36-39] likens the evolutionary approach to creating artificial intelligence to the mechanisation of the *scientific method*, which he asserts was *discovered* rather than *invented*, as it has always existed in nature.

1. Individuals in a population serve as hypothesis concerning some property of the environment.

2. The behaviour of the individuals serves as some prediction about some previously unknown aspect of the environment.

3. Natural selection, *independent verification*, rejects incorrect hypothesis.

4. New hypothesis are spawned from remaining, successful hypothesis, and the process repeats.

Increasingly successful hypothesis emerge as a result of this cycle, until some ultimately successful hypothesis is obtained.

In EP, individuals are represented as *finite state machines*, FSMs. Effectively, these are labeled, directed graphs where nodes represent states and labeled edges represent transitions and associated outputs. The FSM encodes the behaviour of the individual. A start state is defined, state transitions and outputs take place according to the inputs provided from the environment. Mutation of an individual is achieved through either the addition or removal of a state or changing of one or more of the labels. EP emphasises phenotypic mutation rather than genetic, and makes use of self-adaptation to influence mutation rate during the course of evolution.

2.2.2 Genetic Algorithms

Genetic Algorithms, or GAs, were developed by John Holland and colleagues at the University of Michigan [51]. Characteristics of GAs as opposed to other traditional EC methods are described as follows [40]:

- A fixed-length, binary representation of individuals.
- Selection is stochastically proportional to fitness.
- Predominant use of crossover as opposed to mutation.

To the GA, crossover is of paramount importance. The importance of crossover to the GA results from theoretical underpinnings; the so-called *building block hypothesis*, or BBH. The notion underlying the BBH is that, within a population, each individual may contain parts of the optimal solution: the *building blocks* of the solution. Individuals containing such building blocks will tend to be more fit than those that don't, and will be selected for to produce progeny. Then, by the utilising crossover variation operator on two selected parents, such building

blocks may be combined to produce larger building blocks that exhibit the combined fitness of their smaller contributors. The process continues until all the building blocks required of the optimal solution have been brought together.

A slightly different, more mathematical, slant on the BBH is the notion of *schema*. In a binary representation, a schema is a string of the alphabet {0, 1, #}, where # is the 'don't care' symbol. Effectively, a schema represents a building block. For example, the string ###10##1# is a schema for a binary representation of length $L=9$; the *order* of this schema is 3 because there are 3 0 or 1 symbols; the schema's *defining length* is 5 because there are 5 positions between the first and last symbols of the schema. Let us call this schema S. If S is a schema that contributes to high fitness, then it will propagate through the population under crossover if the defining length is relatively short, else, if the defining length is long relative to the representation, it will be disrupted by crossover. When S becomes ubiquitous, the dimensionality of the search is more focussed because the fourth, fifth and eighth positions are known, so the search focusses on those other positions which are unknown. As successful schemas become combined, then the dimensionality of the search becomes even more focused. Holland's *schema theorem* essentially predicts the extent to which a given schema will have propagated in the next generation. Criticisms of the schema theorem surround its lack of utility for predicting long term performance above and beyond the next generation.

2.2.3 Evolution Strategies

Evolution strategies, ES, originated for the purpose of optimisation in engineering design problems [7, 101]. Real-valued, fixed length vectors encode real-valued variables that are associated directly with aspects of the problem at hand. Mutation is typically Gaussian with expectation 0; the resulting, predominantly small, changes mimicking the gradualness of Darwinian evolution. Recombination is also used in which sections of vectors may be interchanged or shuffled, or corresponding values may be averaged. Selection typically follows

one of two schemes:

- $(\mu + \lambda)$ In this ES selection scheme, μ fittest parents are selected deterministically to produce λ offspring. Then, μ parents are selected from the sum of $\mu + \lambda$, and the process repeats. This scheme assumes $\mu \leq \lambda$.
- (μ, λ) This ES selection scheme is similar to the one described above in that μ fittest parents are selected deterministically to produce λ offspring. However, the difference is that the next μ parents are selected from only the λ offspring. This scheme assumes $\mu < \lambda$, because $\mu = \lambda$ would amount to a random walk.

Self-adaptation is also a feature of ES. *Strategy parameters* regarding mutation rate or recombination method may be assigned to individuals. The idea is to allow an individual to retain information about what which parameters are best for it, and allow that information to be used later.

2.2.4 Genetic Programming

Genetic Programming (GP), made popular by Koza [62], is a more recent branch of EC. GP is directed towards the automatic generation of computer programs and has its origins in the work of Smith [109], Cramer [22], Schmidhuber [104] and Forsyth [43]. In the most established form, the representation used is program trees of the programming language LISP. Because of the variable length of the GP representation, GP suffers from a phenomenon called *bloat*. Basically, bloat is the rapid growth in size of solution without improvement in fitness, and is discussed further in section 5.6.

Like the GA, Koza's GP makes predominant use of crossover as the primary variational operator. Crossover in GP works by swapping randomly selected subtrees of two parents to produce two offspring. The predominant use of crossover stems from theoretical underpinnings in a similarly manner to the GA. A BBH has also been considered with respect to GP, as have several variants of the schema theorem. It is argued that good building blocks, namely subtrees,

will propagate through the population in a similar manner to good schemas in a GA. However, GP is different and much more difficult to analyse mathematically than a GA because the GP representation has variable length where the GA has fixed length. There has been considerable debate over the applicability of the BBH to GP, and all GP schema theorems have proven inconclusive. The argument centres around the extent to which crossover is likely to preserve or disrupt building blocks. Langdon & Poli [69] provide a comprehensive review of the theoretical foundations of GP.

Other GP systems

While Koza's GP is probably the most established and popular form of GP, there are many other noteworthy GP systems. These offer alternative representations than LISP trees or attempt to overcome some of the limitations of Koza's GP. Linear GP (LGP) consists of a series of instructions from an imperative language or machine code [6, 87]. Cartesian GP (CGP) [84] employs an explicit mapping between the genotype (integer string) and phenotype (directed graph) representations and is reported to be sometimes able to handle bloat implicitly [80]. Grammatical Evolution (GE) [89] is form a GP that employs a grammar in Backus-Naur Form (BNF), which allows the user to restrict the search space according to some domain knowledge.

2.3 Binary Decision Diagrams

This section provides an overview of the BDD data structure. A discussion of how BDDs are exploited as a genotypic representation in this thesis is deferred until chapter 5.

Introduced by Lee [70] and further by Akers [1], a BDD is a rooted directed acyclic graph representing a function of the form $f(X) : \mathbb{B}^n \longrightarrow \mathbb{B}$. Each non-terminal is labelled with a Boolean variable $x \in X$ and has a *then* child and an *else* child, reflecting the fact that each non-terminal represents an *if-then-*

else operation on x. Terminals are labelled from $\mathbb{B}$. Given an assignment of values for X, the output is determined by traversing the BDD from the root to a terminal following the child indicated by each vertices' variable label value.

Bryant [16] introduced the *ordered* BDD (OBDD), which imposes a total ordering on the appearance of non-terminal labels along any path with π, the variable ordering. Thus, $\pi = [x_1, x_2, \ldots, x_n]$, an ordered list of variables, and $i < j$ must hold for each x_i followed by x_j along any path. It is not necessary that all $x \in \pi$ appear in a path. In this thesis the notation $[x_1, x_2, \ldots, x_n]$-OBDD is used to specify the ordering associated with an OBDD, or simply π-OBDD to emphasise the significance of the ordering without specifying it. The OBDD representation is derived from the Shannon expansion [106]. A Boolean function $f(x_1, \ldots, x_n)$ is decomposed into subfunctions, thus:

$$f = x_i \cdot f|_{x_i=1} + \bar{x}_i \cdot f|_{x_i=0}$$

where $f|_{x_i=b\in\mathbb{B}}$ is the restriction of x to the constant $b \in \mathbb{B}$. The decomposition of the subfunctions continues until the Boolean constants are reached. The resulting list of expressions may contain some redundancy (i.e. duplicate expressions (see [4] for further details).

Redundancy in an OBDD can be removed in two ways (see figure 2.1):

1. **Remove redundant tests** A nonterminal α that has both outgoing edges pointing to the same vertex β is redundant. Redirect all α's incoming edges to β.

2. **Remove duplicate vertices** If α and β are nonterminals and have the same variable label and same children, or α and β are terminals of the same value, one can be removed with its incoming edges redirected to the remaining vertex.

A *reduced* OBDD (ROBDD) is an OBDD that cannot have its complexity reduced further by the reductions described above. Bryant [16] has shown ROB-

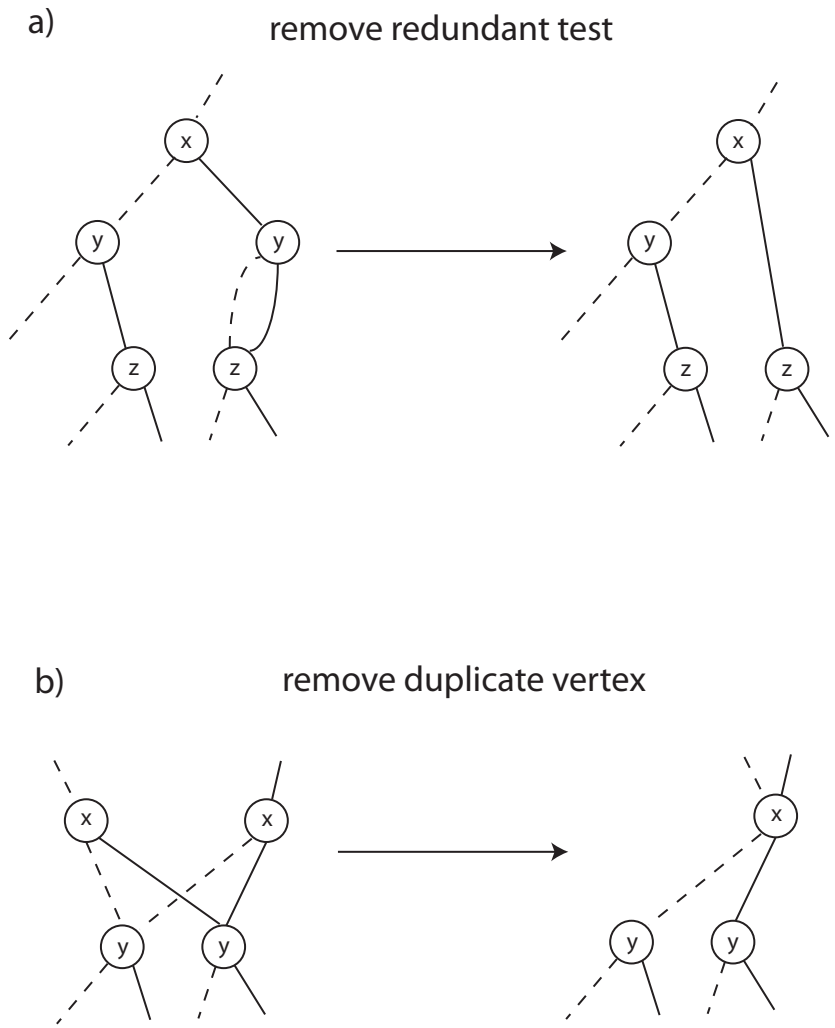


Figure 2.1: OBDD reduction mechanisms. a) Remove a redundant test where both child edges of a vertex point to the same child. b) Where two nonterminals have the same variable label an the same children, or two terminals have the same value, remove one and redirect the incoming edges accordingly.

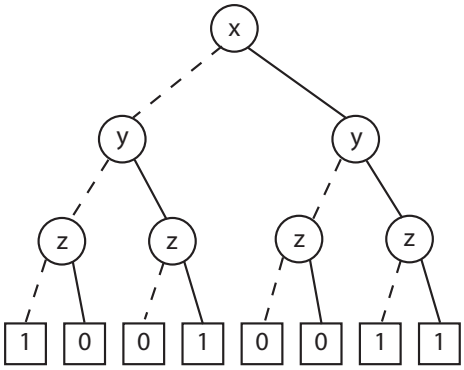
function	best π	worst π
multiplexer	linear	exponential
adder	linear	exponential
parity	linear	linear
symmetric	linear	quadratic
multiplier	exponential	exponential

Table 2.1: Function sensitivity to π in ROBDD complexity.

DDs to be *canonical forms*; meaning that each function has a unique ROBDD representation for each π, allowing easy equivalence and satisfiability checking. Figure 2.2 shows three different representations of the same function. The Binary Decision Tree (BDT), which is a specific form of OBDD, is easily constructed from the truth table, as every input vector (i.e. line in the truth table) has its own path in the BDT. The BDT is reduced to the canonical ROBDD representation by repeatedly applying the reduction mechanisms described above; many intermediate OBDDs are generated in the process.

The variable ordering can have a dramatic impact on the complexity of resulting ROBDD: in this thesis, the *complexity* of an π-ROBDD is the number of nonterminals it contains and the number of unique subfunctions. For example, the best π for the 6-bit multiplexer produces an ROBDD having complexity 7 while the the worst π results in and ROBDD having complexity 29 (see figure 2.3); for the 11-bit multiplexer, the best and worst ROBDD complexities are 15 and 509 respectively; for the 20-bit multiplexer it is 31 and over 130,000 respectively. For the n-bit multiplexer, the complexity grows linearly for the best π and exponentially for the worst. Not all functions are sensitive to the π. Parity is insensitive to π, having ROBDD complexity linear in number of variables for all π. The multiplier is insensitive to π also, but always has an ROBDD having exponential complexity. Table 2.1 details how the ROBDD complexity of the functions referred to this thesis are influenced by π. For many functions, choosing a good variable ordering will often result in a compact ROBDD representation of a function. However, the *variable ordering problem* is NP-complete in both optimal and approximate solutions [14, 107].

x	y	z	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



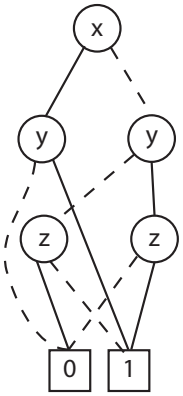


Figure 2.2: Truth table, BDT, and ROBDD representations of the function $f(x, y, z) = \bar{x}\bar{y}\bar{z} + xy + yz$. The BDT is easily created from the truth table. The BDT is then reduced to the canonical ROBDD form by applying the reduction mechanism depicted in figure 2.1. Many intermediate OBDDs are created in the reduction process.

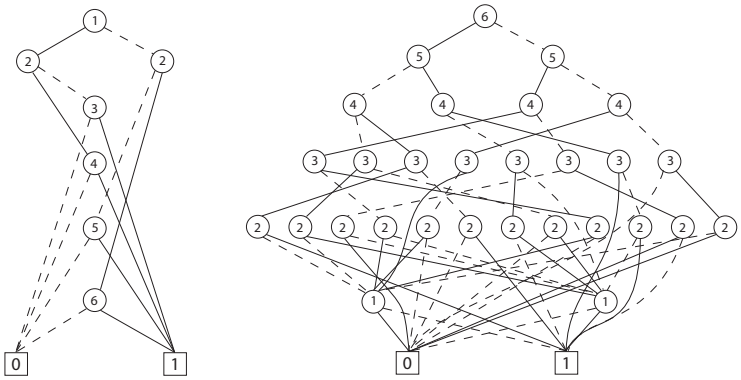


Figure 2.3: The effects differing π on ROBDD complexity for the 6-bit multiplexer function. Both of the above ROBDDs are representations of the same function but have different variable orderings. ROBDD complexity for this function is very sensitive to π, having a linear complexity in the number of variables for the best π, but exponential for the worst π.

Logical operations, such as conjunction and disjunction, can be done in time polynomial in the complexity of the ROBDD operands to the logical operation [4]. The operands must have the same π, and the result ROBDD will have same π.

Chapter 3

Issues in Evolutionary Computation

This chapter introduces the main issues in EC that are pertinent to this thesis. The concept of evolvability is first discussed with a view to providing a working understanding of the concept. The contentious requirements for the evolution of evolvability are also discussed regarding dynamic versus static fitness functions. Modularity in EC is reviewed. Conventional views of search spaces as multi-modal landscapes and the postulated roles of population diversity and mutation rate are then discussed, and debated. The problem of bloat is then reviewed. Finally, the controversial role of neutrality in evolutionary search is discussed.

3.1 Evolvability

The issue of evolvability and its evolution permeates the fields of both evolutionary computation and evolutionary biology, yet there is no consensus definition of the concept. The concept of evolvability can appear intangible, and the many and varied definitions can lead to confusion and misunderstanding. This section examines the concept of evolvability for the purpose of providing a working understanding for this thesis.

The most intuitive definition of evolvability is, perhaps, that it is *the capacity to evolve*[1]. This is also the most general definition. Is it the capacity of the population, or of the individual? Darwinism identifies the population as the unit of evolution, so the former seems the most appropriate answer, but this is not what many researchers mean when discussing evolvability. Evolvability is sometimes considered to be a property of the individual. The two perspectives are discussed in the following paragraphs.

Altenberg [2] defines evolvability as *the capacity of a population to produce variants fitter than any yet existing.* This definition implies that it is the population that is the possessor of evolvability. Under this definition, the rate at which a population adapts depends not only on how individuals in the population respond to random genotypic mutation, but also on selection. For example, random selection would result in a rate of adaptation approximating to zero; the opposing extreme, selecting only the fittest, will typically lead to premature convergence on a multi-modal landscape. Determining the best selection pressure to maximise the rate and extent of adaptation is extremely difficult, and will likely be variable for the duration of the run. Evolvability in this sense incorporates how selection is done: see de Jong [58, ch. 6].

However, Altenberg [2] also states of evolvability:

> "It comes from the genetic operators being able to transform the representation in ways that leave intact those aspects of the individual that are already adapted, while perturbing those aspects which are not yet highly adapted. Variation should be channeled towards those 'dimensions' for which there is selective opportunity."

This notion of evolvability implies it can also be a property of the individual. Variation resulting from random genotypic mutation may differ in its consequences between individuals within the population: individual A may be more likely to produce fitter descendants than individual B even though both have

[1] In this context, "evolve" is meant to express adaptation rather than just genetic change: genetic change can occur without adaptation, but not vice-versa

similar fitness. This makes A more evolvable than B. This type of evolvability is inherent in the genotypic representation of the two individuals, and independent of selection. However, discriminating between the more evolvable A and less evolvable B individual may only be achieved by subjecting them to the evolutionary forces of mutation and selection.

So, the evolvability of a population is dependent on selection, but optimising selection pressure is difficult and somewhat dependent on the distribution of the evolvability of individuals within that population. On the other hand, evolvability of the individual, while purely a genotypic property, can only be realised through variation and selection within a population. Thus, both the evolvability of the population and the evolvability of an individual are interdependent for practical purposes.

Conceptually, it makes sense to adopt an EC perspective for what can be referred to as a *selection component* and *variation component* of evolvability recognised by Kirschner & Gerhart [60] in evolutionary biology. The selection component results from how selection is done: the selection pressure, whether elitism is employed, stochastic or deterministic selection, etc. The variation component of evolvability results from how an individual's lineage varies with respect to fitness under random genotypic mutation. For the purposed of this thesis, variational evolvability is defined as: *the heritable potential to acquire increased fitness as a result of random genotypic mutation*, and is thus a property pertaining to individuals and is a selectable trait [24]. It will be clear from the context whether variational evolvability, selectional evolvability, or evolvability generally is being discussed.

Measuring evolvability

As stated above, evolvability must be determined by subjecting a population to the evolutionary forces of selection and heritable variation. Evolvability is then determined by the performance of an EA. In this thesis, performance will usually be measured as a count of the number of fitness evaluations required

to reach an ideal solution, or the *Average Evaluations to a Solution* (AES) [38] performance measure.

Another popular performance measure is Koza's *Computational Effort* (CE) [62]. CE is popular in the GP branch of EC, but not other branches of EC. CE has received a number of criticisms regarding its reliability as a performance measure in the GP community. Both Christensen & Oppacher [19] and Niehaus & Banzhaf [86] found that CE gave inaccurate estimates of the theoretical performance. Luke & Panait [76] concur, and argue that CE is philosophically flawed in its motivation because it is based on ideal solution counts. That is, CE is motivated by finding the ideal solution often, when perhaps finding a good solution on average is a better philosophy where the success rate for finding and ideal solution is less that 100%. Luke & Panait also point out that CE does not correlate with common EC measures outside of the GP branch of EC.

AES also looks for ideal solutions also, but it depends on a 100% success rate. Thus, AES does not rely on ideal solution counts, and averages performance over fitness evaluations required rather than the best average fitness of run. As stated by Luke & Panait, if the ideal solution can be found reliably, comparing performance is trivial. In this thesis, ideal solutions are found reliably with a 100% success rate or not at all, so AES is the preferred method of comparing performance.

3.1.1 Selection for evolvability

The variation component of evolvability is a selectable trait, though selection is not direct: this is a concept postulated by Dawkins [24]. Directly, selection targets functional fitness, not the variational evolvability of individuals. To select individuals directly on the basis of their variational evolvability would require looking at the lines descent of those individuals into the future and building that into the fitness function. Clearly, this defies the scientific principle of causality: the effect of heritable variation and selection cannot precede its cause, and therein lies much of the sceptism regarding the concept of evolving

evolvability.

However, if variational evolvability is taken to be the heritable *potential* of an individual's descendants to exhibit increased fitness, then evolvability is correlated with increased fitness. If individual A has better evolvability properties than individual B even though they have the same fitness, then A's descendants will typically be fitter than that of B, and A's better evolvability traits will be propagated while B's will not. Evolvability is, therefore, selected for as though by default; this is what Dawkin's refers to *second-order selection* for evolvability [24]. Evidence for second-order selection for evolvability within the context of EBDDIN will demonstrated throughout this thesis, in chapters 6, 8 and 9.

As stated at the start of this section, determining the variational evolvability of an individual requires looking at the lines of descent of that individual, so evolvability is not measurable without subjecting an individual to an EA action. However, it is possible to identify indicators of good evolvability properties a priori, and observe how those properties evolve. Chapter 8 identifies the OBDD variable ordering as highly relevant to evolvability for some functions. Quantitative indicators of evolvability can then be determined a priori by comparing EBDDIN performance on fixed variable orderings. Then, under dynamic variable ordering, the evolution of evolvability can be observed during an evolutionary run by analysis of how the variable ordering is evolving. This provides a lucid depiction of the evolution of evolvability.

3.1.2 Dynamic vs static fitness functions for the evolution of evolvability

Recently, Reisinger et al. [94] proposed an empirical measure for evolvability. As a premise for the method, he argued that there is little selection pressure for evolvability when the fitness function is static, stating:

> "If the fitness function is static, there is little need for evolvability, since any solution with high fitness, even one with low evolvability,

is likely to survive."

This view fails to recognise that evolution itself is testimony to selection for evolvability. Only lineages possessing high evolvability will breed successfully and progress towards high fitness. Even if a low evolvability lineage did make it to high fitness, as Reisinger suggests, it would not be able to compete with high evolvability lineages exhibiting similar fitness; the latter may even emerge from the former in the presence of neutral networks. This disparity would be particularly prominent in an environment where high fitness candidates are not allowed to persist indefinitely, such as in a biological context or (μ, λ) type ES. Low offspring viability would quickly eliminate low evolvability lineages to the favour of high evolvability lineages, where offspring viability is much better. Clearly then, there must be selection pressure for evolvability with static fitness functions, and this pressure will also move the population towards higher fitness.

This indirect selection for evolvability was something acknowledged by Turney [117], though he too employed a dynamic fitness function in his evolvability experiments. In contrast, this thesis investigates selection for evolvability within the context of static fitness functions, and this is demonstrated in chapter 8 particularly.

3.1.3 Robustness

A concept closely related to evolvability is robustness to mutational perturbations [121]. An important feature of biological evolution is that organisms are not only evolvable in a variational sense, but robust: that is, offspring are highly viable. An open question in this area is whether evolvability and robustness are positively or negatively correlated, as discussed by Lenski et al. [73]. Given that mutation is more likely to damage offspring than be beneficial, then greater robustness would appear to reduce the amount of variation for natural selection to act upon and, therefore, reduce evolvability, implying a negative correlation. Lenski also recognises that neutral mutations may also provide robustness, but

questions whether such robustness is beneficial or detrimental to evolvability:

> "Evolving populations can also become robust by finding regions of genotypic space that are flat because they contain a high proportion of neutral mutations." ... "If so, robustness and evolvability might again be positively, rather than negatively, correlated. However, deleterious mutations can also serve as stepping stones to adaptations [72]. Although deleterious mutations tend to be removed by selection and have shorter half-lives than neutral mutations, they are not instantly eliminated. Moreover, deleterious mutations may lead to genetic neighborhoods that are more promising, from the perspective of adaptation, than neutral mutations. In other words, neutral mutations are neutral precisely because they are isolated from important phenotypes, whereas deleterious ones must be connected to phenotypes that matter for fitness. It is unclear, therefore, whether neutral or deleterious mutations are more important for evolvability, and whether robustness associated with increased neutrality will promote or impede evolvability."

This thesis will argue that robustness and evolvability are positively correlated, and will show that neutral mutations are more important to evolvability than deleterious mutations within the context of EBDDIN. Chapter 5 argues that Lenski's assumption that neutral mutations occur in 'flat' regions of genotype space, and so are phenotypically isolated, is unsound within the context of EBDDIN. Chapter 7 demonstrates a positive correlation between robustness and evolvability, and shows that evolution better exploits 'stepping stones' that result from neutral rather than deleterious mutations.

3.2 Modularity

Modularity is often associated with evolvability in both EC and evolutionary biology. The intuitive idea of modularity is fairly straightforward. However,

there are many notions of modularity and the term is used in many different contexts to refer to a multitude of concepts. In computer science and engineering modularity provides systems designed as distinct functional components, each of which solve some particular aspect of the larger problem. A key benefit of this type of modularity is easy reuse of components, so that the same functionality does not have to be invented again and again. In evolutionary biology, concepts of modularity differ. *Developmental modules* are units of embryonic development that are largely independent of the context in which they occur [93]. These kinds of modules materialise as discernable phenotypic entities: a limb or eyeball, for example. *Evolutionary modules* are defined by their variational independence [122]: a pair of limbs vary as one rather than independently, for example. This type of modularity emerges in the genotype-phenotype map.

Chapter 6 further discusses notions of modularity, and emphasises a notion of modularity that facilitates functional reuse as most relevant for EBDDIN. The rest of this section presents on overview of modularity in EC, which typically employs a notion of modularity derived from modularity in computer science and engineering.

3.2.1 Modularity in EC

Many EAs incorporate modularity to their reported benefit. Modularity can help generalisation, facilitate reuse, and help prevent disruption by variation operators [7]. Furthermore, Woodward [126] has shown that, in the presence of modularity, the complexity of solution is independent of the chosen function set. Woodward further emphasises that modularity does not add expressitivity, but simply makes the representation more efficient in terms of its size.

Koza's GP [62] was enhanced with the incorporation of *Automatically Defined Functions* (ADF) [63]. This method requires the number and internal structure of modules to be prescribed by the user. Architecture-Altering Operations [64] extend the ADF concept so that the user is freed from prescribing the number and structure of ADFs. Rosca & Ballard's *Adaptive Representation*

[96] variant of Koza's GP, AR-GP creates a hierarchy of modules. The method works by extending the function set on the fly by periodically searching the population for blocks of code having high 'merit'.

Angeline & Pollack's [5] method of *Module Acquisition* utilises a library of modules. Modules to add to the library are selected at random: this is the *compress* operation. An inverse operation, *expand*, expands the module in the population and removes it from the library. Compressed components are protected from the variation operators. Unlike Rosca's AR-GP, it is the reproductive advantage facilitated by a compressed component that determines its survival as a module. Like Rosca's approach, a hierarchy of modules is facilitated by 'atomisation' of modules.

More recently, the incorporation of module acquisition into Miller's CGP [84] by Walker & Miller [123, 124], resulting in Embedded CGP (ECGP), also resulted in significant improvements in performance. Again, *compress* and *expand* operators determine modules, and other operators serve to mutate modules in situ; modules survive if the are associated with a fitness improvemment. Modules have a size cap, which is user-specified. Unlike Angeline & Pollack's approach, however, modules within modules are not facilitated, so no hierarchy of modules can form, though Walker & Miller state that such an extension is currently being investigated.

All of the above approaches incorporate modules in a manner that require some explicit prescription of modules or module defining or acquiring mechanism. This *can* have some drawbacks. For example, if the number or size or structure of modules must be specified, the optimal parameter settings cannot be known in advance. Moreover, what is an appropriate set of modules (or building blocks) at one stage of an EA run may not be optimal for a another stage, and altering existing, well-utilised, is likely to have a catastrophic affect on offspring[2]. Furthermore, module incorporation in an EA that requires some

[2]There is an assumption here that module variation is non-neutral, and will usually result in a negative change in fitness. The consequence will be little fitness-neutral variation within a population, and second-order selection for modular configurations that enhance evolvability will be stifled as a result.

explicit mechanism, such as a module library or methods to evaluate prospective modules, are computationally expensive and defy biological plausibility.

EBDDIN employs a notion of modularity which is based on subfunction reuse, and is similar in its effects in this respect to modularity in those EAs discussed above. As stated by Woodwood [126]:

> "The ability of a representation to include modularity does not add expressiveness, it simply makes the expression more efficient in terms of its size."

However, while explicit modularity may have the limitations and disadvantages discussed, this thesis argues that modularity in EBDDIN has an inherent characterisation that overcomes some of those limitations. That is, there are no parameters that restrict modularity, and there are no explicit module defining or acquiring mechanisms and the drawbacks associated with them. Modularity in EBDDIN is able to vary gradually as a result neutral mutations, which are capable of finding all of the redundant genotypes of a given function. By this, genotypes having differing configurations but same fitness can compete through second-order selection, and favourable configurations can accrete. While this thesis does not deny that other EAs can exhibit neutral variation which targets functional redundancy, the author is not aware of any other EAs that can be shown to have neutral mutations that target functional redundancy explicitly and to the same extent. Chapters 5 & 6 discuss modularity in EBDDIN further, and contrasts it with modularity in other EAs.

3.3 Landscapes, search spaces & local optima

To assist in his description of the problem of evolution under Mendelian genetics, Sewall Wright introduced what have become commonly known as *adaptive landscapes*, or *fitness landscapes* [127]. All possible gene combinations are laid out as if to form a contour map, hence the *landscape* metaphor. Gene combinations that are mutationally near to a given point on the landscape are located

more closely than those combinations that are mutationally distant. The fitness, or adaptive value, of the different gene combinations are reflected by the different heights. Wright perceived a hilly landscape with several peaks of differing heights, surrounded by gradients and valleys in between. Populations reside at and around the tops of peaks. Wright states:

> "The problem of evolution as I see it is that of a mechanism by which species may continually find its way from lower to higher peaks in such a field. In order that this may occur, there must be some trial and error mechanism on a grand scale by which the species may explore the region surrounding the small portion of the the field which it occupies. To evolve, the species must not be under the strict control of natural selection. Is there such a trial and error mechanism?"

In EC, a slightly different perspective is often taken on navigating the landscape. Rather than moving from lower to higher peaks, the problem of evolution is seen as having a widely dispersed initial population gravitate to a global rather than local optima. The idea of moving from a lower peak to a higher peak is still valid, but often considered much less feasible.

The terrain of the landscape is important to facilitate efficient search. It is the combination of fitness function, representation and variation operators that determine the shape of the landscape. A smoother landscape with fewer peaks is considered to better facilitate evolution than a more rugged landscape.

3.3.1 The plausibility of Wright's landscapes

Wright's landscapes have had a profound influence in both evolutionary biology and EC. In his comprehensive book on the contributions of Sewall Wright to evolutionary biology, Provine [91] states that allegiance to the concept is intense because of its great heuristic value in graphically conveying the relationship between organisms, mechanisms of evolution, and adaptation. In EC similarly,

Wrights landscapes serve as the foundation for hypothesis pertaining to mechanisms of artificial evolution and overcoming the problem of local optima.

However, Provine [91, ch. 9] also argues that despite the great appeal of the concept, it is one of Wright's most confusing and misunderstood contributions. Firstly, the landscapes are depicted as a continuous surfaces with gradations. Yet, the axes, as originally conveyed, represent gene combinations and are discrete, so no continuous surface can be formed. Secondly, the landscapes are sometimes depicted as gene combinations, sometimes as gene frequencies, and sometimes as phenotypic characters; each of which has its problems. Thirdly, Provine questions the heuristic value of attempting to conceive of what are high-dimensional spaces in this way, as they become unintelligible. In EC, where allegiance to Wright's landscape concept is just as intense, Jones [57] has also questioned the plausibility of the concept.

One of the contributions of this thesis is to demonstrate that not all search spaces[3] are of the type conceivable as multi-modal landscapes. The search spaces considered in this thesis can all be shown to be free of local optima (see section 5.4), and have a very different and more biologically faithful search characterisation under the mechanisms of evolution proposed. The thesis will, however, concur with Wrights assertion that evolution is not under the strict control of natural selection, but argue that the 'grand trial and error mechanism' is, in fact, neutral evolution.

3.3.2 Population diversity

Population diversity is seen as playing a crucial role in evolution on a multi-modal landscape. The initial population is often seen as starting points in a parallel adaptive search process [58]. The randomly generated initial population, if large enough and exhibiting sufficient diversity, will cover the landscape sufficiently so that most of the peaks will have individuals within their basin

[3]The term *search space* will be preferred over the term *landscape*, though they will be used interchangeably in this thesis.

of attraction. If that diversity can be maintained, the population will move up the gradients of many peaks simultaneously, leaving the lower peaks behind and eventually converging on the global optima.

The aforementioned ideal, more often than not, does not meet with reality in most real-world cases. There are two primary reasons for this. Firstly, the landscape is often so rugged, having many peaks, that an impractically large population will be required to ensure that every peak has an individual within its basin of attraction. Secondly, maintaining diversity across different hills on the landscape is extremely difficult, even in moderately rugged landscapes. Furthermore, landscapes may be deceptive, leading the search towards local rather than global optima. In a detailed study of the role of the population in tree GP [62], Gustafson [48] found that the diversity could provide could facilitate parallel search, but required a mechanism to explicitly maintain portions of the population in different areas of the space.

In search spaces that are free of local optima by way of neutrality, population diversity is thought to play a different role. A contribution of this thesis is to show that population diversity is beneficial in such a space, and facilitates the evolution of evolvability (chapter 9). Barnett [10] had previously argued that population diversity is not beneficial in such a space, and this thesis demonstrates that this is not necessarily so. The reason why Barnett came to a different conclusion is beyond the scope of this thesis, but differences in the representation and the nature of the neutral networks may be relevant here.

3.3.3 Mutation rate

If a population converges to a local optima peak, it can become stuck there. Continued search is then considered pointless. The only chance to escape from the local optima peak is to produce a mutation large enough to jump out of a local optima into the basin of attraction of a higher peak. However, it is not sufficient to simply arrive within the basin of attraction of a higher peak; it must arrive high enough up that higher peak to compete effectively for selection

against the rest of the population that may be residing toward the top of a lower peak. Clearly, this becomes increasingly difficult as the search matures.

Small mutations are required to exploit and climb a resident peak, but larger mutations are required to explore the landscape and escape a local optima. How big a mutation is sufficient to escape a local optima, and in which direction, and how often? The answers to these questions cannot be known in advance, and will depend on the landscape and current configuration. For these reasons, determining the optimal mutation rate is a very difficult problem for successfully navigating multi-modal landscapes. While a high rate of mutation will always allow the possibility of escaping a local optima, it also neglects heredity, and brings the search closer to approximating random search.

Knowles & Watson debate the relative potential of mutation rate and neutrality for exploration, and argue that the former is preferential. A contribution of this thesis is to argue that neutrality is the superior source of exploration over a higher mutation rate (see chapter 7).

Crossover

Some EAs use the *crossover* variation operator, and this is also considered as a mechanism to escape local optima in those systems. This thesis does not investigate crossover directly. Claims that crossover can escape local optima typically consider the modality of the search space independently of the variation operators (see Jones [57] for a discussion). For example, some researchers contrive a troublesome landscape for a hill-climber simply for the purpose of showing how a GA crossover, or such, is able to do better. This thesis takes the view that such approaches have limited value, and that the modality of the search space cannot be considered independently of the variation operators. For this purpose, section 7.3 introduces a simple search space model based on accessibility, which is *not* independent of the variation operators applied.

3.4 Bloat

Bloat is the growth in size of candidate solutions beyond what is necessary, and is generally considered to inhibit evolution; see [111] for and overview. This means that, as the evolutionary run progresses, the population increases in the amounts of redundant code, code that is inefficient, or code that has a disproportionate impact on fitness compared to its size. Bloat is normally associated with tree GP [62] but, as highlighted by Luke & Panait [76], is common among a variety of variable length representations. Incidentally, in CGP, bloat is argued not to occur [83, 80]. CGP employs a fixed length genotypic representation and only the phenotype is allowed to vary in size. It is argued that the search favours a high percentage of inactive genes as the search matures, as this protects the active genes from perturbation. CGP is discussed further later in this chapter in the section on Neutrality in EC. Here, the discussion is limited to bloat in tree GP.

As an example of redundant code in GP, consider the following:

$$(a \wedge b) \vee (b \wedge a)$$

Clearly, the above subtree could be replaced with just $a \wedge b$, making the rest of the code redundant. In the BDD representation, an example of bloat would be an OBDD that is unreduced, or an ROBDD that does not have the optimal variable ordering to minimise its complexity. Many different types of bloat are reported in the literature (see [108] for a brief overview), but these are typically representation dependent, so are not pertinent to the discussion here.

3.4.1 Why does bloat occur?

There are several theories of why bloat occurs, some of which are discussed below. For a more detailed overview see [7, 75, 111].

Hitchhiking

Tackett [115] suggests bloat occurs when redundant code is located in close proximity to highly fit code. As the highly fit code propagates, the redundant code hitches a ride, and thus propagates also.

Defence against crossover

This theory claims that bloat results in order to resist the destructive effects of crossover in GP [7, 13, 78]. The crossover variation operator is know to have primarily destructive effects on progeny [7, ch. 6]. However, the greater the proportion of redundant code contained in a parent, the more likely the chosen crossover site will not impact on the fitness of offspring. This becomes increasingly significant as the run progresses, and improved fitness becomes increasingly harder to achieve. Results predict that redundant code will grow exponentially if unchecked.

Solution distribution

Langdon & Poli's [68] principal argument revolves around the observation that variable length representations facilitate a large number of functionally equivalent solutions. The distribution of these functionally equivalent solutions is biased towards longer solutions, so these are more likely to be found when genetic operators are applied.

Removal bias

Soule [112] suggests that bloat accumulates as a result of a bias in the nature of the crossover operator. It is similar, in principle, to the defence against crossover theory described above. In order for candidates to survive crossover, the crossover point typically has to be located towards the leaves of tree, which is more likely to retain redundant code. At this location, the subtree removed is likely to be relatively small. However, because the crossover point is redundant, the attached subtree can be any size, without consequence to function. Thus,

the tendency of surviving progeny is to remove smaller subtrees and attach larger ones, so the larger candidates propagate.

3.4.2 Methods to combat bloat

Size restriction

This method puts a cap on the maximum size or depth of offspring. Offspring that exceed this are not allowed to survive and breed. The first, and most obvious, problem with this approach is that the cap may exclude the optimal solution from the search space. However, Gathercole & Ross [44] report that such a cap leads also to a reduction in diversity, stifling evolvability.

Parsimony pressure

This method seeks to discourage larger offspring from breeding. Luke [76] identifies two primary types: parametric and nonparametric. *Parametric* parsimony pressure penalises larger individuals in the fitness function. While this can often improve parsimony, it can also lead to degradation in diversity and overall performance [88, 112]. *Nonparametric* methods typically rely on contrived selection methods to encourage parsimony. While some success has been reported it is limited, and none offer a universal remedy; see [39, 76].

Operator modification and code editing

Soule [111, p. 19] cites various approaches to modifying genetic operators. Though genetic redundancy is thought to be important for the mechanism of natural evolution, he states that these methods have "the goal of reducing the evolutionary importance of inviable code.".

Regarding the editing out of redundant code, Soule [111] argues that the difficulties inherent in the process offer little potential. However, in the OBDD representation, removing redundant code is a trivial matter, and one which can be exploited in EBDDIN: section 5 discusses this further. A further source

of bloat in EBDDIN is the variable ordering, and this is addressed in chapter 8, demonstrates that variable orderings that minimise solution complexity can emerge under normal evolutionary forces.

3.5 Neutrality in EC

There has been considerable work recognising the potential of neutrality using both RNA models [52, 53, 105] and other artificial representations [8, 37, 50, 85, 116, 119]. These works typically highlight the exploratory potential of neutrality, highlighting such properties as stagnation avoidance whilst maintaining phenotype, and robustness to both high mutation rates and the starting configuration. However, other works offer more cautionary and sceptical opinions on neutrality [20, 97, 61, 110]. Thus, there remains considerable debate on the usefulness of the concept of neutrality within the EC community.

The objective is to exploit the notion of intraspecific variability to help the search process. Typically, a redundant genotype-phenotype, or genotype-fitness, mapping is employed in which many genotypes map to the same phenotype or fitness value. *Neutral networks* are formed through which a so-called *neutral walk* can progress. A neutral network is an area of the search space in which fitness-neutral genotypes are connected to each other by single applications of the genetic operator(s). There may be one large, or many smaller, neutral networks for each fitness value, or phenotype. The set of all networks will be interconnected to some degree. That is, regions of a given network will facilitate jumping onto another network with a single application of the genetic operator; these may be envisaged as borders between networks. The aim of a neutral walk, then, is to move along a neutral network until a border is reached with a network of genotypes that map to a higher fitness phenotype, then jump across.

There are properties of the neutral search spaces that determine the likelihood of successful search. Large and highly interconnected networks are more likely to have borders with higher fitness networks than small and poorly con-

nected ones. Such spaces may alleviate local optima by facilitating a maintained rate of innovation, as discussed by Huynen [52] in his study of RNA spaces, and also recognised by Barnett in his study of NKp spaces [9]. Part of the contributions of this thesis is show that neutrality completely alleviates local optima for a significant problem domain under EBDDIN (see chapter 5).

3.5.1 Genotypic redundancy

A necessity for neutrality is genotypic redundancy. Rothlauf & Goldberg [97] highlight some of the issues regarding types of redundancy in their argument for a distinction between what the term *synonymously* and *non-synonymously* redundant representations. They state that a representation is synonymously redundant if all the genotypes that represent the same phenotype are similar to each other, homogenous genotypes being isolated from other phenotypes. If not, the representation is non-synonymously redundant, but this does not allow the variation operators to work properly, lowering the performance of evolutionary search. They state:

> "When using synonymously redundant representations a mutation results in either the same or a similar phenotype. Contrastly, when using non-synonymously redundant representations the mutation of a genotype results in completely different phenotypes."

Figure 3.1 shows their distinction pictorially, and they also provide a formal characterisation on the sum of the distances between genotypes that represent the same phenotype:

$$\left(\sum_{p} \frac{1}{2} \left(\sum_{g \in G^p} \sum_{g' \in G^p | g \neq g'} distance(g, g') \right) \right), \tag{3.1}$$

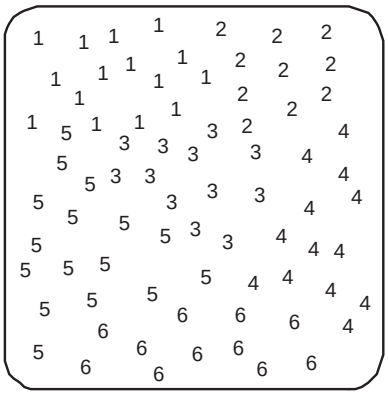
where G is the set of all genotypes, P is the set of all phenotypes, and $G^p \subseteq G$ is the set of all genotypes representing the same $p \in P$. The *distance* between two genotypes may be determined by a method that appropriately captures

the notion of how mutationally distant they are, e.g. hamming distance. Synonymously redundant representations will exhibit small values of (3.1) for all phenotypes, and non-synonymously redundant representations exhibit comparatively larger values. Rothlauf & Goldberg argue that non-synonymously redundant representations result in randomised search and poor EA performance, and they consider that only synonymously redundant representations may facilitate effective evolutionary search. The synonymously redundant representation is of the type perceived by Lenski, and discussed in section 3.1.3 regarding robustness and evolvability. That is, Lenski [73] suggests neutral mutations are unimportant to evolvability because they are phenotypically isolated. Rothlauf & Goldberg cite Knowles & Watson [61] as showing that performance on the non-synonymously redundant Random Boolean Network representation decreases with increased redundancy in support of their argument. This thesis will argue that Rothlauf & Goldberg's [97] classification is inadequate to describe the properties of desirable redundancy that facilitates neutral networks, and a third classification will be introduced in chapter 5.

3.5.2 Types of neutrality

Geard et al. [45] compare three types of encoding; NK, NKp and NKq landscapes. They conclude that the *type* of neutrality provided by a system is important to the search performance. This section reviews some of the types of neutrality that have been identified and clarifies the type of neutrality that EBDDIN exploits.

Harvey & Thomspson [50] argue that not all neutrality can be helpful to evolutionary search. They highlight that redundant loci added to an encoding will form neutral networks, but if those loci can never, under any circumstances, be included in the determination of the phenotype, nothing can be gained. They call this *useless junk*. However, redundant loci that have the potential to be used in the determination of the phenotype, perhaps by being activated by some mutation on another part of the genotype, become *potentially useful*



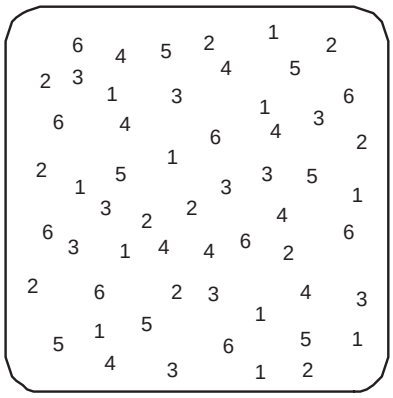


Figure 3.1: Rothlauf & Goldberg's depiction of synonymously vs non-synonymously redundant representations (redrawn). Each integer instance represents a genotype. Different integer values represent different phenotypes (not different fitness values). A synonymously redundant representation sees those genotypes that represent the same phenotype grouped into 'globules', being similar and relatively easily accessible to each other via the variation operators. In contrast, a non-synonymously redundant representation sees the genotypes that represent the same phenotype widely dispersed, and not easily accessible to each other via the variation operators.

junk. Harvey & Thomspson investigate this kind of neutrality with FPGA on a tone recognition problem. In their system, they argue that evolution can, and mostly does, take place in a converged population (i.e. genotypes in the population are very similar to each other). This implies that small populations may be perfectly acceptable in this kind of search, as opposed to the very large populations typically used in GP that do not attempt to exploit neutrality. They suggest that neutral networks percolate through genotype space eliminating local optima, but provide no proof of this claim.

Ebner et al. [37], compare three types of encoding one with no redundancy and the other two with differing degrees of redundancy. They stated that there is little bias in the solution space, the distribution of phenotypes being relatively even. They argue that the neutral networks that result are highly intertwined, and report increased accessibility between phenotypes as a result. Increased redundancy increases this accessibility through neutral walk, and generally improves evolvability. This also implies that local optima may be alleviated by neutrality, because increased accessibility between phenotypes from neutral walk implies a greater potential to find an improving phenotype.

Jakobi [54], however, is able to show how neutrality can completely alleviate local optima, but in a manner which is very limited for practical purposes. The method works by introducing redundant loci. The original encoding having n loci is transformed into one have $2n + 1$. The first n loci encode solutions as before, as do the second n loci, and the final loci determines whether the first or second set of loci is active. Assuming the first set of loci is active according to the deciding bit, but stuck in a local optima, the second set of loci will be free to vary without hinderance. Given enough time, the second set of loci will happen upon an encoding having higher fitness than the first set of loci by what is effectively random search, at which point mutation of the deciding bit can activate the second set of loci and increase fitness of the genotype as a whole. While this example highlights the exploratory potential of neutrality to eliminate local optima, it relies heavily on a random search component and

would generally require what Jakobi describes as a *monkey's-typing-Shakespeare* amount of time.

Yu & Miller [130] make a distinction between what they term *implicit* and *explicit* neutrality. Implicit neutrality takes two forms: *functional redundancy* and *introns*. Functional redundancy refers to many ways of representing the same function. For example, the following two expressions represent the same function (by De Morgan's law) but are expressed differently:

$$\neg(a \wedge b) \Leftrightarrow (\neg a) \vee (\neg b)$$

Introns are parts of the genotype that are semantically redundant to the calculation of the function. Yu & Miller state that functional redundancy and introns can emerge during evolution, but they are not easy to identify or control. This is also supported by the discussion of bloat in section 3.4, in which Soule [111] acknowledged the difficulties in removing redundancy from candidate solutions in GP.

Thus, Yu & Miller propose using *explicit neutrality*. In this approach, some parts of the genotype are *active* while other parts are *inactive*. Mutation on inactive parts of the genotype transform one genotype into another and have a neutral effect. Mutation on active parts of the genotype, however, is non-neutral and, furthermore, can activate inactive parts of the genotype. Both CGP [84] and Galvan-Lopez's [100] variant of Koza's GP exploit this type of neutrality. However, it is also the type of neutrality employed in Jakobi's trivial example of how neutrality can completely alleviate local optima. The distinction between Jakobi's explicit neutrality and that employed in CGP and Galvan-Lopez's GP, however, is that Jakobi activates redundant loci which encodes the whole solution; Yu & Miller and Galvan-Lopez activate loci which represents only part of the solution.

While explicit neutrality is seen as more manageable than the implicit form, it is not to say that implicit neutrality does not occur in CGP and GP. Yu

& Miller [132] argue that the combination of implicit and explicit neutrality is important in CGP, stating that the ratio of active to inactive gene changes during neutral walk is important and self-regulating there. However, again, it is acknowledged that implicit neutrality is difficult to identify and control in both CGP and GP.

Explicitly neutral mutation on functional parts of the genotype

The type of neutrality investigated in this thesis is explicitly neutral mutations on *functional* parts of the genotype only: there are no introns in EBDDIN and no inactive parts of the genotype.[4] This is the type of redundancy which falls into the category of *implicit* neutrality according to Yu & Miller, and which is acknowledged to be difficult to identify and control in other representations. Under EBDDIN, however, this is not so, and identifying functional redundancy and targeting it with explicitly neutral mutations is a simple matter.

This thesis will argue that the neutral networks that result in EBDDIN are large and highly intertwined, and show that local optima are completely alleviated by neutrality for the class of Boolean functions (chapter 5). Unlike Jakobi's trivial example of how neutrality can alleviate local optima using inactive genes, the neutral mutations in EBDDIN on active parts of the genotype vary both modularity and pleiotropy in the genotype in a gradual manner. These properties then become subject to second-order selection for evolvability, and favourable configurations are able to accrete; this will be discussed throughout the thesis.

3.6 Summary

This chapter has reviewed some of the main issues in EC. The issue of evolvability and its emergence was discussed. In particular, an attempt was made

[4] Thus, *explicitly neutral mutations on functional parts of the genotype* are distinct from Yu & Miller's [130] *explicit neutrality*, the latter of which acts on inactive, non-functional parts of the genotype.

to disambiguate the concept of evolvability by identifying a variation component and a selection component that both contribute towards evolvability of the population as a whole. Notions that selection for evolvability requires a dynamic fitness function were also discussed and challenged. Modularity in EC was discussed. The role of population diversity and mutation rate were discussed also, and the relevance to mechanisms of evolution on multi-modal landscapes was identified. It was suggested that, although Wright's landscape concept was generally considered an essential tool for hypothesising mechanisms of evolution, it is not without its critics. It was further pointed out that neutrality, although controversial, has been argued to alleviate the problems of local optima, suggesting a very different structure of search spaces than that envisaged by Wright and adopted by the majority of evolutionists. The problems of bloat were also discussed, and the many facets of the problem and proposed solutions gave testimony to the significance of the problem. Types of neutrality were also discussed, and it was highlighted that this thesis employs explicitly neutral mutations acting on a completely functional genotype, where there are no introns or inactive parts of the genotype.

Chapter 4

Critical review of previous approaches to evolving BDDs

The BDD data structure is a popular choice in applications such VLSI design and verification. Coupled with the significance given to the choice of genotypic representation in EC, this makes BDDs an interesting proposition for the choice of genotypic representation in EC where candidate solutions can be suitably encoded. This chapter provides a critical review of some the previous attempts at exploiting BDDs as the genotypic representation in EC for function optimisation. The emphasis is on those approaches which attempt to automatically synthesise a function in the OBDD representation given some specification and are, thus, directly comparable to the approach developed in this thesis.

EC is often applied to BDDs for BDD minimisation, i.e. finding good variable orderings rather than synthesis of correct functions in the BDD representation. Rolf Dreschler and his group at the University of Bremen in Germany are prominent in this area [95, 99]. Here, EC is used to evolve a good variable ordering only [18, 21, 33, 71]. While there are many applications for the BDD

minimisation problem, particularly in VLSI design and verification, the focus of this review is EC applied to the synthesis of functions in the BDD representation using EC techniques. All except Droste [35] and Kühne & Drechsler [66] are directly comparable to the system developed in this thesis in their objectives. The characteristics of the systems are discussed here critically. A performance comparison on a number of benchmarks is given in chapter 5 against EBDDIN for those approaches that are directly comparable and have published data.

4.1 Droste

Droste et al. [35] presented a distributed hybrid approach to evolving BDDs. Droste's objective is to find a compact and *generalising* OBDD for the 20-bit multiplexer using a partial specification. It claims to be the first BDD-based GP system that facilitates dynamic variable reordering.

Each subpopulation has a specified variable ordering, π, allowing all the individuals to be stored as a multi-rooted ROBDD, saving memory. Migrating individuals have their π changed when entering the destination population. Periodically, each subpopulation has its π subjected to a π heuristic optimisation algorithm, the group sifting algorithm of Panda & Somenzi [90] or other algorithm. Variation operators consist primarily of recombination and mutation with low probability.

Individuals are constrained to be consistent with the training set at initialisation. Offspring not meeting this constraint are replaced by one of the parents. The system was able to find orderings better than random, though they were far less than optimal. Generalisation was poor.

Droste argues that this is the only BDD-based GP system that evolves the variable ordering. However, it is clearly a hybrid system, and does not appear to evolve the variable ordering at all. To evolve, there must be variation and heritable selection of the variable ordering, yet there is none. Each subpopulation simply applies the heuristic globally and periodically, and independently of

other subpopulations. It is only the heuristic that affects the variable ordering because migrating individuals adopt the ordering of the destination subpopulation. Effectively, this is a static variable ordering GP system, periodically applying a heuristic, but distributed. Any other system could be modified similarly.

4.2 Sakanashi

The approach of Sakanashi et al. [103] to evolving BDDs utilises an extended form of Koza's GP [62]. Their objective is to synthesise a BDD according to a given specification. (detailed results are given for comparison in chapter 5. The search space is restricted to unordered BDTs, which facilitates the application of the conventional GP variation operators. This representation was chosen because the authors had no techniques to evolve graphs[1]. The fitness function rewards functional correctness and penalises unorderedness, the balance of which is controlled by a parameter, k. The variable ordering is evolved, but without respect to the eventual compactness of the resulting ROBDD.

Redundancy is removed from selected, highly fit BDTs, with a view to improving evolutionary performance. Redundancy takes two forms in BDTs. First, when the two children of a given nonterminal are identical. Second, when a given vertex is unreachable due to a repeat occurrence of that nonterminal variable label along any given path. In the former case, the nonterminal is replaced by the child. In the latter case, the repeated nonterminal is replaced by its reachable child. These reductions are applied to fully correct BDTs and BDTs with high relative overall fitness.

Some success is reported with their technique, but it failed on some benchmark problems and performed poorly in comparison to other approaches on other problems (see section 5.7). They suggest problems could be overcome with adaptive mechanisms for the k parameter and more selective redundancy

[1] Personal communication 20 January 2005

removal, though the work was not pursued.

Sakanashi's approach to evolving BDDs exhibits numerous shortcomings in my opinion, and the BDT representation employed evades exploiting many of the potential advantages of true BDDs. Firstly, from a size and performance perspective, BDTs are typically much larger than their functionally equivalent ROBDD counterparts, leading to increased space and time costs. Furthermore, BDDs can be constructed by first constructing a complete, ordered BDT, then applying the reduction mechanisms. Such BDTs generally contain a lot of redundancy, and could easily be as large as its truth table. Therefore, it appears at least as easy to generate the ordered BDT from the truth table, reducing it to the required ROBDD as required.

Secondly, consider modularity. A BDT may contain a large amount of functional redundancy. This redundancy has the potential for modularisation, and are what a ROBDD exploits to facilitate compactness. Thus, the BDT restricts modularity, while the reduced and partially reduced OBDD exploits it. Modularity is believed to have an important role in evolvability, and is often employed in GP systems (section 3.2).

Thirdly, genetic variation and uniqueness of individuals are important properties of adaptive evolution [23]. Sakanshi's approach, however, eliminates much potential genetic variation, not only by failing to employ true BDDs, but also by reducing BDTs at the earliest opportunity. The resulting reduction in genetic variation is likely to be detrimental to the search performance.

Finally, variable ordering is considered. The search space utilised by Sakanshi approach incorporates BDTs that are not ordered. These genotypes cannot be reduced to canonical form, so the search space is much larger than is necessary. In addition, the approach does not specify a variable ordering explicitly, but seeks to evolve one using a penalty in the fitness function. While it is true that an appropriate ordering is essential for producing a compact ROBDD, this approach and its fitness function pay no attention to the consequence of the variable ordering on eventual compactness. Therefore, the evolution of the

variable ordering appears wasteful and unnecessary.

4.3 Yanagiya

Yanagiya's *BDD-based GP* [128] stores the entire population in a single, multi-rooted BDD, minimising storage requirements. Again, their objective is to synthesise a BDD according to a given specification. This approach also accelerates fitness evaluation by employing logic operations on the BDDs of candidate and target functions. Logic operations are speeded by employing a hash-based cache that stores the results of recent operations. Good results are demonstrated, solving the 20-bit multiplexer problem. Yanagiya suggests that memory requirements are a limiting factor.

Yanagiya's approach is interesting in its ability to save space by storing the population in a single multi-rooted ROBDD. However, it is fundamentally flawed in that this negates the potential of individuals to possess differing π, and further restricts genotypic diversity by sharing genotypic material between individuals. Yanagiya's approach to fast fitness evaluation using BDD logic operation also requires the BDD representation of the target function in order to evaluate solutions: it requires the solution to the problem being solved, limiting the practicality of the approach. Yanagiya states that large memory requirements are also an inhibiting factor.

4.4 Van Remortel

Van Remortel [118] experimented with the effect of *cube transformations* [12] on the size of a function's ROBDD. Their objective is to synthesise an ROBDD according to a given specification. Cube transformations effectively map one function to another, and can be seen a row-wise permutation of the truth table. The goal of the cube transformation is to map a function with a complex ROBDD representation into a function that has a simpler ROBDD representa-

tion.

Van Remortel's states that his motivation for using cube transformations is to by-pass the canonicity property of ROBDDs. Because Van Remortel's method searches for ROBDD's, which are unique, cube transformations permit alternative representations of the same function, increasing the frequency of the target in the search space. Results were inconclusive. Fixed variable ordering appeared to give the best results in terms of speed, but the potential of cube transformations to reduce the complexity of otherwise large ROBDDs was also recognised.

Van Remortel's results were preliminary, and did not demonstrate any improvement over the other approaches. Claims that cube transformations could improve performance do not appear to have been investigated further.

One of the problems with cube transformation is that the transformation has to be stored as well as the ROBDD, which can be prohibitive. Given that Van Remortel's objective in using cube transformations was to increase the frequency of the target in the search space, expanding the search space to include OBDDs may have been more effective here. This would have not only increased the frequency of the target, but also avoided the need to store that transformation.

4.5 Kühne and Drechsler

Kühne & Drechsler [66] takes an approach to evolving generalised BDDs (GBDD) [15] rather than the most common form of BDD that is OBDD. GBDDs do not have the ordering restriction on the variables and can have duplicate variable labels along paths. Their objective is to synthesise a function that is both *correct* and *compact*, and this is built into the fitness function with the correctness property taking priority. The motivation for using GBDD rather than OBDD is that functions that have no compact ROBDD representation can sometimes have compact GBDD representations: they wish to investigate the potential for exploiting this using EC.

They define a number of variation operators, some of which are function-preserving (i.e. functionally neutral). This has some similarities to the technique presented in this thesis, though it should be noted that the approach presented in this thesis predates that of Kühne & Drechsler and was first published in [27]. They report that their approach is sometimes able to find slightly more compact GBDDs than ROBDDs on a number of benchmarks, but only on small functions, i.e. up to about 40 nodes. For larger functions, they state that the algorithm takes too long to find a *correct* solution when the population is initialised randomly. They state that this can be avoided if the population is initialised to correct OBDDs, but this induces local optima in terms of the compactness property.

Kühne & Drechsler's findings suggest that this approach fails to scale to larger problems, and the work does not appear to have been pursued. The approach is interesting in that it too, like EBDDIN, attempts to exploit function-preserving mutations to introduce diversity. However, function-preserving mutations are *not* necessarily neutral under this approach because the fitness function depends on both functional correctness *and* compactness. The compactness component of the fitness function actually reduces neutrality and induces local optima, therefore, and this will actually reduce diversity in the population. The neutral networks present in this space are likely to be small and disconnected, limiting the potential for neutral walk. This perhaps explains their observations on limitations of performance and scalability.

4.6 Summary

This chapter has critically reviewed other approaches to synthesizing functions in the BDD representation using EC. None of these systems have gained popularity. Performance and scalability issues have been prominent. The EBDDIN approach developed in this thesis will attempt to overcome some of the limitations of these other systems. In particular, it will exploit the redundancy

of the OBDDs to create large and percolating neutral networks to facilitate exploration without loss of fitness (section 5.3). Modularity (i.e. reuse of sub-function) will also be exploited extensively. Seamless, gradual evolution of the degree of modularity is coupled to variation in pleiotropy. An emergent bias in pleiotropic effects (chapter 6) aids in directing phenotypic variation in a manner not inconsistent with Kirschner & Gerhart's theory of *facilitated variation* [60] or Hansen's [49] suggestion that it is evolved pleiotropic effects that facilitate evolvability and its evolution. Comparisons of performance are given in chapter 5 of EBDDIN against the above approaches, and the implications for understanding and characterising evolvability are discussed throughout the thesis.

Chapter 5

Evolving BDDs using inherent neutrality

This chapter introduces a new approach to evolving BDDs. The neutral networks that result are both internally and externally highly interconnected. The search space is shown to be free of local optima by way of neutrality for a significant class of fitness functions. Explicitly neutral mutations allow a significant proportion of fitness evaluations to be circumvented. BDD logical operations reduce the time cost of each fitness evaluation by orders of magnitude. Compared to other approaches to evolving BDDs, this new approach requires much fewer fitness evaluations to find a solution on a number of benchmarks. The result is a new EA that will be employed in this thesis as a computational model of an evolutionary system for studying important evolutionary concepts.

5.1 Introduction

Typically, when the designer of an artificial genotypic representation wants to incorporate neutrality, it is introduced by way of redundancy via a many-to-one mapping. However, this approach to introducing neutrality can be problematic because the nature of the redundancy that results is unpredictable. To facilitate

the exploration that accompanies neutral walk, the redundant neutral variants of a given function must be highly interconnected through the variation operators. If not, a large number of small and disconnected neutral networks will result, each consisting of one or few genotypes, stifling the potential for neutral walk.

These issues were introduced in section 3.5, where Rothlauf & Goldberg's [97] two-category distinction of the nature of redundancy was discussed. However, categorising redundant representations into only one of these two possibilities is misleading. Proponents of neutrality convey a very different characterisation of highly neutral spaces that differ from both of those presented by Rothlauf & Goldberg: for example, see Ebner et al. [37]. Rather than the *globules* in genotype space depicted by the synonymously redundant representation, or the disconnected genotypes of the non-synonymously redundant representation, phenotypically homogenous genotypes form fine, highly interconnected structures. The structure of such networks is difficult to convey pictorially in two dimensions. However, one can imagine that each genotype is surrounded by both neutral variants and non-neutral genotypes, forming fine-stranded structures rather than the globules of Rothlauf & Goldberg's synonymously redundant representation which Lenski [73] argues makes neutral mutations unimportant to evolvability.

The EA introduced in this chapter exploits the neutrality inherent in the OBDD representation, revealing much about the nature of the neutral networks that result. Redundancy in the OBDD representation is evident by the fact that all OBDDs are reducible to an ROBDD representation. Thus, a function with a compact ROBDD representation has a plethora of redundant representations lying somewhere between its ROBDD representation, which may have size linear in the number of variables, and the tree representation, which is exponential in size. Furthermore, it is clear that all the redundant variants of a given function are connected via the reduction mechanisms described earlier in section 2.3, making it trivial to navigate between all of them via neutral walk. Thus, the approach presented in this chapter is referred to as *Evolving Binary Decision*

Diagrams using Inherent Neutrality[1] (EBDDIN):

> *The characterising feature of EBDDIN is the use of explicitly defined neutral mutations on BDDs that, when applied on their own, are capable of exploring many (or all) redundant variants of a given function through neutral walk, and this will be true for all functions.*

In section 5.2 the variation operators are introduced and the effect on the search space discussed in section 5.3. Section 5.4 shows the search space to be free of local optima for a significant class of fitness functions. Methods to reduce the cost of fitness evaluation are discussed in sections 5.5 and 5.8, and bloat is addressed in section 5.6. The EBDDIN approach is compared to other approaches in section 5.7. Section 5.9 promotes EBDDIN as a model for an evolutionary system.

5.2 Approach

The *genotype* of an individual in EBDDIN is an OBDD and the *phenotype* is the corresponding ROBDD having the same variable ordering, which is easily mapped to by OBDD reduction. However, for most purposes it will be sufficient to consider the phenotype of an individual to be the Boolean function represented.

The following mutation operators are defined on the OBDD genotype, all except one of which is explicitly neutral (i.e. not functionally modifying). See figure 5.1.

Definition 1. *Let* **N1** *be the neutral mutation of removing a redundant test.*

Definition 2. *Let* **N1'** *be the neutral mutation of inserting a redundant test, the inverse of* **N1**.

[1] In some of the previous publications associated with this thesis, the word 'inherent' is substituted for 'implicit' (i.e. *implicit neutrality*), but is changed to avoid ambiguity with Yu & Miller's use of the term [130], which differs in meaning.

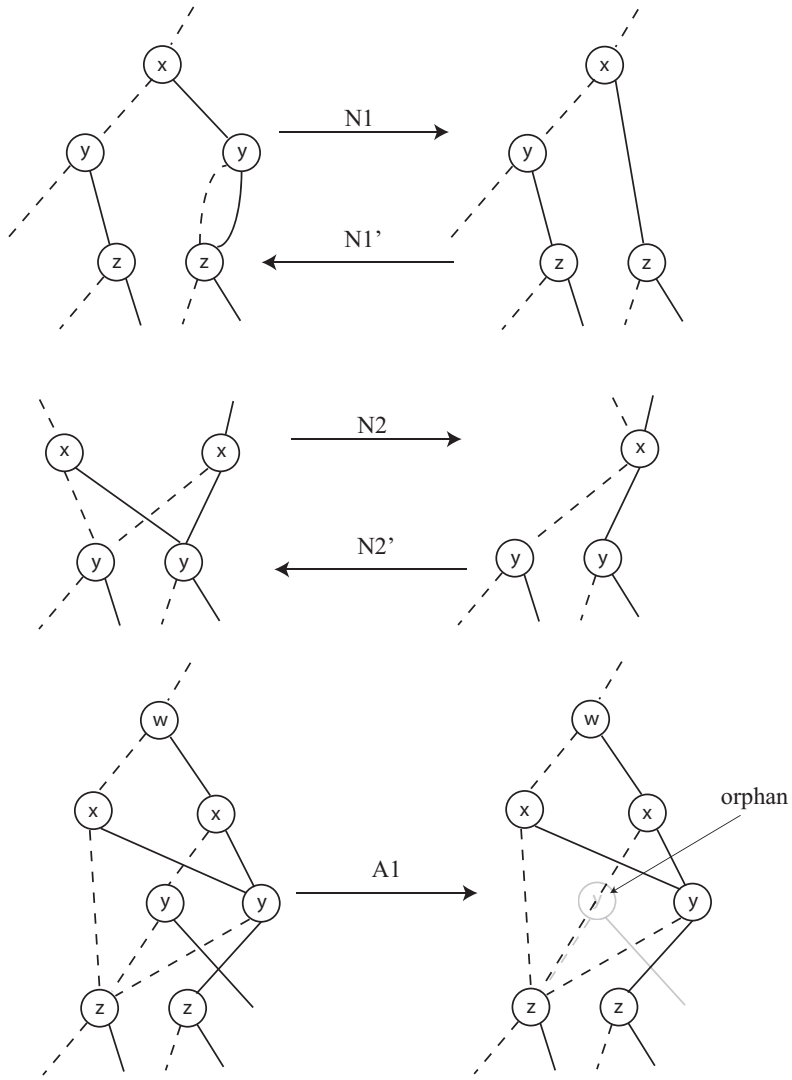


Figure 5.1: EBDDIN mutations.

Definition 3. *Let* **N2** *be the neutral mutation of removing a redundant non-terminal (merging two equivalent non-terminals), where two vertices have the same variable label and the same respective children. An* 'aggressive' *version of* N2 *chooses a random vertex, then merges* all *vertices with both the same variable label and same children.*

Definition 4. *Let* **N2'** *be the neutral mutation of inserting a redundant non-terminal, the inverse of* **N2** *(splitting a non-terminal).*

Definition 5. *Let* **A1** *be the 'potentially' adaptive (or functionally modifying) mutation of changing one of the children of a non-terminal, to another vertex, potentially orphaning a sub-graph.*

Definition 6. *Let the* **mutation set, M,** *consist of those mutation operators that are used in an evolutionary run.*

All of the above mutations are applied in a way that maintains the variable ordering, ensuring a valid OBDD is maintained as the genotype. A fifth neutral mutation, N3, that affects the variable ordering will be introduced in chapter 8. Before discussing the effect of these mutations together, the effect of each individually is discussed. Firstly, notions of *pleiotropy* and *modularity* are introduced, defined and relate for the OBDD representation.

In biology, pleiotropy occurs when a single gene influences multiple phenotypic traits. Consequently, a mutation to that gene can affect any or all of those traits simultaneously. For the purposes of the computational model presented in this thesis, pleiotropy is defined as:

Definition 7. Pleiotropy[2] *is when a genotypic feature (i.e. an edge or vertex) influences multiple phenotypic traits (fitness cases). It is quantified by counting the number of fitness cases processed by a given edge or vertex, or it may be given as a percentage of the* 2^n *fitness cases.*

[2]According Radcliffe & Surry's [92] formal categorisation of representations, the OBDD representation falls into the category of an *allelic* representation. An allelic representation does not necessarily have genes, but a collection of properties. Thus, the use of the term pleiotropy to describe the influence of edges and vertices on the phenotype is not an entirely accurate biological analogy, but suffices for the purposes of this thesis.

For example, consider an OBDD in tree form. The root vertex processes all fitness cases, so pleiotropy is maximum here. Each vertex encountered along a path divides pleiotropy between its children, so pleiotropy decreases logarithmical until the last edge where pleiotropy = 1. Reduction of an OBDD genotype, however, will result in an increase in pleiotropy as fewer genotypic features are involved in processing the same number of fitness cases. See figure 5.2.

There are many notions of modularity, and a more detailed discussion of the various notions is deferred until the following chapter. Woodward states of modularity [126]:

> "The ability of a representation to include modularity does not add expressiveness, it simply makes the expression more efficient in terms of its size."

For the purposes of thesis, we use a notion of modularity that is defined in terms of reuse and the removal of functional redundancy. Thus, the definition focusses on the effects of modularity rather than stating what is a module and what is not.

Definition 8. *An OBDD genotype* g *is more modular than genotype* h *if they have the same number of variable and the size of* g *is less than the size of* h.

Pleiotropy will typically be higher as a result of increased modularity because, from the above discussion and figure 5.2, it is clear that modularity and pleiotropy are positively correlated. A change in modularity as a result of a single neutral mutation is reflected at the level of the whole genotype, but the change in pleiotropy may be very localised: this will made explicitly clear in the following discussion. So, although modularity has been defined globally on the whole genotype, its effects can be localised.

While the pleiotropy of a genotypic feature has been quantified as a count of the number of fitness cases influenced, this says nothing about whether the feature's influence is predominantly good or bad. For this purpose, the notion of *pleiotropic utility*, or simply *utility*, is introduced:

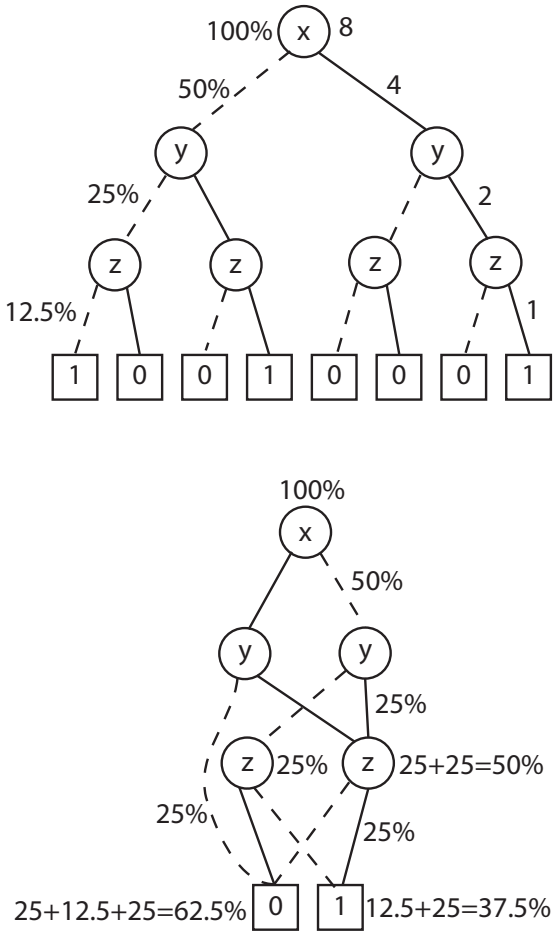


Figure 5.2: Pleiotropy in the OBDD representation. (Top) Pleiotropy on a BDT given as percentages (left) and absolute values (right). Pleiotropy decreases logographically as each vertex encountered along each path divides pleiotropy between its children. (Bottom) In a reduced or partially reduced OBDD, pleiotropy is generally higher and vertices may have more than one incoming each. Vertices have pleiotropy which is the sum of the pleiotropy of all incoming edges. Note that at the terminals pleiotropy sums to 100%.

Definition 9. Pleiotropic utility, (U_p), *is the ratio of correct fitness cases processed by a genotypic feature to pleiotropy:*

$$U_p = \frac{+cases}{pleiotropy} \tag{5.1}$$

Utility says nothing about the amount of good or bad influence of a genotypic feature. It says only what proportion of the total influence is good. If mutation perturbs a genotypic feature with relatively high utility, it can be expected that a greater drop in fitness will result than from the perturbation of a genotypic feature with low utility. An investigation into how utility evolves, and the implications, is presented in chapter 6.

With the above definitions in place, the effects of each of the mutations can now be considered regarding their effects on pleiotropy, modularity and utility.

N1 and N1' A redundant test, α, with variable, $var(\alpha)$, processes some set of fitness cases, A. A_0 and A_1 are subsets of A distinguished on $var(\alpha)$, each associated with a child edge of α; see figure 5.3. $A_0 \cup A_1 = A$ and $A_0 \cap A_1 = \emptyset$. The sets A_0 and A_1 behave as though their union in that their associated edges have the same destination, but each is free to vary independently under A1 mutation. Removal of the redundant test α by N1 removes the potential for A_0 and A_1 to vary independently by A1 mutation on $var(\alpha)$, protecting the collective behaviour of A.

A_0 and A_1 will have the same pleiotropy, but may well have differing utility. If both A_0 and A_1 have high utility, then there may be an evolutionary advantage to removing α and protecting all of A from perturbation at the variable $var(\alpha)$. If, on the other hand, one or both of A_0 and A_1 have low utility, they may present a good target for A1 mutation providing an evolutionary advantage to keeping or inserting α with N1'.

N2 and N2' Two vertices α and β, with $var(\alpha) = var(\beta)$ and the same two children, process two sets of fitness cases, A and B, respectively. As above,

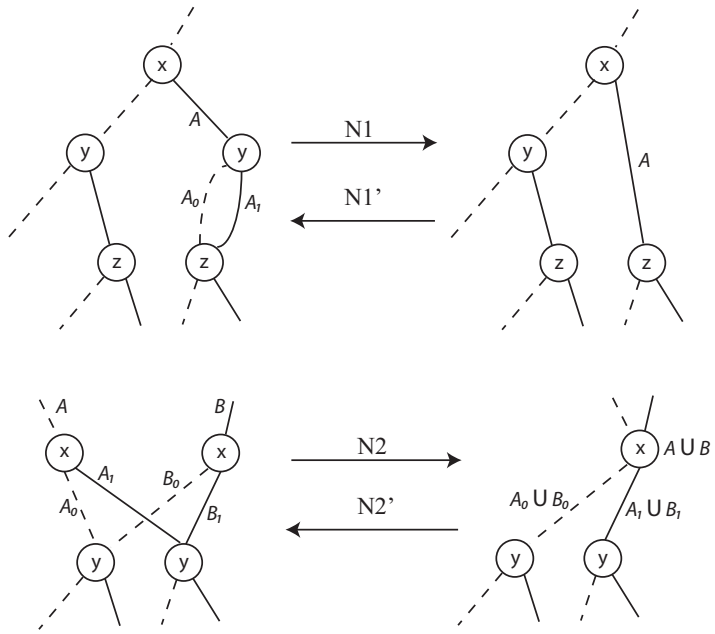


Figure 5.3: The effects of neutral mutation on variational dependencies between phenotypic traits (i.e. fitness cases). (Top) Removing a redundant test (N1) removes the potential of the subsets fitness cases A_0 and A_1 to vary independently on variable y from A1 mutation. Inserting a redundant test (N1') inserts the potential for A_0 and A_1 to vary independently on the variable y. (Bottom) Merging two nodes (N2) removes the potential for A_0, A_1, B_0 and B_1 to vary independently on variable x from A1, coupling variation on x to the sets $A_0 \cup B_0$ and $A_1 \cup B_1$. Conversely, splitting a vertex (N2') creates the potential for A_0, A_1, B_0 and B_1 to vary independently on variable x.

each vertex has subsets distinguished on the their respective variables and each subset is associated a child edge: see figure 5.3. The subsets A_0, A_1, B_0 and B_1 are free to vary independently by A1 mutation before application of N2. After application of N2, however, variational independence of the four sets is removed and two new sets are formed, $A_0 \cup B_0$ and $A_1 \cup B_1$. Again, whether the evolutionary advantage is greatest before or after N2 application will depend on the utility of the edges.

A1 Any edge under A1 mutation may result in a change in subfunction along some point in a path. Given that that some subset of fitness cases map onto that path, the mutation can be functionally modifying. It is also clear that some change in modularity, pleiotropy and utility may also result from A1 mutation, though these changes will often, but not always, be coupled to a negative change in fitness.

So, it is clear that the neutral mutations vary modularity, the amount of pleiotropy and pleiotropic utility. This results in changes to variational dependencies between fitness cases. Neutral genotypes have different configurations of these properties, and it can therefore be expected that different genotypes have a different expected average response to A1 mutation. Configurations that infer some evolutionary advantage represent the variation component of evolvability and are subject to second-order selection, and so can evolve: evolvability can evolve. This will be investigated in chapter 6, where it is shown that biased configurations of pleiotropic utility can emerge in evolution.

5.3 Neutral network structure

In section 3.5, the problematic effects of introducing redundancy on neutral network structure were discussed. In particular, the problems endemic in Rothlauf & Goldberg's categorisation of redundant representations [97] were discussed. It was suggested that this two-category distinction into synonymous and non-

synonymously redundant representations was inadequate, and that proponents of neutrality considered that the redundancy that resulted in neutral networks was characterised quite differently to either of the categories proposed by Rothlauf & Goldberg. This section discusses the structure of the neutral networks of the OBDD genotype representation under the neutral mutations introduced above. The resulting neutral networks are characterised as being large, fine structures which have high intra-network and inter-network connectivity.

Neutral network size

For a given genotype, g, the size of its neutral network is determined by counting all those functionally equivalent genotypes that are reachable by *neutral walk* from g. A *neutral walk* in EBDDIN is a series of single applications of neutral mutations from the set $\{N1, N1', N2, N2'\}$, or $\{N1, N1', N2, N2', N3\}$ where dynamic variable ordering is being used. Generally speaking, a given phenotype may have many disconnected neutral networks associated with it. However, given that the neutral mutations in EBDDIN are derived from Bryant's reduction operations [16] and their inverses, it follows that all genotypes associated with a given function will be connected on the same neutral network. This is because Bryant's reductions are capable of reducing any OBDD to the canonical ROBDD, and it follows that the inverse operations must be able to expand an ROBDD to any other functionally equivalent OBDD. Thus, there is one neutral network for each of the 2^{2^n} functions.

How big each of the 2^{2^n} neutral networks actually is depends very much on the function. If a function under a given variable ordering has an ROBDD representation of complexity polynomial in n, then there will be an exponential number of functionally equivalent OBDDs lying between it and the BDT representation of that function which has size $2^n - 1$. This follows from the fact that there must be at least one OBDD of each size between the sizes of the BDT and ROBDD representations in order for the canonicity property to hold via Bryant's reductions, though there are probably many more. If, however,

the ROBDD representation of a function is exponential in its complexity then there will be fewer functionally equivalent OBDDs of size lying between it and the BDT representation of the function, resulting in a smaller neutral network. Across all π, the relative size of the neutral network will depend on how the complexity of the ROBDD representation of that function varies with π.

In summary, each function has a single neutral network of genotypes associated with it. Where the function has a compact (i.e. polynomially-sized) ROBDD representation, then the size of the neutral network will be at least exponential in n. It is not known what proportion of functions have compact ROBDD representations, but the number is expected to grow at a similar rate as the number of functions (i.e. doubly exponentially). We can therefore expect, with regard to n, a doubly exponential number of at least exponentially large neutral networks in OBDD space, along with a large number of neutral networks lesser in size.

Intra-network and inter-network interconnectivity

Within a neutral network, connectivity is typically high. This follows from the fact that it will generally be possible to mutate an OBDD in very many ways using one of the neutral mutation operators. For example, any nonterminal with two or more parents can have N2' applied; any long edge with space to insert one or more variables can have N1' applied, possibly with several outcomes; there will be many options for reduction using N1 and N2 as indicated earlier; N3 can always be applied in many places. These possibilities for applying neutral mutations can be considered as *neutral degrees of freedom*, the number of which will vary from genotype to genotype, but will typically be high.

Similarly, inter-network connectivity is also high. Any non-trivial OBDD has a large number of opportunities to apply the A1 mutation, typically leading on to a different neutral network. A1 can be applied to any child edge of any nonterminal. That application of A1 to a given child edge, itself, will typically have many possible outcomes. While A1 may sometimes lead to another point

on the same network, it must generally be considered adaptive, leading onto a differing neutral network. Thus, the *non-neutral degrees of freedom* for each genotype are also high.

Thus, the neutrality inherent in this representation results in a minimal number of neutral networks (i.e. one for each function). Each neutral network is fully connected internally in many degrees of freedom, but also highly connected externally with other networks in many degrees of freedom throughout. As a result, the potential for exploration through neutral walk becomes great. This characterisation is very different to either of the two classifications of redundancy described by Rothlauf & Goldberg and discussed in section 3.5. Rothlauf & Goldberg's classifications were depicted in figure 3.1 on page 49. Here, synonymously redundant representations had genotypes that represented the same phenotype clumped tightly together in globules. Genotypes at the centre of those globules were phenotypically isolated, and the surface of each globule had connectivity with few differing phenotypes. The non-synonymously redundant representation, in contrast, depicted all the genotypes that represented the same phenotype as disconnected, so that the variation operators can not easily find redundant variations for neutral walk. They characterise these classifications formally on the sum of the distances between genotypes that represent the same phenotype (formula 3.1 on page 47) as having small values and large respectively.

Rothlauf & Goldberg's characterisation neglects the desirable properties of redundant representations that have genotypes that represent the same phenotype highly dispersed, but connected. This type of redundancy is what this thesis claims is indicative of the best properties of neutral networks. It can be formalised in a similar manner to formula 3.1, but crucially takes into account how connected the redundant genotypic representations of a given phenotype are.

$$\left(\frac{1}{2|G|}\sum_{p\in P}\left(\frac{1}{\gamma^p}\sum_{g\in G^p}\sum_{g'\in G^p|g\neq g'} distance(g,g')\right)\right) \tag{5.2}$$

where

- G is the set of all genotypes.
- P is the set of all phenotypes.
- $G^p \subseteq G$ is the set of all genotypes representing the same $p \in P$.
- $|G|$ is the cardinality of G.
- γ^p is the number of neutral networks of genotypes representing the $p \in P$.
- $distance(g, g')$ is the distance between two genotypes.

In EBDDIN $distance(g, g')$ is the fewest number of neutral mutations to get from g to g'. Crucially, formula 5.2 takes into account how connected the set G^p is, where formula 3.1 does not. Large values of (5.2) represent desirable redundancy with typically large distances between $g, g' \in G^p$ that are connected by neutral walk. Dividing the sum of the distances by $|G|$ gives an average distance between genotypes connected by neutral walk so that the average size of neutral network can be compared against the maximum possible size.

Formula 5.2 will return high values for redundancy of the type characterised for EBDDIN and implied by others (e.g. [37, 105]), but low values for both the synonymous and non-synonymously redundant representations characterised by Rothlauf & Goldberg [97]. In EBDDIN, $\forall p \in P, \gamma^p = 1$ and distances between the BDT and ROBDD representations will be exponential in the number of variables for many $p \in P$. However, for a synonymously redundant representation, $distance(g, g')$ is small for all $g, g' \in G^p$ by definition (formula 3.1). For a non-synonymously redundant representation, γ^p will be very large, approximating $|G^p|$, because $\forall p \in P, g \in G^p$ is not connected to any other member of that set.

It is also worth reemphasising a distinguishing feature of the neutrality under EBDDIN's representation. Recall from section 3.1.3 Lenski's negative argument regarding the usefulness of neutrality:

> "..., deleterious mutations may lead to genetic neighborhoods that are more promising, from the perspective of adaptation, than neutral mutations. In other words, neutral mutations are neutral precisely because they are isolated from important phenotypes, whereas deleterious ones must be connected to phenotypes that matter for fitness."

The implication is that neutral mutations happen in parts of the genotype that have no influence on phenotypic function. In EAs such as Galvan-Lopez's approach to GP [100], and CGP [84][119], this may be true to a certain extent, though there is always the potential for any non-influential part of the genotype to become influential through a single application of a variation operator. However, in an OBDD genotype there is no part of the representation that does not influence the phenotype.[3] That is, any mutation changes the way the representation calculates function. As noted in section 5.2, neutral mutations vary the pleiotropic influence of different parts of the genotype on the phenotype, and chapter 6 argues that this is important for evolvability under EBDDIN.

Similarities to RNA spaces

Neutrality has been extensively studied in RNA spaces, where RNA sequences map to secondary structure, or shape [105, 52]. Shape space is found to be considerably smaller than sequence space, implying a high level of redundancy. There are few common shapes and many rare shapes. Random mutation at a few loci typically leave shape unchanged. The likelihood of shape change, however, increases rapidly as the number of mutated loci increases. Similar sequences on a network have largely overlapping neighbourhoods, but less so for dissimilar sequences. One fifth of all shapes, the common shapes, are represented by a neutral network whose sequences are so diverse at the extremes so as not to have a single base in common. The few common shapes are represented by

[3] This was also discussed in section 3.5, where it was emphasised that EBDDIN exploits functional redundancy only.

neutral networks that are so pervasive that each is accessible from any other by few base mutations. Thus, intra-network and inter-network connectivity is high in RNA spaces also, and neutral networks are large. Furthermore, both RNA and OBDD spaces exhibit a differential in the commonality of phenotypes: functions with compact ROBDDs are more commonly represented than those with non-compact ROBDDs.

There remain very significant material differences between the RNA representation and the OBDD representation. However, these commonalities in the structure of the space may imply similarities in the search characteristics. The principal benefit highlighted by RNA researchers is that exploration is better facilitated. By drifting along a neutral network, the search is exposed to neighbours which are highly infrequent, providing a virtually endless supply of innovation according to Huynen [52]. There is little potential for getting stuck at a local optima under this scenario: there are always new phenotypes to explore.

5.4 A search space free of local optima

Proponents of neutrality often suggest that neutrality alleviates the problem of local optima [84, 50, 105, 52, 54, 119]. Yet, proof of this for any non-trivial system has been elusive. In this section EBDDIN is shown to be free of local optima for a significant class of fitness functions.

The fitness functions considered here are those in which each fitness increment is determined by a single input vector in isolation of all other input vectors. This includes those typically used for both fully and partially specified Boolean functions, and those typically used for classification problems. It is all those fitness functions where fitness is determined by counting the correct number of outputs for each specified input vector.

Theorem 1. *Where fitness is determined by counting correctly classified input vectors, and* $fitness(f_{origin}) \leq fitness(f_{target})$*, it is possible to transform any OBDD,* $f_{origin}(x_0, ..., x_n)$*, to any higher fitness OBDD,* $f_{target}(x_0, ..., x_n)$*, with-*

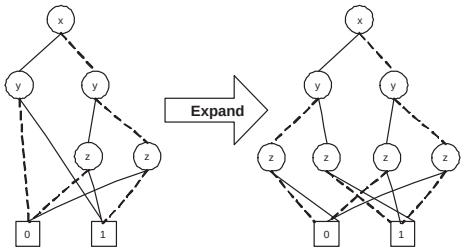


Figure 5.4: Expanding an OBDD to form a complete BDT down to $n-1$ layers.

out loss of fitness in any of the successive intermediate OBDDs, using a series of single applications of operators from the set $M = \{N1, N1', N2, N2', A1\}$.

Thus, there is always a neutral path to a higher fitness value should one exist. It is a simple matter to show that theorem 1 is correct. Consider an OBDD where all the nonterminals form a tree in which there are no long edges and every path visits all the variables. Each input vector is now represented by an independent path from the root to a terminal, the last edge of which affects only one input vector (i.e. fitness case). Thus, each input vector can be manipulated independently of any other simply by redirecting the final edge of that path to another terminal using the A1 mutation. Thus, when an OBDD genotype is in this tree form, maximum fitness can always be reached simply by applying A1 to all those paths whose output is not correct. Manipulating any OBDD into the tree form required is achieved by using the N1' and N2' neutral mutations, see figure 5.4. These have the effect of filling in gaps in a path, and splitting nonterminals that have more than one parent. Because N1 and N2 are known to always be able to reduce an OBDD in tree form to its canonical ROBDD representation, the inverse mutations can always expand any OBDD to its tree form. Given that any OBDD can be transformed to a higher fitness OBDD, should one exist, the search space can be concluded to be free of local optima. More will be said on this in chapter 7, where a search space model based on accessibility is introduced.

It is strictly not necessary for the OBDD to be expanded to tree form in order

to make progress. It is only necessary that the smallest portions of the OBDD periodically comply with the tree conditions, and this too can be achieved solely with N1' and N2'. Furthermore, theorem 1 clearly holds regardless of the variable ordering chosen. Moreover, theorem 1 holds if a neutral mutation for dynamic variable reordering is introduced, as it will be in chapter 8: the inclusion of dynamic variable reordering does not prohibit the N1' and N2' expanding an OBDD to tree form.

While it is of considerable theoretical interest to prove that there is an infinitely scalable class of problems that are free of local optima by way of neutrality, the empirical studies (section 5.7) show that this is not the way the search progresses. If it was, fitness would always accumulate in increments of 1, which would be a very slow and tedious process indeed. In practice, fitness accumulates in much larger increments.

5.5 Avoiding the cost of evaluating neutral offspring

Fitness evaluation is usually the most costly aspect of an EA and is considered one of the biggest obstacles to the scalability of EC systems. Given that EBDDIN employs mutation operators some which are explicitly neutral, those offspring produced by such a mutation can have the fitness value copied directly from the parent, circumventing a fitness evaluation. Avoiding the cost of evaluating neutral offspring is the novelty of this approach, and it is applicable regardless of the selection method used.

Instead of a mutation rate which provides a probability for mutating each genotypic feature, a mutation *bound* is used. The mutation bound caps the number of applications of the variation operators to the genotype in offspring generation.

1. generate initial population
2. select parents
3. FOR EACH n offspring to be generated
   (a) clone parent to produce child genotype
   (b) REPEAT *bound* times
      i. choose a mutation from the set M randomly
      ii. attempt to apply chosen mutation to random location in child
      iii. IF mutation failed and child not mutated
         A. GOTO 3(a)i
      iv. IF mutation is non-neutral
         A. evaluate child
      v. ELSE
         A. copy parent fitness value to child
4. kill n individuals in population
5. insert n offspring into population
6. IF terminating condition is false
   (a) GOTO 2
7. finish

Neutral offspring produced by an A1 mutation still require a fitness evaluation because it is not possible to determine the effect of A1 in advance. Using a low mutation bound and high percentage of explicitly neutral mutations, however, a large proportion of function evaluations can be avoided, typically reducing the number of fitness evaluations by around 50%.

This method is not specific to EBDDIN, and it will work regardless of the selection mechanism used, and regardless of whether steady-state of a generational algorithm is used. The method can be applied where a variation operation is known a priori to have a neutral consequence on the offspring. This can be as a result of an explicitly neutral operator, or as a result of applying a general operator to a part of the genotype which is known to be non-functional. The former approach is applied here, and the latter approach was applied by Galvan-Lopex [100] to tree-based GP.

5.6 Managing bloat

There are two types of redundancy that can be associated with the OBDD representation, intra-π redundancy and inter-π redundancy. The former results from genotype OBDDs being unreduced. The latter results from the variation in complexity of ROBDD in response to π. A good π will produce a very simple ROBDD representation of a function, but a bad π will produce a very complex representation: this is merely the variable ordering problem restated. In this section, I will introduce methods for managing intra-π bloat; the variable ordering problem will be addressed in a later chapter.

Controlling intra-π bloat

Intra-π redundancy can easily be removed by reducing an OBDD to its ROBDD form. However, a certain amount of redundancy is beneficial as it facilitates genotypic diversity independent of fitness and phenotypic diversity. Without such diversity, the exploration resulting from neutral drift would be lost.

Therefore, rather than simply removing all redundancy at the earliest opportunity, the amount of redundancy must be managed. The optimal level of redundancy to minimise the number of function evaluations required to find a solution is difficult to determine, but empirical studies suggest a modest amount of redundancy, having genotypes with size typically less than twenty times that of the ROBDD solution. For practical reasons too, a lower amount of redundancy requires less memory: bloated OBDDs approximating tree form require an amount of memory exponential in the number of variables.

Three broad approaches are suggested:

1. Periodically fully reduce the genotypes of every member of the population to ROBDD form.

2. Attempt to control the relative frequency of reducing mutations so as to control the level of redundancy, or by using the 'aggressive' form of N2.

3. Allow the inherent tendency for genotypes to become reduced to handle bloat automatically; the mechanism of this is discussed in chapter 6.

5.7 Examples

In this section EBDDIN is tested on a number of benchmark problems to characterise the search. It is also compared to several other approaches to evolving BDDs, demonstrating an improvement in performance of orders of magnitude. The approaches used for comparison are Yanagiya [128], Sakanashi [103] and van Remortel [118].

A range of n-parity problems were investigated in the range of $n = 7, \ldots, 17$, to see how the EBDDIN approach scaled. Other problems were also looked at for comparison with previous approaches and to investigate how this approach coped with less compact ROBDD solutions. There were 100 runs on each problem, except 20-mux, which was done over 10 because of time constraints. Population size = 5, tournament size = 2 and mutation bound = 1. The fitness functions used for all these problems was a negated count of incorrectly classified fitness cases except for the parity problems which used the fitness function:

$$fitness = abs(fitness_{even} - 2^n/2) + 2^n/2 \tag{5.3}$$

where $fitness_{even}$ is the number of fitness cases identified correctly for even-parity. This fitness function exploits the fact that a below median fitness value can be pivoted about the median simply by swapping the terminals in an OBDD [26]. Effectively, (5.7) permits the evaluation of two solutions for the cost of evaluating one. However, in practice, the second solution is almost always subordinate to the first, so there is little or no benefit over a straightforward count of correctly classified fitness cases. (5.7) was used on parity functions simply to investigate its potential. The compared approaches of Yanagiya , Sakanashi

and van Remortel use a fitness function equivalent to that used for non-parity problems.

The population was initialised by randomly generating OBDDs varying in size. Table 5.1 shows the results for the parity problems and table 5.2 shows the results for some other symmetric functions and some multiplexer functions. Tables 5.3, 5.4, and 5.5 show the results of Sakanashi, Yanagiya and van Remortel for comparison with the EBDDIN approach.

For the parity problems, a 100% success rate is observed for each n variant within a very modest number of fitness evaluations. Modest values for σ and mean run length are also observed. In terms of *scalability*, this algorithm exhibits excellent behaviour. The mean run length required increases at a lesser rate than the number of fitness cases, as does σ. The tendencies of these properties are clearly observed when plotted against n. See figure 5.5.

EBDDIN was also tested on a number of symmetric and multiplexer problems for comparison with other approaches. This included two experiments on multiplexer problems with reverse variable orderings, i.e. mux6(r) and mux11(r). This was done to determine the influence on the algorithm of variable ordering and compactness of ROBDD solution. Table 5.2 shows the results. A similarly high success rate was exhibited for all these problems except mux11(r). Table 5.2 suggests that the more compact a problem's ROBDD solution, the easier it is to find. For example, the ROBDD solution to mux6 has 7 non-terminals, while the solution to mux6(r) has 29 (see figure 2.3), resulting in a much shorter mean run length for mux6.

The reason this algorithm performs better against problems with compact ROBDD solutions is possibly related to the combination of redundancy and high degree of subfunction reuse in compact ROBDDs, with regularity in the solution being effectively exploited throughout the run, focusing the search on promising areas of the search space.

n	7	8	9	10	11	12	13	14	15	16	17
SUCCESS RATE	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
MEAN FITNESS ($2^n max$)	128	256	512	1024	2048	4096	8,192	16,384	32,768	65,536	131,072
MEAN RUN LENGTH	1,272	1,865	2,659	3,906	4,921	6,541	8,046	9,768	11,852	14,993	16,594
SHORTEST RUN	510	719	1,545	1,788	2,920	3,217	4,495	5,943	8,652	9,475	11,765
LONGEST RUN	2,857	4,407	4,806	6,727	9,273	10,848	13,139	15,017	19,289	24,354	30,349
σ	428	571	650	1,165	1,193	1,469	1,819	2,049	2,105	2,957	2,789
RUNS WITHIN $\frac{1}{2}\sigma$	34%	51%	38%	37%	43%	37%	37%	40%	40%	40%	42%
RUNS WITHIN 1σ	66%	79%	72%	73%	65%	72%	70%	73%	67%	70%	80%
RUNS WITHIN 2σ	97%	93%	97%	91%	96%	95%	96%	95%	96%	96%	95%
RUNS WITHIN 3σ	99%	98%	98%	100%	99%	100%	100%	100%	98%	99%	98%
RUNS WITHIN 4σ	100%	99%	100%	100%	100%	100%	100%	100%	100%	100%	99%

Table 5.1: Results for parity problems. The success rate and mean fitness are shown in the two rows at the top. Below, the mean, shortest, longest and standard deviation (σ) are given for the length of runs, where the length of a run is defined by the number of function evaluations required. At the bottom, as an indication of the run length distribution, five rows give the percentage of completions within the stated number of standard deviations from the mean.

n	9 SYM	10 SYM	11 SYM	MUX6	MUX6(R)	MUX11	MUX11(R)	MUX20
SUCCESS RATE	100%	100%	100%	100%	100%	100%	0%	100%
MEAN FITNESS (0 MAX)	0	0	0	0	0	0	N/A	0
MEAN RUN LENGTH	10,719	15,745	32,231	424	3,321	3,010	N/A	17,253
SHORTEST RUN	5,001	6,387	11,239	93	985	1,550	N/A	9,679
LONGEST RUN	19,938	38,579	76,601	1,051	6,987	7,858	N/A	31,519
σ	3,121	7,499	11,969	240	1,249	1,138	N/A	5,878
RUNS WITHIN $\frac{1}{2}\sigma$	36%	42%	41%	33%	33%	48%	N/A	50%
RUNS WITHIN 1σ	68%	71%	76%	69%	76%	76%	N/A	70%
RUNS WITHIN 2σ	95%	91%	94%	95%	95%	96%	N/A	90%
RUNS WITHIN 3σ	100%	99%	99%	100%	100%	97%	N/A	100%
RUNS WITHIN 4σ	100%	100%	100%	100%	100%	99%	N/A	100%

Table 5.2: Results for other problems.

PROBLEM	3 PARITY	4 PARITY	5 PARITY	6 PARITY	MUX6	MUX11	4 SYM	5 SYM	6 SYM
SUCCESS RATE	100%	60%	20%	20%	0%	20%	100%	100%	0%
LONGEST RUN	3,500	N/A	N/A	N/A	N/A	N/A	10,300	36,000	N/A

Table 5.3: Sakanahsi's results (reformatted). This approach is able to solve up to 6-parity with a 20% success rate, but only up to 3-parity with a 100% success rate.

PROBLEM	MUX11	MUX20
SUCCESS RATE	100%	100%
MEAN RUN LENGTH	104,000	1,360,000
SHORTEST RUN	88,000	820,000
LONGEST RUN	160,000	1,696,000

Table 5.4: Yanagiya's results (reformatted). This approach requires 35 times and 79 times more function evaluations, on average, for MUX11 and MUX20 respectively.

PROBLEM	MUX6	MUX6(R)	6 PARITY
SUCCESS RATE	100%	100%	100%
MEAN RUN LENGTH	4,700	5,400	6,250
σ	350	300	250

Table 5.5: van Remortel's results (reformatted). This approach requires 11 times more function evaluations for MUX6, but only 1.6 times more for MUX6(R), demonstrating how the performance of the implicit neutrality approach is affected by the size of solution.

The results of previous approaches reproduced in tables 5.3, 5.4 and 5.5 respectively, indicate the that the EBDDIN approach is far superior. It is typically able to solve more challenging problems than the compared approaches. Where the compared approaches have solved the same problem, the EBDDIN approach is typically more efficient in the number of fitness evaluations required. For example, on the mux6 problem, it requires 10 times fewer fitness evaluations than van Remortel's [118] approach; on the mux11 problem, EBDDIN requires 35 times fewer fitness evaluations than Yanagiya's [128] approach. However, on the mux6(r) problem, the EBDDIN approach proves less superior, though still requires fewer fitness evaluations than van Remortel. This supports the conjecture that this algorithm excels at finding solutions with compact ROBDDs.

The EBDDIN approach also proved favourable in terms of memory requirements. Its small population size and straightforward implementation only required about 30 MBs for mux20, whereas Yanagiya's large population size (4000) and complex caching scheme required over 300 MBs for the same problem. Yanagiya suggests that memory requirements are a limiting factor for the application of his approach.

5.8 Difference evaluation

In section 5.5, the problems associated with the cost of fitness evaluation were emphasised, and it was shown that explicitly neutral mutations allowed around half of all fitness evaluations to be circumvented. This section introduces the method of *difference evaluation*, which is able to reduce the cost of each necessary fitness evaluation considerably by orders of magnitude.

The steps of the method are as follows:

1. Apply the logical XOR operation to parent and offspring and obtain the result OBDD. (See [4] for a detailed description of applying BDD logical operations.)

2. Copy the parent fitness value to the offspring.

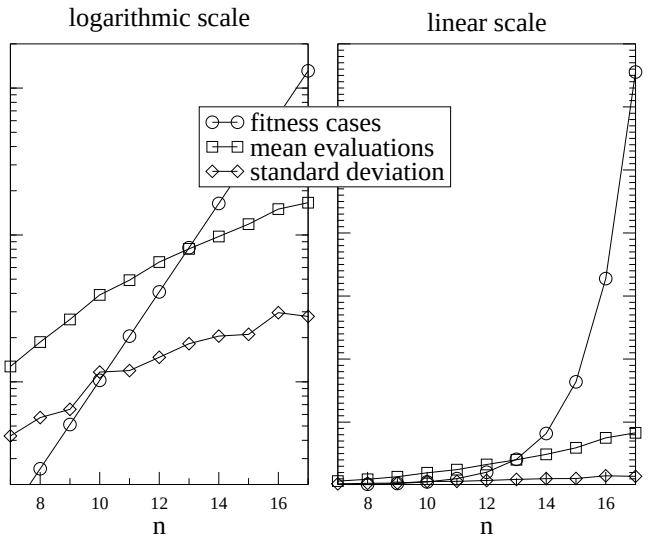


Figure 5.5: Scalability. The graphs show that, while the number of fitness cases increases exponentially, the number of function evaluations required increases approximately linearly. This demonstrates good scalability in the number of fitness evaluations required. It also confirms that the mechanism of evolution differs from that described to show freedom from local optima in section 5.4, which requires fitness increments of 1 exclusively.

3. Determine the input vectors leading to the 1-terminal in the result OBDD: call this the *discrepancy set*. The discrepancy set is all those input vectors in which the parent and offspring differ in their outputs.

4. For each input vector in the discrepancy set, determine whether it is a positive or negative discrepancy with reference to the target function, incrementing or decrementing the child fitness accordingly.

The logical XOR operation on two operand OBDDs results in a OBDD representing the difference between the parent and offspring. In the result OBDD, the discrepancy is represented by all the paths from a root to a 1-terminal, the 1-paths. Any input vector that maps onto a 1-path in the result represents a discrepancy between the two operands: each discrepancy is equivalent to a line in the truth table where the respective functions would differ. The input vectors mapping onto the 1-paths can be determined using a bottom-up recursive algorithm.

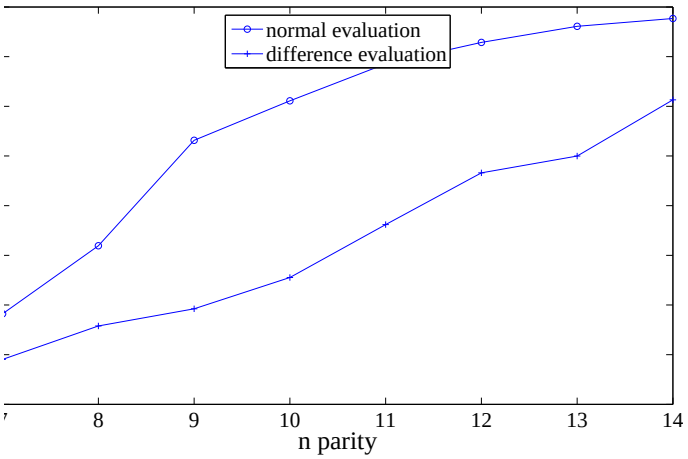


Figure 5.6: Fraction of total computation time spent in fitness evaluation. Compares normal evaluation against difference evaluation averaged over 10 complete runs for each value of n.

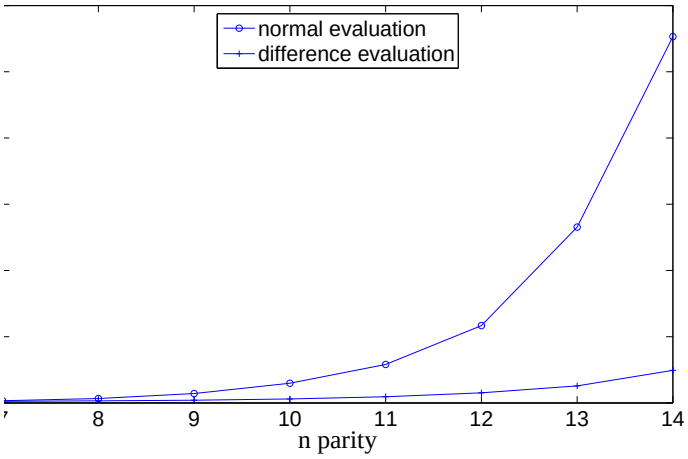


Figure 5.7: Mean time per fitness evaluation. Compares normal evaluation against difference evaluation averaged over 10 complete runs for each value of n

Comparison of the difference evaluation and normal evaluation methods were carried out using an arbitrarily chosen $(5+10)$ES and employing the aggressive N2 operator to help manage bloat. Figure 5.6 shows that the fraction of total computation time becomes saturated by fitness evaluation with increasing n for the parity problem. The fraction is much higher for normal evaluation than for difference evaluation. Figure 5.7 compares the mean time per fitness evaluation. A target function that is fully specified has a cost for each evaluation that is exponential in the number of variables, 2^n, and this is reflected in the figure for the normal evaluation method. Difference evaluation, however, results in a much less rapid increase in the time required, requiring less than 10% of the time when $n = 14$, and the gap is clearly increasing.

Yanagiya [128] employed a similar method in his work on evolving BDDs. However, Yanagiya applied the logical XOR operation to the *target* and the offspring rather than the parent and the offspring. The method of difference evaluation has two advantages over Yanagiya's method of evaluation. Firstly, Yanagiya's method requires a BDD representation of the target function at the beginning of the run: it requires the solution that is being searched for. This is of no consequence for benchmarking on well-known functions, but is prohibitory for functions where the solution is not known in advance. Secondly, as noted in the following paragraph, the number of discrepancies between the parent and the offspring will impact on performance. Yanagiya's approach will result in very large discrepancy sets at all but the latter stages of a run because the discrepancy between the target and the offspring will be large. The difference evaluation method, however, using a low mutation severity, will see high parent to offspring fitness correlation and smaller discrepancy sets throughout a run.

Generating the result OBDD can be done in time polynomial in the complexity of the operand ROBDDs [4]. Finding the underlying discrepancies can be achieved in time polynomial in the number of discrepancies. Thus, the efficiency of this method is dependent on the complexity of the operand OBDDs, and on the average size of the discrepancy sets produced during a run. The for-

mer implies that parsimonious genotypes must be encouraged in the population, and this can be aided through good bloat management. The latter implies a low mutation rate to encourage close fitness correlation of parent and offspring; this is consistent with gradualism.

The size of the discrepancy set will be typically small, but this will not always be the case, resulting in a bottleneck for the method. It is possible to further enhance the difference evaluation method by choosing a policy for abandoning counting the discrepancies between parent and offspring. Consistent with the Darwinian notion of gradualism, the more severe the effect of a mutation on function, the less likely it is that the resulting offspring will be fitter than the parent. This likelihood can be exploited, perhaps by taking only a sampling of the discrepancies at first to see if a fitness improvement is possible. If a fitness improvement appears unlikely, then the evaluation can be abandoned. Using such enhancements, coupled with tight bloat control, difference evaluation might possibly be able to reduce the cost of fitness evaluation from exponential in the number of variables to approximating polynomial.

5.9 Discussion

EBDDIN is a highly practical EA to set up and use: there is no choice of function set to make; redundancy is easily removed via reduction; fitness evaluation can be very fast compared to other approaches. Furthermore, performance is comparatively superior to other approaches to evolving BDDs. The main emphasis of this thesis is understanding the subtleties of evolvability within the context of EBDDIN, rather than development of EBDDIN for particular practical applications. It should be remembered, though, that it is a logical outcome that a deeper understanding of evolvability issues will benefit the development of EBDDIN for practical applications where the solution domain can be suitably encoded into BDD form. In this thesis, EBDDIN will be used as a computational model of an evolutionary system. The model captures many of the important

elements for studying some key evolvability concepts. The rest of this section argues the validity of EBDDIN for such a model.

In nature, evolution is dependent on the environmental conditions. The chemical compounds that are present necessitate the molecules and processes that facilitate the organism. The features necessary for organismal evolution are not necessarily essential for artificial evolution, however. The genes, proteins, cells and developmental processes, etc., that facilitate the biological organism are a consequence of evolution in the natural environment rather than prerequisites for any evolutionary system generally. Attempts to address evolvability in EC by too closely mimicking features of natural evolution results in complexity that serves only to obscure or inhibit the important and general evolutionary concepts being studied.

Any individual in any evolutionary system can be considered a function: a mapping between inputs and outputs. In nature, that mapping is extremely complex, being determined mostly by the phenotype which, in turn, is determined by genes and developmental processes that sit in between. However, at its heart, the phenotype can be considered as determining a *relationship* between the inputs (values of the environmental variables) for producing the outputs (response). An OBDD represents a function in its structure which determines a hierarchical relationship between the inputs for producing the outputs, void of many of the features of natural genotypes.[4] Mutation here serves to change the relationship between the inputs more directly using edges, and it is this that makes the model both non-trivial but simple and useful. A trivial representation would specify the output for each and every input vector independently, as in a truth table for Boolean functions. The OBDD, however, captures the output for many input vectors in a more modular fashion, also offering many alternative representations for the same function. This makes the model useful for studying both neutrality and modularity within the genotype, which a trivial

[4]According Radcliffe & Surry's [92] formal categorisation of representations, the OBDD representation falls into the category of an *allelic* representation. An allelic representation does not necessarily have genes, but a collection of properties.

representation would prohibit.

5.10 Summary

This chapter introduced a new approach to evolving BDDs, EBDDIN. Neutrality inherent in the representation is exploited by the approach. The neutral networks that result are large and fine-structured, with high intra-network and inter-network connectivity. These neutral network are quite different to either of the categories proposed by Rothlauf & Goldberg [97], but similar in some ways to that of RNA spaces. The resulting search space is shown free of local optima for an infinitely scalable class of fitness functions, including those typically used for classification problems and Boolean functions. Neutral mutation varies modularity and pleiotropy, giving these properties the potential to evolve. Explicitly neutral mutations allow a large number of function evaluations to be circumvented. The method of difference evaluation reduces the cost of fitness evaluation by orders of magnitude. EBDDIN proved superior in performance against the compared approaches on several problems, and EBDDIN was shown to scale well on parity problems. Performance was less good where the target did not have a compact ROBDD representation. Finally, EBDDIN was argued to be a viable to computational model of an evolutionary system that can be used for studying evolvability-related concepts.

Chapter 6

On pleiotropy, modularity and neutrality: constraining phenotypic variation

This chapter aims to determine what genotypic properties facilitate evolvability within EBDDIN. Evolvability requires the variation operators to transform the genotypic representation in ways which perturb the maladapted aspects of the phenotype while leaving unperturbed the adapted aspects. The hypothesis presented is that modularity facilitates the patterns of pleiotropy within the genotype that constrain phenotypic variation to facilitate the variation component of evolvability within the context of EBDDIN. The hypothesis is supported by experiments demonstrating the emergence of pleiotropic patterns which differ significantly from what would be expected of randomly chosen genotypes.

6.1 Introduction

A principal requirement for the facilitation of evolvability is targeting phenotypic variation on properties susceptible to positive selection. Altenberg [2] states of

evolvability:

> "It comes from the genetic operators being able to transform the representation in ways that leave intact those aspects of the individual that are already adapted, while perturbing those aspects which are not yet highly adapted. Variation should be channeled toward those "dimensions" for which there is selective opportunity."

But how, and what are the properties and mechanisms that facilitate it? For evolvability to be applied to systems of artificial evolution, it must be understood and characterised. In a trivial representation in which there is a one-to-one correspondence between genotypic features and phenotypic traits, mutation is just as likely to affect one feature as it is any other, so variation is not channeled at all. In a non-trivial representation, for which there is not a one-to-one correspondence between genotypic features and phenotypic traits, the genotypic representation of adapted and maladapted traits overlaps. Thus, perturbing the maladapted traits while leaving the adapted unperturbed appears to be an insurmountable problem only exacerbated by any increase in fitness..

This chapter presents an hypothesis regarding the role of modularity in channelling phenotypic variation towards those dimensions for which there is selective opportunity. This hypothesis is referred to as *Evolvability via Modularity-induced Mutational Focussing* (EMMF). EMMF postulates that modularity better permits pleiotropic patterns to emerge that separate out and compress the genotypic representation of adapted traits, leaving the maladapted traits exposed to perturbation, effectively channelling phenotypic variation.

Section 6.2 discusses different notions of modularity. Section 6.3 discusses pleiotropy and the pleiotropic patterns that might be beneficial to evolvability. Section 6.4 presents a thought experiment to aid in understanding modularity and pleiotropy work together to produce pleiotropic patterns beneficial to evolvability. Section 6.5 presents experiments used to support the hypothesis. Finally, section 6.6 discusses the findings and their implications.

6.2 Notions of modularity

The intuitive idea of modularity is fairly straightforward. However, there are many notions of modularity and the term is used in many different contexts to refer to a multitude of concepts. In computer science and engineering modularity provides systems designed as distinct functional components, each of which solve some particular aspect of the larger problem. These components have clearly specified interfaces through which they interact with other components, screening off the internal complexity. Such design principles facilitate easy reuse of components so that common functionality does not have to be re-designed again and again.

In evolutionary biology, concepts of modularity differ. *Developmental modules* are units of embryonic development that are largely independent of the context in which they occur [93]. These kinds of modules materialise as discernable phenotypic entities: a limb or eyeball, for example. Wagner & Altenberg's notion of *evolutionary modules* [122] are defined in terms of variational independence: a pair of limbs vary in length as one unit, or a pair of eyes vary in colour as a unit.

Wagner & Altenberg's [122] notion of evolutionary modularity has been the subject of particular interest in EC in recent years because it is purported to be responsible for channelling phenotypic variation and appears to be important for evolvability. This type of modularity is depicted as existing through a close coupling of pleiotropic effects between a gene group and *character complex*. That is, a gene group's pleiotropic influence is primarily, but not necessarily exclusively, within a group of traits which have become integrated through the genotype-phenotype mapping process to serve some primary function. See figure 6.1.

In a later work, Altenberg [3] states of evolutionary modularity:

> "The extreme example of modularity would be the idealized model of a genome in which each locus maps to one phenotypic trait. For

> the converse, the extreme example of non-modularity would be a genotype-phenotype map with uniform pleiotropy in which every gene has an effect on every phenotypic variable. Real organisms, one could argue, have genotype-phenotype maps that range somewhere in between these extremes."

Under Altenberg's notion of evolutionary modularity then, and fitness functions covered by theorem 1 on page 77, an OBDD representation of a function in tree form would allow each and every trait (fitness case result) to vary independently by the A1 mutation, and would therefore be maximally modular in an evolutionary sense. However, under the notion of the kind of modularity that emphasises reuse (definition 5.2, p.67), the OBDD in tree form is minimally modular because no reuse is exhibited. The tree is maximally modular in one sense but minimally modular in another! Conversely, the ROBDD representation of the same function is minimally modular in an evolutionary sense because traits have little capacity to vary independently, but maximally modular in a reuse sense because all redundant function is removed and reuse is optimal. These two notions of modularity are completely antagonistic within the context of EBDDIN.

A further point to note with the notion of evolutionary modularity as described above is the question of determining traits. Altenberg further states:

> "Intuitive notions about the advantages of modularity for evolvability run into the problem of how we parse the organism into traits. In order to resolve the "question of multiplicity", there needs to be a way to get the human observer out of the way, and define modularity in terms of physical processes." ... "Until this problem is resolved, we cannot say whether variation is modular or not."

That is, 'traits' determined by humans are not necessarily optimal, or even good, units of variation. Better units of variation may consist of multiple 'traits' varying together in some correlated manner: for example, under EBDDIN, it

would be preferable to see fitness increases having a magnitude much greater than 1, as noted in section 5.4. How are these optimised units of variation to be determined in evolution? Altenberg concludes that:

> "My main proposal is that the evolutionary advantages that have been attributed to modularity do not derive from modularity *per se*. Rather, they require that there be an "alignment" between the spaces of phenotypic variation, and the selection gradients that are available to the organism. Modularity in the genotype-phenotype map may make such an alignment more readily attained, but it is not sufficient; the appropriate phenotype-fitness map in conjunction with the genotype-phenotype map is also necessary for evolvability."

That is, what is more fundamental to evolvability than evolutionary modularity in the genotype-phenotype map, is how the genotype-phenotype map aligns phenotypic variation with fitness. With this crucial insight in mind, an investigation into evolutionary modularity can be set aside in favour of the more pertinent investigation into how the genotype-phenotype map aligns phenotypic variation with fitness, and the role that modularity, in a reuse sense, may have to play in that.

6.3 Pleiotropy

If evolutionary modularity is not responsible for facilitating the variational component of evolvability, then pleiotropy may be. Pleiotropy happens where single gene influences multiple phenotypic traits. Hansen [49] argues that patterns of pleiotropy may facilitate evolvability rather than evolutionary modularity.

As long ago as 1930, it was recognised that the probability of a mutation being advantageous is a steeply decreasing function of the number of traits it influences [41]. High levels of pleiotropy cause interference amongst traits, inhibiting their capacity to vary independently. Assuming that the majority of traits are well-adapted, perturbing more of them simultaneously is more likely

to result in a reduction in fitness. Conversely, minimal pleiotropy (i.e. one gene per trait) permits each trait to vary independently of any other and is precisely the same thing as Altenberg' notion of evolutionary modularity in the extreme discussed in the preceding section. Therefore, minimal pleiotropy suffers from the same problem as extreme evolutionary modularity in terms of how to parse the organism into traits of useful variation. Hansen [49] argues that an intermediate level of pleiotropy may best facilitate evolvability, and suggests that variation in pleiotropic effects within a genotype will be important to facilitating favourable pleiotropic patterns.

What patterns of pleiotropy might be beneficial in an OBDD genotype?

Section 5.2 defined pleiotropy in the context of EBDDIN as the number of fitness cases which a single genotypic feature is involved in processing. *Pleiotropic utility* (U_p), or simply *utility*, was defined as the fraction of pleiotropy that are adapted fitness cases. So, while pleiotropy quantifies how many fitness cases a given edge or vertex influences, utility provides an indication how good that influence actually is.

The pleiotropic patterns that can be expected to benefit evolvability in EBDDIN will be those which expose maladapted fitness cases to perturbation without exposing the adapted fitness cases. For example, consider an edge in an OBDD which has $U_p = 0$. Mutating such an edge using A1 cannot result in a fitness drop, but will likely result in a fitness increase because only maladapted fitness cases are influenced by that edge. Such edges clearly make good targets for mutation. A high percentage of edges exhibiting $U_p = 0$ in the genotype will benefit evolvability, therefore. It may be too much too expect to have a number of edges with $U_p = 0$, particular as fitness increases, but relatively low U_p will serve a similar purpose, though to a lesser degree. Assuming no mutation or representation bias, A1 mutation of any random subset of fitness cases clearly has the expectation that half will turn out to be correct on average:

$$\text{Expected outcome of mutating an edge by A1 is } U_p = 0.5 \tag{6.1}$$

Therefore, edges with $U_p \lessapprox 0.5$ are reasonable targets for A1 mutation and will frequently result in a fitness increase. Some variance from (6.1) permits a fitness increase where U_p of an edge is slightly greater than 0.5 before A1 application, but this is clearly limited if the variance follows a normal (Gaussian) distribution.

Another pleiotropic configuration may be beneficial to robustness. An edge that has high U_p represents only adapted fitness cases, which means they do not have to be perturbed in order for the maladapted fitness cases to be perturbed. If the adapted fitness cases can be represented by a small percentage of edges having high U_p, robustness of the adapted fitness cases will be enhanced.

In conclusion, pleiotropic patterns that favour evolvability may be those which separate the representation of adapted and maladapted fitness cases, and this will be evident in the U_p distribution. The expectation of U_p (U_E) is:

$$U_E = \frac{fitness}{2^n} \tag{6.2}$$

Where $fitness$ is the number of adapted fitness cases and $2n$ is the total number of fitness cases. The distribution of U_p might be expected to approximate a normal (or, perhaps, skew-normal) distribution with mode U_E in a randomly generated genotype. That is, any edge in a randomly generated genotype is expected to processes an unbiased proportion of adapted and maladapted fitness cases on average. As the population matures and the gap between (6.1) and (6.2) increases (see figure 6.2), fitness improvement becomes increasing unlikely if the actual distribution of U_p approximates a normal distribution. A distortion in the actual distribution of U_p away from the normal, exhibiting secondary peaks or fattening in the upper and lower tails of the distribution indicates separation of the genotypic representation of adapted and maladapted traits. Such pleiotropic patterns will better permit the perturbation of the maladapted

while leaving unperturbed the adapted and are therefore be responsible for aligning phenotypic variation with fitness.

6.4 Focussing mutation: a thought experiment

This section examines how modularity can assist in producing patterns of pleiotropy that might facilitate evolvability. Section 5.2 defined modularity, for the purposes of this thesis, in terms of reuse and the removal of redundancy, making the representation of a given function more efficient in terms of its size. This notion of modularity is used in many EAs.[1] Section 6.2 showed that this notion of modularity could be antagonistic with Wagner & Altenberg's [122] notion, which Altenberg [3] now acknowledges may not be as important to evolvability as previously thought. Section 6.3 looked at what patterns of pleiotropy might be beneficial to evolvability in EBDDIN. This section looks at how pleiotropy and modularity, together, can facilitate evolvability within the context of EBDDIN by way of a thought experiment.

Using only the N1' and N2' mutation operators, any ROBDD can be expanded so that the nonterminals form a tree. This tree representation has size exponential in the number of variables in contrast to the ROBDD representation which may be linear in size; 11-mux will be used as the example. The tree representation of 11-mux has the highest level of redundancy, with many redundant subfunctions. It is the fact that the tree representing 11-mux has high regularity of subfunction that allows the tree to be compressed to a maximally modular ROBDD having linear complexity.

Now, consider the effect of random mutation, without selection, on the tree representation of 11-mux. The mutation operators are both neutral and non-neutral and applied to random locations repeatedly. Further, assume that the size of the OBDD remains similar to the tree. Clearly, the function represented by the OBDD will change from 11-mux. However, and more importantly, mu-

[1] Section 3.2.1 reviewed some of the approaches to exploiting modularity in EC.

tation will disrupt the regularity that was present before. It must therefore be expected that a greater variety of subfunctions will result. The consequence will be that the mutated OBDD will be less compressible than 11-mux and have a larger ROBDD representation: the potential for modularity will be reduced.

Now consider the tree representation of 11-mux again. This time, however, mutation is applied only to one side of the tree: the other half is left untouched. Clearly, the untouched half will retain both its function and compressibility, but the mutated half will retain neither. The representation of the functional part of the OBDD that is correct can be compressed into a modular representation, while the part that is not functionally correct will have a comparatively larger representation after compression. That part of the genotype that represents the incorrect part of the function now presents a relatively bigger target for mutation. Thus, the part of function that is incorrect attracts a disproportionate amount of applications of the A1 operator under random mutation. That part of the function that is correct is protected from mutation by the fact that it is compressed.

The situation described above could not arise under normal evolutionary forces. It simply serves as an example to illustrate that the representation of the correct part of the function, which would be under selection, is compressible while the representation of the randomised maladapted part, which would not be under selection, is not. The significance of the modularity is that the genotypic representation of both the adapted and maladapted fitness cases does not have to be proportional to their relative number. If that were not the case, a positive mutational perturbation would become prohibitively unlikely as fitness increases and the genotype becomes swamped by the representation of adapted fitness cases.

How might favourable pleiotropic patterns emerge?

The short answer to this question is by second-order selection for pleiotropic patterns which favour evolvability. To address the question in a little more

detail, the combined effects of all the mutation operators and selection must be considered over several generations. It will assist to group the effects of mutation into three principal forces:

- *Disruptive force.* The expanding neutral mutations, N1' and N2', combined with the non-neutral A1 mutation serves to decompress and disrupt regularity, inhibiting compression.

- *Compressive force.* Reducing neutral mutations, N1 and N2, serve to compress parts of the representation, regardless of whether those parts represent adapted functionality.

- *Preservative force.* Selection serves to preserve the adapted part of the function, propagating it to future generations.

Expanding parts of the genotype that represent adapted functionality (i.e. fitness cases) greater exposes the adapted functionality to disruption by A1. Given that disruption of adapted functionality will more likely result in deleterious offspring, such lineages will be selected against. This is nothing more than stabilising selection preferring those configurations that less expose adapted functionality to mutation. Expanding parts of the genotype that represent maladapted function, however, will greater expose the maladapted functionality to disruption by A1, which will serve to improve fitness and inhibit compression of the representation of the maladapted function, and will be selected for. Pre-existing or newly adapted functionality that is uncompressed is compressible using N1 and N2, while maladapted functionality is less so because it is randomised. That maladapted functionality that is compressible is relatively more susceptible to expansion and functional disruption, and so the cycle continues. The result is a neutral shuffling of pleiotropic influence within the genotype to favour pleiotropic patterns that compress and protect adapted functionality while exposing and perturbing maladapted functionality, and is dependent on the the target function having a compact ROBDD representation.

6.5 Hypothesis

The previous discussions lead to the following hypothesis, which is termed *Evolvability via Modularity Induced Mutational Focussing* (EMMF):

> *Patterns of pleiotropy that separate the genotypic representation of adapted and maladapted traits can emerge in evolution to facilitate the variation component of evolvability. The emergence of such patterns is better facilitated where there is potential for modularity to compress the genotypic representation of adapted traits.*

The hypothesis is testable by comparing how distributions of U_p evolve on two target functions: one function that has the potential for modularity, and one which does not. Where there is no potential for modularity, the distribution of U_p should approximate a normal distribution with mode near U_E. In contrast, where there is potential for modularity, the distribution of U_p should exhibit secondary peaks or fattening of the upper and lower tails.

The two test functions chosen are variants of the 11-bit multiplexer. The first, 11-mux, has an optimal variable ordering and a compact ROBDD representation: this function has a high potential for modularity in the genotype. The second test function, 11-mux(R) has the reverse variable ordering and a complex ROBDD representation: it has little potential for modularity in the genotype.

Comparing the distributions of U_p

Figure 6.3 contrasts frequency polygons for 11-mux and 11-mux(R). Each frequency polygon has 20 buckets with centres in the range 2.5 through 97.5 in steps of 5, representing frequencies of U_p as a percentage. There are five sub figures, each representing intervals of fitness as U_E in the range indicated by the vertical bars. Samples are taken of the parent OBDD genotype at each fitness improvement step and averaged over the interval, though the distributions were found not to be significantly different at other times also. To accommodate

genotypes of varying size, U_p frequencies are normalised for each individual so that the sum of frequencies is 1. A $(1+5)$ ES was employed using both neutral and adaptive mutations.

The shape of the distributions are telling. 11-mux(R) approximates a normal distribution of U_p as predicted, with mode near U_E. This indicates each edge influences a relatively unbiased proportion of adapted and maladapted fitness cases, on average. Such a pattern of pleiotropy is unlikely to be beneficial to evolvability. However, 11-mux has a very different distribution. As fitness increases, the distribution spreads considerably from U_E, with a much higher proportion of edges exhibiting maximum values of U_p, or values much less that U_E. The distribution begins to form a hollow at U_E; the upper tail soon disappears and the lower tail exhibits a significant secondary peak. Like 11-mux, the parity function has a compact ROBDD and high potential for modularity, and exhibits a similar distribution to 11-mux (figure 6.4). The experiments support the hypothesis.

Comparing the fitness curves

Figures 6.5 and 6.6 show the fitness curves for the best individual for 11-mux and 11-mux(R) using a (5+10) ES. There is a very clear contrast here. The figures show that it is not only the pace of evolution that differs, but also the manner. While 11-mux(R) exhibits a fairly consistent gradual curve, the curves for 11-mux are much more erratic. Long periods of stasis are interspersed with periods of rapid evolution, and this is indicative of Gould's [46] punctuated equilibria phenomenon.

Two of the main stasis points are indicated in figure 6.5. These points of stasis happen at fitness levels of -128 and -256. Once a stasis point is broken free of, fitness increases at a rate not appearing to significantly decrease from the rate prior to stasis, and sometimes even accelerates. The implication is that the solution has emerged in component parts somewhat consistent with the subfunctions of the ROBDD target representation. The stasis points are indicative

of high-level components the evolutionary foundations of which have not yet emerged. Once the foundation of the missing component is discovered, functional adaptation continues apace, building on the foundation using preexisting lower-level components. That no similar stasis points occur for 11-mux(R) is, therefore, perhaps due to the fact that there is little potential to exploit preexisting lower-level subfunctions when there is little potential for modularity.

These experiments strongly support the hypothesis. Maladapted fitness cases are clearly greater exposed to perturbation through pleiotropic patterns exhibiting a relatively high percentage of low U_p values, but this only occurs where there is potential for modularity in the genotype. In addition, the experiments suggest that vertices having high U_p values may serve as useful building blocks to be exploited and reused, being good destinations for an edge subject to A1 mutation. A high percentage of high U_p vertices in the genotype may even raise the expected outcome of A1 mutation of an edge well above $U_p = 0.5$ as per (6.1).

Exploiting the pre-existing to generate significant viable phenotypic variation is entirely consistent with facilitated variation [60, ch. 7]. That the fitness curves are indicative of punctuated evolution also suggests some consistency of the model with natural evolution. While the similarities are extremely limited, as must be the case with any model of a complex process, it is important to draw comparisons where they may be insightful. In the following section, the properties of the model that are considered important for the search characteristics are discussed , and contrast made with some other EAs.

6.6 Discussion

The properties of EBDDIN that facilitates EMMF are important to appreciate. The *massive redundancy* of the OBDD representation facilitates a plethora neutral variants to explore. Without this redundancy, genotypes exhibiting favourable pleiotropic patterns would not occur so readily. The *neutral networks*

that connect all the genotypic representations of a given function facilitates exploration of that redundancy, allowing the favourable configurations to emerge by the reproductive advantage they impart. The neutral evolution of modularity towards favourable configurations is necessarily *gradual*, else favourable configurations would find difficulty accreting and be difficult to maintain within the population.

Another important property is variation in pleiotropy within the genotype; this was also recognised by Hansen [49] in his theoretical models. While the variation in pleiotropy is very evident within EBDDIN and easily reasoned to facilitate the patterns of pleiotropy that represent evolvability, it is no so apparent that similar variation in pleiotropy occurs in some other EAs. For example, take a typical tree GP [62] representation of a Boolean function. The inputs are situated at the terminals so the whole genotype may be involved in processing each and every fitness case. What results is 100% pleiotropy throughout the genotype, so there can be no variation in pleiotropy. As a result, a mutation at any location has the potential to change any or all of the fitness cases. Furthermore, 100% pleiotropy results in constant pleiotropic utility (U_p) throughout the genotype. So the problems evident with a normal distribution of U_p (figure 6.2) worsen because the distribution of U_p becomes a spike, reducing the overlap with the expected outcome of A1 even more quickly than a normal distribution.

This is clearly limiting for evolvability in GP and other genotypic representations that do not exhibit variation in pleiotropy within the genotype, though it does not deny that the variation component of evolvability cannot be achieved in other ways. One possibility is clear from figure 6.2: the probability of fitness improvement may be increased if the expected outcome of A1 mutation, (6.1), becomes distorted away from a normal distribution about $U_p = 0.5$. This may be facilitated, for example, by the proliferation of subfunctions which prove particular useful in evolution. This was alluded to towards the end of section 6.5 for EBDDIN, where it was suggested that genotypic features with high U_p may serve as good building blocks and explain the performance characteristics

depicted in figure 6.5. However, Altenberg [2] has also made similar arguments about blocks of code in GP which have high *constructional fitness*, having a higher probability than average of increasing the fitness of the genotypes they reside in. However, it is clear from figure 6.2 that EAs that can both exhibit variation in pleiotropy within the genotype and bias the expected outcome of mutation offer the greatest potential for achieving the variation component of evolvability.

It is beyond the scope of this thesis to investigate the potential to vary pleiotropy in the genotype in other EAs or problem domains to any depth. However, the absence of this property is likely to be significant in determining their performance characteristics or limitations thereof. On the other hand, designing EAs with such a property is a research direction that may prove extremely fruitful. This is an area that has received little attention in the EC literature to date, and it is hoped that this contribution of the thesis will spurn interest in that direction.

6.7 Summary

Notions of modularity were discussed, and it is was made clear that a notion of modularity that emphasises reuse and the removal of redundancy is employed for EBDDIN. Modularity is hypothesised to play a role in facilitating pleiotropic patterns that separate out and compress the genotypic representation of adapted traits, leaving maladapted traits exposed to perturbation. Experiments support the hypothesis. The properties of EBDDIN that are considered important were identified and discussed. In particular, the important role of variation in pleiotropy in the genotype was emphasised and argued to present a possible future direction for EC research aimed at understanding and enhancing evolvability.

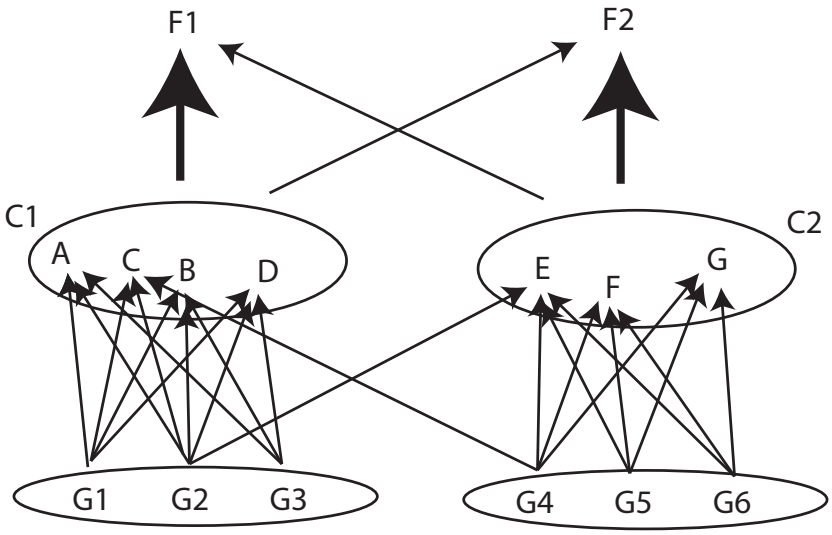


Figure 6.1: Wagner & Altenberg's [122] notion of evolutionary modularity. Each character complex, C1 and C2, serves some primary function, F1 and F2 respectively. Only weak influences exist of C1 on F2, and vice-versa. The genotypic representation is modular because the pleiotropic influences of each gene in a given gene group is primarily limited to a particular character complex.

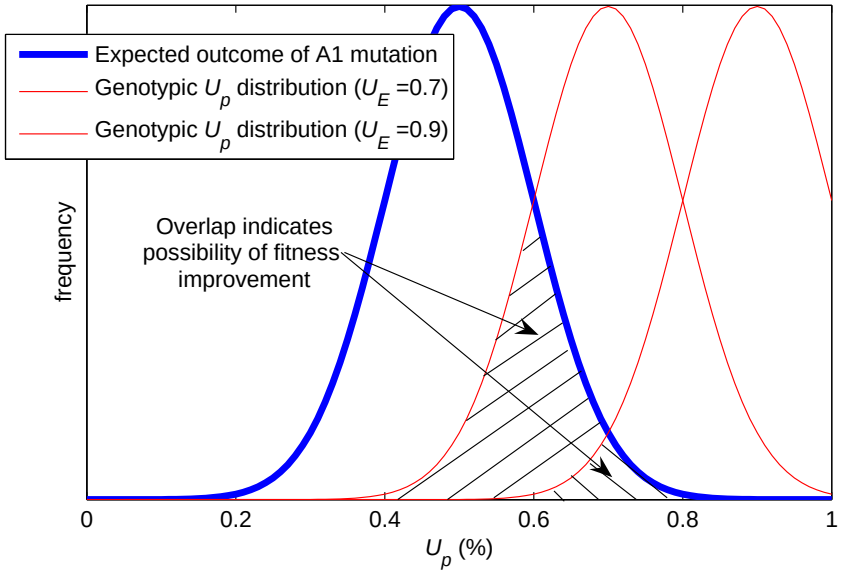


Figure 6.2: The problem of achieving fitness improving variation. (Heavy line) The outcome of A1 mutation for a given edge normally distributed about $U_p = 0.5$. (Thinner lines lines) U_p of edges within a genotype given by a normal distribution about U_E. The distributions depicted are not intended to be accurate representations of actual or theoretical distributions, but estimates which serve only to illustrate the significance of the relative shape and position of the distributions for effecting a fitness increase. At the start of a run, the two distributions overlap and fitness improvement is easily attainable. As fitness improves and U_E increases, however, the possibility of fitness improvement diminishes as a result of decreasing overlap (arrowed) between the distributions of expected outcome of A1 and genotypic U_p.
The figure highlights two ways in which increasing overlap and the probability of fitness improvement might come about. Firstly, a flattening or similar distortion of the genotypic U_p distribution away from the normal about U_E. Secondly, a distortion or shifting of the distribution representing the outcome of A1 mutation. Clearly, however, either of the distributions will become narrower with increasing pleiotropy, and the genotypic U_p distribution will become a spike when pleiotropy is 100%, stifling the possibility of fitness increase. Variation in pleiotropy, therefore, is of crucial importance.

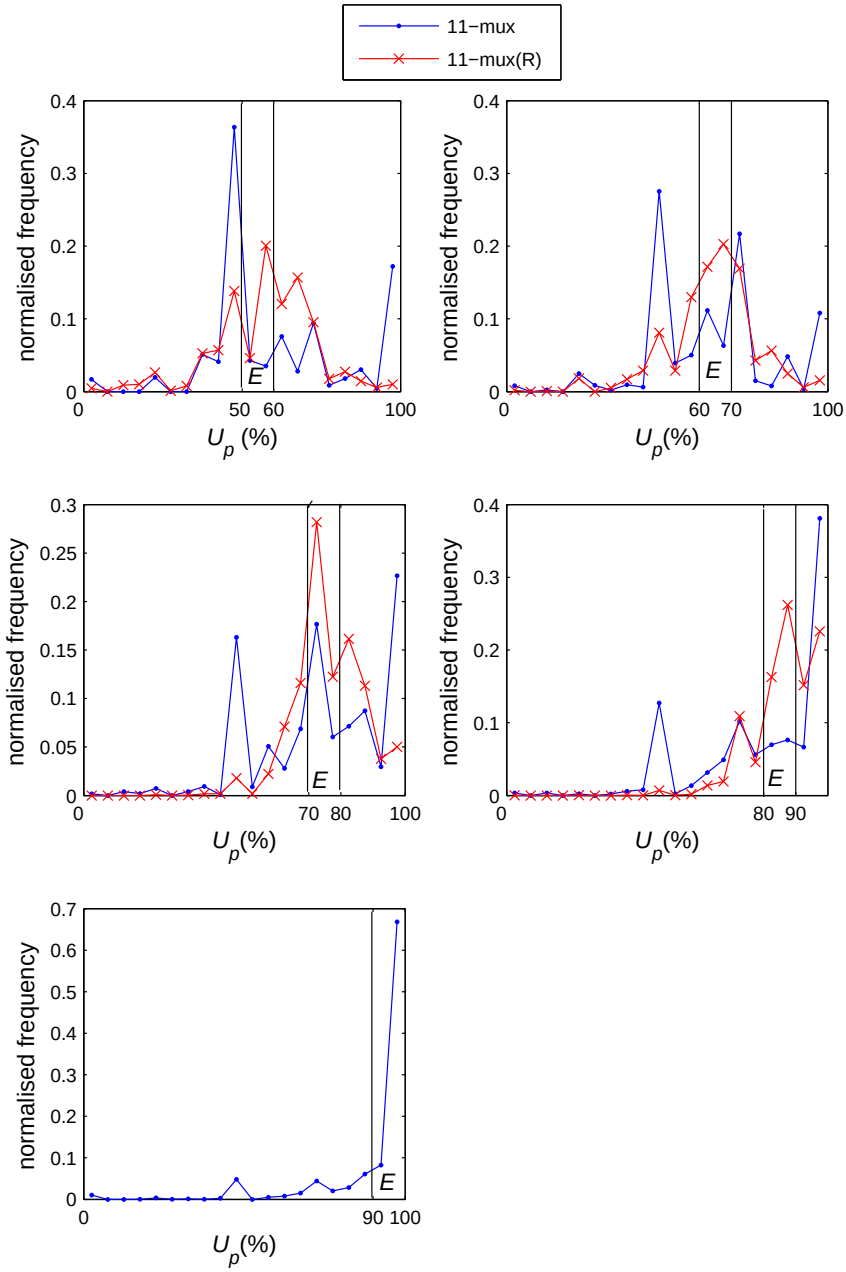


Figure 6.3: Frequency distributions of U_p for 11-mux and 11-mux(R). Each plot shows an interval of fitness as U_E. No plot is shown for 11-mux(R) in the interval 90-100% as fitness improvement stagnated here. The distribution of U_p for 11-mux(R) approximates U_E. The distribution of U_p for 11-mux, however, sees a distortion in the distribution in which the upper tail disappears completely and the lower tail is extended and exhibits a secondary peak.

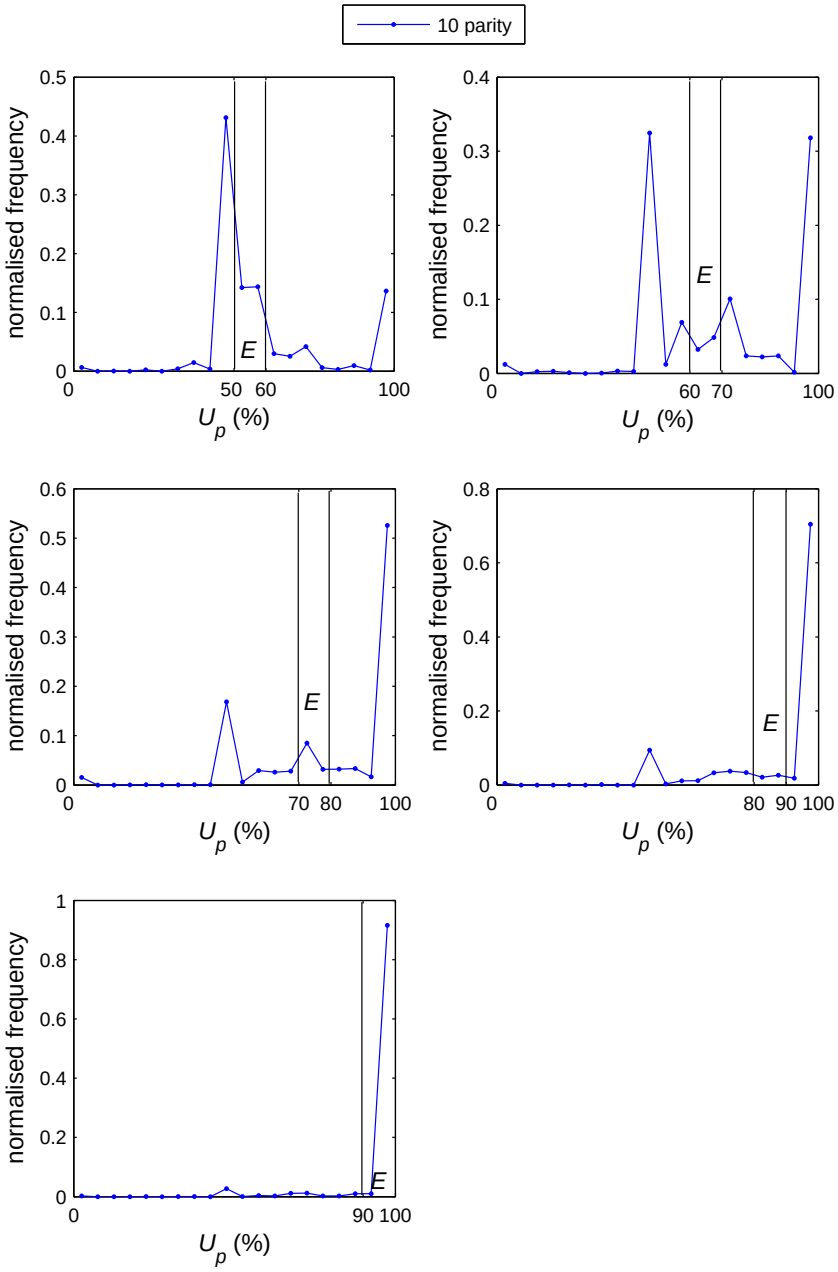


Figure 6.4: Genotypic U_p distributions for 10 parity.

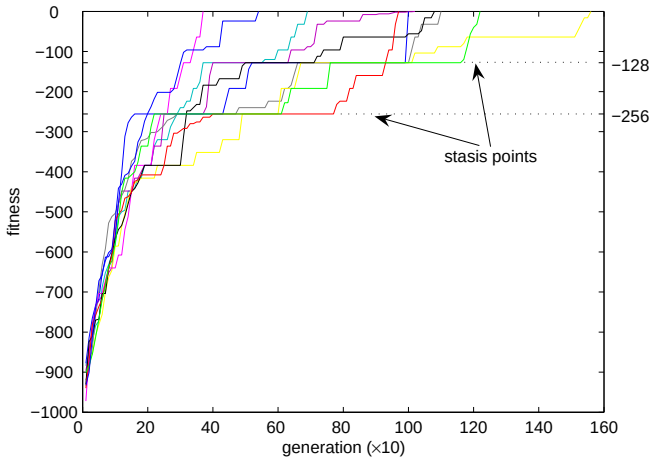


Figure 6.5: Fitness curves for 11-mux exhibiting punctuated equilibria-like characteristics. 10 curves are shown, most of which get exhibit periods of stasis at the points indicated. The stasis points are located at points 2^n, implying the absence of a higher-level functional component. All runs eventually return to rapid fitness improvement once the foundations of the absent component are discovered, exploiting lower-level components.

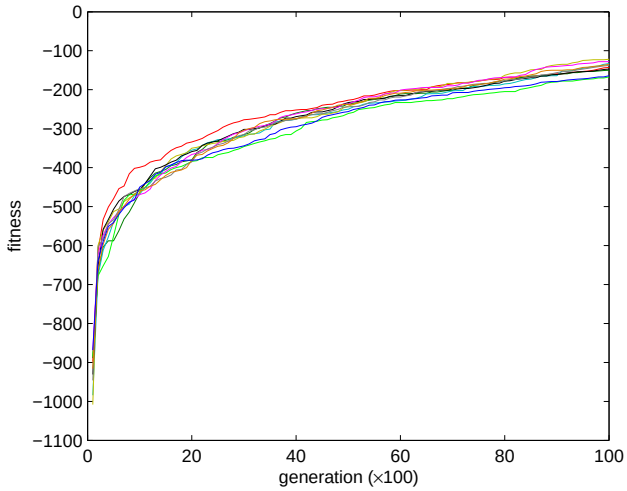


Figure 6.6: Fitness curves for 11-mux(R). 10 curves are shown. Fitness improvement is gradual with no significant periods of stasis. Lower-level components are not being exploited by higher-level components. These curves are not indicative of punctuated evolution.

Chapter 7

On gradualism and neutrality: encouraging exploration and exploitation simultaneously

EAs and other search algorithms are usually susceptible to becoming trapped in local optima. Attempts to address the problem are often framed in the context of needing to balance, or trade-off, exploitation against exploration. Ideally, it is best to maximise both simultaneously but this is usually seen as infeasible in the presence of multi-modal search spaces. This chapter investigates the potential for exploration of both neutrality and mutation rate within EBDDIN, and argues that the former is the more important.[1] The most interesting result, however, is that the necessity for a trade-off between exploitation and exploration can be avoided when the cost of evaluating neutral offspring can be circumvented.

[1] A shorter version of this chapter was presented as a paper at CEC 2006 [29].

7.1 Introduction

The assumed necessity for a balance between exploration and exploitation is highlighted by Michalwicz & Fogel [79, p.45]:

> "How can we design a search algorithm that has a chance to escape local optima, to balance exploration and exploitation, and to make the search independent of the initial configuration?"

In EC, exploitation is encouraged by strong, elitist selection and smaller population sizes, or by using lower mutation rates to promote stronger correlation between parent and offspring. Conversely, exploration is encouraged by promoting greater population diversity and weaker selection, or by increasing mutation rate and thereby lessening correlation between parent and offspring. Clearly, attempting to encourage both exploitation and exploration simultaneously using mutation rate or population size is antagonistic, hence the need for a trade-off or balance.

However, neutrality has also been identified as a source of exploration. For example, a number works on CGP [84] suggest that neutrality is an important source of exploration, and the best performance can often, though not always, be achieved with a low mutation rate [83, 130, 131, 132]. In contrast, Knowles & Watson [61] suggest caution about expecting neutrality to generally improve evolvability. They argue that other sources of exploration, such as an increased mutation rate, are potentially more appropriate. Thus, the relative potential for neutrality and mutation rate for exploration is the subject of some uncertainty, and that is the motivation for this chapter.

EBDDIN offers a unique perspective on the relative potential for neutrality and mutation rates for exploration because it circumvents the need to evaluate neutral offspring, providing cost-free exploration with respect to the Average Evaluations to a Solution (AES) [38] performance measure. A modified version of EBDDIN is also investigated which simulates the need to evaluate neutral offspring. This *impaired* EBDDIN aids in understanding the influence of neutrality

and mutation rates more generally, and serves to compare and contrast against the standard EBDDIN. The conclusions suggest a combination of gradualism (low mutation rate) and neutrality for superior performance in EBDDIN.

The chapter is set out as follows. The concepts of gradualism is reviewed in section 7.2. In section 7.3 a simple search space model is first presented, and the concept of *adequacy* introduced for reasoning about the potential for gradualism in a space with respect to modality. The resulting hypothesis that the greater this potential, the more evolvable is the search space, is presented in section 7.4. The hypothesis is tested experimentally in section 7.5.

7.2 Background

Gradualism

Gradualism postulates that differences between species arise gradually, in small steps, through a large number of intermediate forms. Darwin knew nothing of molecular genetics, and formulated gradualism in respect of phenotypic change. However, gradualism can be recognised at the molecular level also where mutation rates are known to be very low; it stands to reason that gradual phenotypic change is facilitated by gradual genetic change. Artificial evolution that is closely aligned with the principles of natural evolution might also exhibit good performance with relatively low mutation rates. In EC, however, more severe mutation is often considered beneficial, defying gradualism. For example, Knowles & Watson [61] find higher mutation rates preferential over neutrality for exploration, and Yao *et al.* [129] found beneficial the more severe mutation offered by the Cauchy distribution in evolutionary programming.

Gradualism, is not, as is sometimes suggested, contradictory to Gould's *punctuated equilibria* [46]: an apparently common misconception [77]. The relatively short periods of rapid change in between the long periods of stasis implied by Gould's theory are indeed rapid, but also gradual. The two theories are, in fact, compatible.

Gradualism and facilitation variation

The theory of facilitated variation [60] superficially appears inconsistent with gradualism. Facilitated variation postulates how minimal random genotypic mutation can generate significant and viable phenotypic variation: an extra mammalian digit or a second pair of wings on an insect, for example. Kirschner & Gerhart explain that the existing organism constrains how it responds to genotypic mutation, facilitating phenotypic variation. The apparent novelty of the insect's extra wings, or the mammals extra digit, is already built into the organism: random, minor genotypic mutation simply triggers it so it materialises in a differing context. Thus, if it is conceded that Darwin knew nothing of the genetic material and the nature of embryonic development, that random mutation occurred not directly on the phenotype, then facilitated variation and gradualism can be considered compatible. Kirschner states[2] that facilitated variation does not deny gradualism, or dispute that gradual change is predominant: it explains how complex transitions can occur that are non-lethal.

7.3 Search space model

The section presents a search space model that will be used to reason about the potential for gradualism with respect to modality. The term *search space* is preferred over Wright's [127] *landscape* metaphor, which has received some criticism in both evolutionary biology [91] and computation [57], and was discussed in section 3.3. The model is based on set relationships, and is comparatively simple compared to other non-metric models [57][113]. It is used here to hypothesise about the effects of varying the mutation rate (the bound) and degree of neutral drift. The probabilities associated with moving between points in the space are not represented directly in the model, but are considered only relatively with spaces that are constructed differently.

The concept of neighbourhood is first introduced. The search space model,

[2]Personal communication 20/9/2007.

S, is then defined as mutation relation on the set of genotypes, G. Finally, the notion of search space *adequacy* is introduced for reasoning about the potential for gradualism in a space with respect to modality. Clarifying arguments then conclude the section.

Neighbourhoods

The *neighbourhood* is a function, $N_M^b : G \rightarrow \mathcal{P}(G)$, where $\mathcal{P}(G)$ is the power-set of G, capturing the notion of which genotypes are accessible with a given mutation bound, b, and mutation set, M:

$$N_M^b(g) = \{g' \in \mathcal{G} | g' \text{ is accessible from } g \text{ with } b \text{ mutations from } M\} \quad (7.1)$$

The *immediate neighbourhood* (or 1-mutant neighbourhood) of $g \in G$ is $N_M^1(g)$, makes g's neighbours accessible with one mutation. Similarly, $N_M^2(g)$ is the 2-mutant neighbourhood, etc. Larger neighbourhoods are built on immediate neighbourhoods, thus:

$$N_M^{i+1}(g) = \bigcup_{g' \in N_M^i(g)} N_M^1(g') \quad (7.2)$$

It can be said that $N_M^i(g)$ is more *selective* than $N_M^{i+1}(g)$ because $N_M^i(g) \subseteq N_M^{i+1}(g)$. Furthermore, $|N_M^{i+1}(g)|$, for small b, can be much larger than $|N_M^i(g)|$ because each member of $N_M^i(g)$ potentially has an immediate neighbourhood size similar to the set from which it comes. The *extended neighbourhood* of $g \in G$, denoted $X_M^b(g)$, incorporates all the neighbourhoods of g's neutral network:

$$X_M^b(g) = \bigcup_{g \in G | g' \equiv g} N_M^b(g') \quad (7.3)$$

where $g \equiv g'$ means that both g and g' represent the same phenotype (i.e. the same function). In OBDD genotype space, all phenotypically equivalent genotypes form a single neutral network connected by the aforementioned neu-

tral mutations. $|X_M^b(g)|$ will be larger where g has more phenotypic equivalents. Functions with more compact ROBDD representations typically have more functionally equivalent OBDDs than functions with less compact ROBDDs. Thus, $|X_M^b(g)|$ is typically larger where g represents a function with a compact ROBDD representation.

The search space as a mutation relation

The search space has the form $S_{M^b} \subseteq G \times G$, relating genotypes to genotypes in respect of all neighbourhoods for a given b and M.

$$S_{M^b} = \bigcup_{g \in G} \bigcup_{g' \in N_M^b(g)} \{(g, g')\} \tag{7.4}$$

Increasing b incorporates more severe mutational transitions (through less selective neighbourhoods) into S.[3] Decreasing b has the opposite effect, restricting the severity of mutational transitions in S. It is said that S_{M^i} is more *gradual* than $S_{M^{i+1}}$ due to the fact that $S_{M^i} \subseteq S_{M^{i+1}}$ and $S_{M^{i+1}} \setminus S_{M^i}$ contains those genotypes only accessible by more severe mutation. Thus, S can be made more or less gradual by construction using the mutation bound b. On its own, however, the degree of gradualism in S is of little use; the modality of S must also be considered. In this thesis, a local optima is defined as:

$$\forall g' \in X_M^b(g), fitness(g') \leq fitness(g) \tag{7.5}$$

where g is suboptimal. It is clear that a less gradual S is less likely to incorporate local optima because $X_M^b \supseteq X_M^{b-1}$.

Adequacy

The notion of *adequacy* provides a way to reason about the potential for gradualism in a search space with respect to modality. Gradualism has limited potential in a multi-modal space because the absence of large transitions in S

[3]Subscript and superscripts may be omitted where no meaning is lost.

prohibits the possibility of escape from, what would otherwise be, local optima. Therefore, for gradualism to exhibit its full potential, it must be provided in a search space free of local optima for a minimal mutation bound.

S is said to be *adequate*, denoted S^A, if it provides a search space free of local optima for a given problem with a given b and M. That is, for all $g_{origin} \in \mathcal{G}$, there is a series of transitional elements in S:

$$(g_{origin}, g_i), (g_i, g_j), \ldots, (g_{target-1}, g_{target}) \tag{7.6}$$

where g_{target} represents the global optimum.

S is said to be *completely adequate*, denoted S^C, if $S = S^A_{M^1}$ (i.e. $b = 1$). S is *universally completely adequate*, denoted S^U, if it is completely adequate for all problems within a specified domain.

S^A is easy to achieve, simply by increasing the mutation bound to approximate random search, eliminating local optima. However, high mutation rates defy gradualism and deny heritability, and are not conducive to evolvability. A more useful property is *complete adequacy*, S^C. While S^C suggests maximum adherence to the concept of gradualism, S^A does not. This suggests an ordering of decreasing adequacy based on increasing mutation bound whilst maintaining a local optima free S – the *adequacy ordering*:

$$S^C_M, S^A_{M^2}, \ldots, S^A_{M^\infty} \tag{7.7}$$

Adequacy, whether complete or not, does not necessarily imply optimality, hence its name. Adequacy simply provides a way to reason about the relative potential for gradualism within S whilst maintaining equality in respect of modality and the mutation set.

Adequacy for EBDDIN

$\{$**N1'**, **N2'**, **A1**$\}$ is the minimal set that will produce S^U. This follows from the fact that these are the only mutations required by theorem 1 (p77) to prove

that the search space is free of local optima for the specified problem domain. Removing any of these mutations means that an initial population that does not contain an optimal genotype cannot be transformed to one that does.

Also, $S_{\{N1,N1',N2,N2',A1\}^1}$ satisfies the requirements for S^U in the context of the fitness functions covered by theorem 1 (p77). Adding mutations to the minimal set does nothing to improve or worsen the adequacy of S, as the adequacy ordering relates only spaces with identical M. However, adding mutations may impact on the performance of the algorithm. In this case, the addition of the reducing neutral mutations **N1** and **N2**, encourage parsimony and modularity in the genotype.

Performance can also be influenced, without affecting adequacy, by biasing mutation selection, and thereby influencing the ratio of adaptive to neutral mutations. This change is not reflected in S directly because elements of S remain unchanged as a result of biasing mutation selection. However, the probability of choosing elements in S corresponding to neutral mutations changes. This allows the influence of differing biases, and therefore differing degrees of neutrality, to be assessed comparatively while maintaining the degree of adequacy.

Why prefer a more gradual S?

Assuming no knowledge of the vicinity of improving genotypes, any attempt at choosing and applying a single application of a mutation operator during offspring generation will have certain probabilities associated with it being deleterious or advantageous (or neutral). As a run progresses towards optimal fitness this probability distribution changes and deleterious applications of a mutation operator become increasingly likely. Furthermore, the degree to which an application of an operator can be deleterious to fitness will increase while the degree to which it can be advantageous will diminish. For higher mutation bounds, therefore, an improving early intermediate application of a variation operator during offspring generation is increasingly likely undone by a later application. Similarly, an improving later intermediate application is likely preempted by an

earlier deleterious one. As a result, for higher mutation bounds, it is increasingly likely that a deleterious mutant offspring will occur.

The counter argument to this thinking is that a smaller neighbourhood constrains the sampling area, restricting exploration. Thus, it is argued that a larger neighbourhood is necessary to defy this containment. However, if there is a reasonable degree of neutral drift, the population is constantly changing, as too are the population neighbourhoods. The problem of restriction of exploration through small neighbourhood containment is, therefore, nullified by neutral drift. The search can remain focussed on the 1-mutant neighbourhood of the neutrally evolving population, where the likelihood of an improving intermediate mutation being preempted or undone is absent and heritability and exploitation is relatively high.

A further negative argument regarding neutrality equates many neutral steps with one large mutation[4]:

> "Personally, I regard "neutral steps that require no fitness evaluation" as one large mutation. There is no surprise that large mutations help in neutral networks because that's a common strategy. In fact, this is similar to search on a step function with many plateaus."

Yao cites [129] as showing this empirically. However, a large mutation is clearly distinct from several neutral steps followed by a non-neutral step. While a large mutation will have both a large genotypic effect and a large phenotypic effect, a series of small neutral mutations followed by a small non-neutral mutation will have a large genotypic effect, but a relatively *small* phenotypic effect. A smaller phenotypic perturbation clearly has a greater likelihood of producing a fitness improvement than a larger one. Thus, a series of neutral steps followed by a non-neutral step must be preferred over a larger mutation where local optima are alleviated by neutrality.

[4]Personal communication dated 18/2/2005 following a review of a paper [27] associated with this thesis.

Model summary

The model and algorithm facilitates construction of search spaces with three properties primary interest:

1. Freedom from local optima, regardless of mutation bound (theorem 1).
2. The ability to vary the potential for gradualism in S via the mutation bound (adequacy).
3. The ability to vary the degree of neutrality in S by biasing mutation selection, without consequence to adequacy.

These properties can now be studied experimentally, and questions regarding the relevance of, and relationship between, mutation rate and neutrality addressed. Moreover, these properties can be studied in the contrasting contexts of the standard EBDDIN, in which neutrality is cost free, and the impaired EBDDIN, in which it is not.

7.4 The adequacy hypothesis

Before studying the aforementioned properties experimentally, a hypothesis is first presented regarding the expected results. Standard EBDDIN is able to circumvent evaluation of neutral offspring, so the cost of neutral drift is circumvented. Thus, neutrality-induced exploration should always be preferred over a mutation-induced exploration. The hypothesis, then, suggests increased adequacy results in increased evolvability: maximal adequacy provides maximum evolvability. More formally:

$$AES(S^C_M) \lessapprox AES(S^A_{M^2}) \lessapprox \ldots \lessapprox AES(S^A_{M^\infty}) \tag{7.8}$$

where AES is the performance measure (smaller is better). The prediction is expected to hold for any bias that is introduced for mutation selection, affecting the ratio of adaptive to neutral mutations but not the adequacy. Any increase

in neutral drift due to the lowering of this ratio only provides an increase in exploration and it should do so without negative consequence to exploitation.

Impaired EBDDIN

What now happens if EBDDIN is modified, to its detriment, so it is forced to evaluate neutral offspring?[5] Neutrality is no longer cost free, and increasing neutrality-induced exploration will have a negative impact on exploitation. Too much neutral drift and exploitation is neglected in favour of evaluating excessive neutral offspring. Conversely, too little neutral drift and exploration is neglected, constraining the search to a stagnating neighbourhood. Because increasing mutation bound also stifles neutral drift, a trough in the graph of AES against mutation bound is expected to form in the former case, and a steepening slope in the latter. However, the degree of neutrality can be biased through mutation selection, independently of adequacy, altering the ratio of adaptive to neutral mutations. This is expected to remove the trough where there is excessive neutral drift, and lessen the slope where there is too little neutral drift, producing a shallowing curve whose tangent approaches the horizontal at the vertical axes. An optimal ratio should be reachable where the optimal performance over all mutation bounds resides at the minimal. Thus, the hypothesis assumes an optimal ratio of adaptive to neutral mutations for the impaired EBDDIN, which must be found by directed trial and error. Finding this ratio is *not* necessary for the standard EBDDIN.

7.5 Experiments

For the following experiments a tournament selection with population size = 2 is employed; this nullifies potential for exploration through population diversity, but facilitates maximum exploitation in terms of always breeding the fittest phenotype seen so far. Figure 7.1 shows how performance, in terms of variance and

[5]This impaired EBDDIN is only required to evaluate neutral and potentially adaptive offspring, not clones.

mean run length, is influenced for varying mutation bound. Figures 7.2 and 7.3 examine the effects of introducing a bias to mutation selection for the standard and impaired versions of EBDDIN respectively. The heavier lines indicate a higher ratio of adaptive to neutral mutations (lower neutrality), and vice versa. The juxtaposition of figures 7.2 and 7.3 aid in contrasting the standard and impaired versions of EBDDIN - the former having the advantage of circumventing the cost of evaluating neutral offspring. The results are analysed and further discussed in the remainder of this paper.

Analysis

As can be seen from figure 7.1, increasing mutation bound typically results in a decrease in performance in terms of both run length and variance. Figure 7.2 examines the effects of introducing a bias to mutation selection for the standard EBDDIN. The behaviour is largely as predicted. Increasing the ratio of adaptive to neutral mutations, stifling neutral drift, typically results in an increase in AES (lesser performance). Conversely, lowering the ratio increases neutral drift, and thus exploration, resulting in lower AES (better performance). Though the behaviour exhibited 7.2(c) is borderline in respect of the expected behaviour regarding bias, the results are consistent with the hypothesis. Also note that AES values for each problem at mutation bound = 1 are similarly good for all bias values, indicating that performance is extremely robust to the bias at the minimal mutation bound.

Figure 7.3 examines the effects of introducing a bias to mutation selection for the impaired EBDDIN. The results are not as predicted, and three very different behaviours are exhibited. Firstly, excessive neutral drift can be observed where a trough is formed; in both figures 7.3(a) and 7.3(b) the trough is formed without bias, but 7.3(c) demonstrates how such a trough can be introduced with a lower bias. Removing the trough, however, is problematic. Where a trough exists, reducing neutral drift is expected to shift the trough toward the vertical axes, improving on the overall optimal as it goes. This is indeed the behaviour

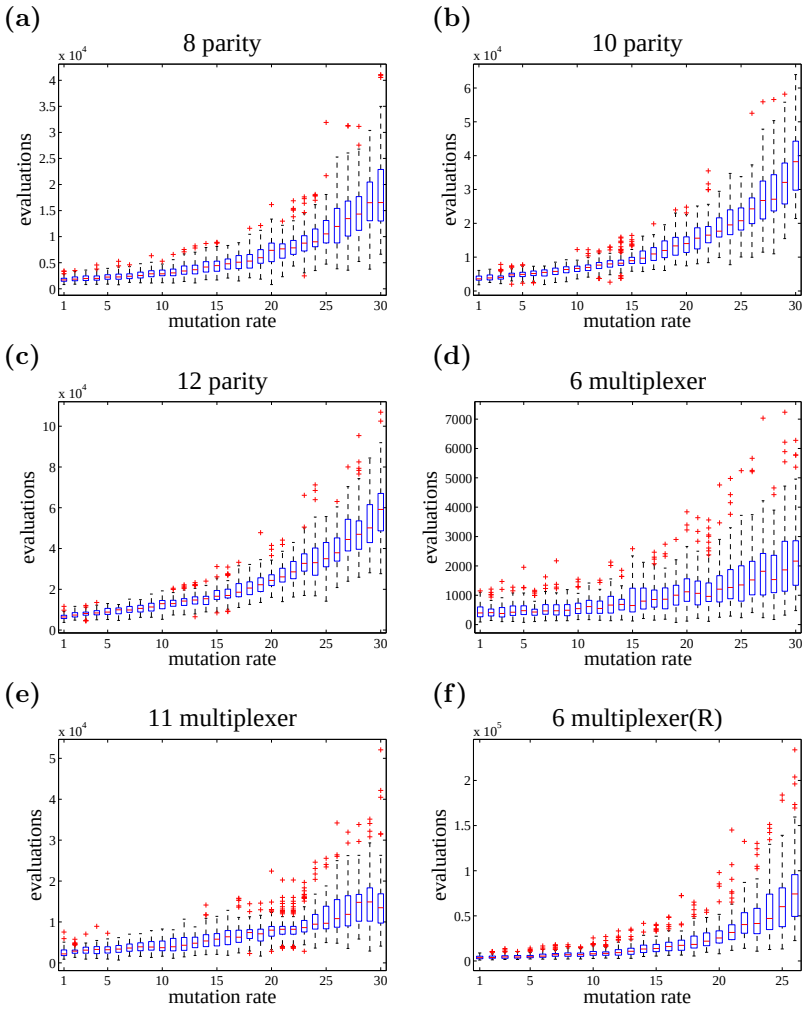


Figure 7.1: Box and whisker plots for standard EBDDIN (100 runs). (a)-(c) parity problems. (d) and (e) multiplexer problems with optimal variable orderings. (f) is 6 multiplexer with reverse variable ordering (see [32]). For all problems, optimal performance is achieved at minimal mutation bound. AES and variance increase rapidly with mutation bound. These results are consistent with the hypothesis.

exhibited in both 7.3(c) and, to a slightly lesser extent, 7.3(b). However, in contrast, it is not the behaviour exhibited in 7.3(a); while the trough does indeed move toward the axes as the adaptive to neutral ratio is increased, performance deteriorates universally across all mutation bounds. Secondly, where no trough exists, decreasing neutrality is expected to deteriorate performance universally, as for the standard EBDDIN; this is because there is no excessive neutrality and exploration is stifled further. This is indeed the behaviour exhibited in 7.3(b), but is not the behaviour exhibited in 7.3(c), which actually sees an improvement in performance at lower mutation bounds. Only (b) of figure 7.3 exhibits the kind of performance characteristics predicted.

Thus, the hypothesis does not generally hold for the impaired EBDDIN, the relationship between neutrality and mutation rate apparently being complex and problem-dependent. For the standard EBDDIN, however, the relationship is a trivial one; for all the varying degrees of neutrality tested, lowering the mutation bound improves performance. In addition, in contrast to the standard EBDDIN, performance for the impaired EBDDIN is no longer necessarily robust to the bias at minimal mutation bound.

For the standard EBDDIN, two important things can be concluded. Firstly, for fitness functions covered by theorem 1, a mutation bound of 1, the minimal, provides a high degree of confidence for producing superior performance over all other mutation bounds. This confidence follows from the argument presented in section 7.3, which itself is based on gradualism, and supported experimentally in section 7.5. Secondly, having maximised exploitation with minimal mutation bound and minimal population diversity, exploration can be increased without cost through mutation selection bias, confidently improving, or at least maintaining, performance. Thus, exploitation and exploration are maximised simultaneously, and the need for a trade-off effectively alleviated. In theory, exploration can be increased indefinitely.[6] Furthermore, these findings also sug-

[6]In practice, other costs to that covered by AES will limit the degree of neutrality induced exploration. The informed reader will acknowledge, however, that the cost of fitness evaluation is the predominate one.

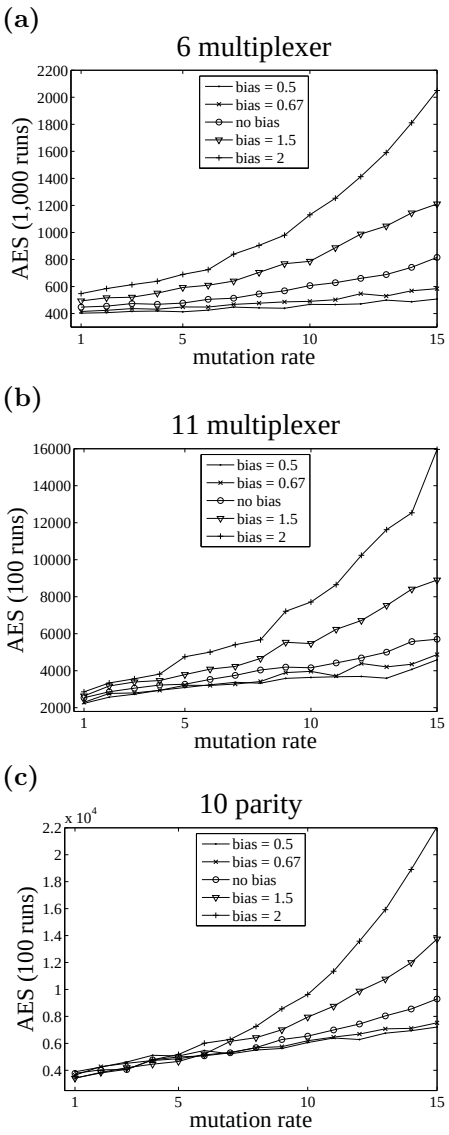


Figure 7.2: The effects of biasing mutation selection for EBDDIN. The probability of selecting an adaptive mutation is multiplied by the bias. Thicker lines indicate a higher ratio of adaptive to neutral mutations (less neutrality). Decreasing the adaptive to neutral ratio through a <1 bias (more neutrality) results in an overall increase in performance for all mutation bounds; conversely, increasing the ratio decreases performance. Increasing mutation bound typically results in increased AES, regardless of bias, as predicted. These results are consistent with the hypothesis.

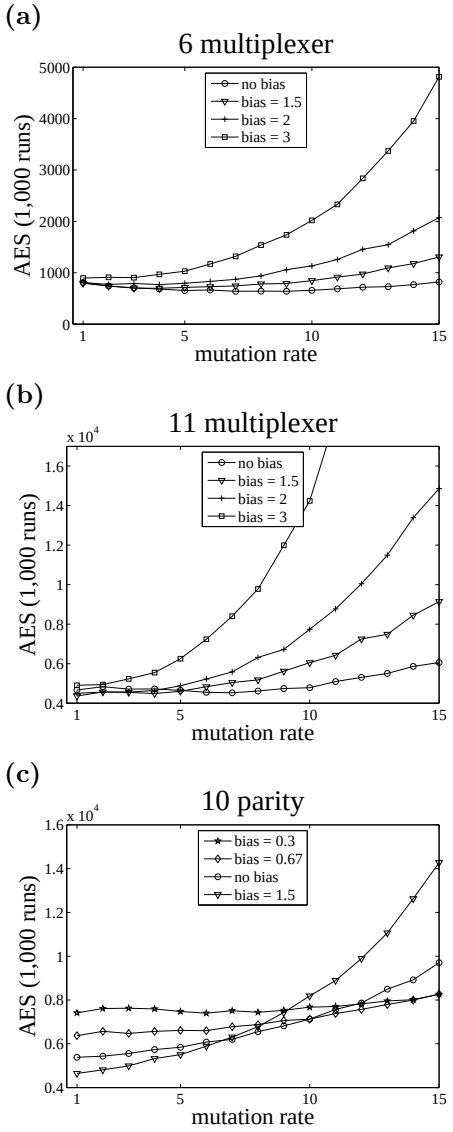


Figure 7.3: The effects of biasing mutation selection for the *impaired* EBDDIN. (a) No bias results in the optimal residing away from the minimal mutation bound, forming a trough. Increasing the bias shifts the optimal towards the vertical axes, eliminating the trough, but fails to improve overall AES. (b) For no bias, a trough is formed. Increasing the bias shifts the optimal towards the axes, but this time improves overall optimal AES, up to a point. (c) For no bias, no trough is formed, indicating no excessive neutral drift. Introducing more neutral drift increases AES, introducing a trough. Reducing neutral drift unexpectedly improves overall optimal AES.

gest a positive correlation between robustness and evolvability, the significance of which was discussed in section 3.1.3. Even with the minimal population size employed for the above experiments, the increased robustness resulting from increased neutrality results in increased evolvability.

At this point, a reader might ask how to choose the degree of neutrality through the mutation selection bias. However, for a mutation bound of 1, there is clearly significant license granted for setting this bias, as can be observed in figure 7.2. Regardless of the bias, AES values for the minimal mutation bound are clumped tightly together, AES becoming dispersed only for higher mutation bounds. It has already been established that the minimal mutation bound will always provide the best performance, so the choice of bias is somewhat arbitrary.

The picture is less clear for the impaired EBDDIN. This is of no significance for the standard EBDDIN, but aids in obtaining an understanding of the relationship between neutrality and mutation rate in the general case. Though the results suggest that neutrality remains an important source of exploration, the relationship between neutrality and mutation bound is complex and problem dependent, the consequence of varying either being largely unpredictable. This only emphasises the benefits of standard EBDDIN's capability to circumvent evaluation of neutral offspring.

With regard to which is the better source of exploration, neutrality is the clear winner where the cost of evaluating neutral offspring can be circumvented. In the general case, however, where neutral offspring must be evaluated, the answer is less certain. As demonstrated in figure 7.3(a), overall optimal performance can sometimes be achieved with a larger mutation bound, defying the adequacy hypothesis. However, what is telling from figure 7.3 is that, where there is minimal mutation rate induced exploration (i.e. mutation bound = 1), good performance can still be achieved with neutrality induced exploration only. Conversely, good results *cannot* be achieved with mutation bound induced exploration only; as explained in section 7.3, the neutral mutations are a necessity for removing local optima. It is also clear from figures 7.2 and 7.3 that decreas-

ing neutrality has an increasingly detrimental consequence at higher mutation bounds. Thus, neutrality is the more important source of exploration for the impaired EBDDIN also.

7.6 Discussion

These results contradict the findings of Knowles & Watson [61], and finds that neutrality is the better source of exploration over increased mutation rate. It is important to try and understand the contradictory results, and differences in the representations and nature of redundancy are likely to be significant here. In EBDDIN, the redundancy has been shown to be organised into neutral networks so large and pervasive that local optima are alleviated completely. However, the redundancy of the Random Boolean Network representation used by Knowles & Watson is of the type which Rothlauf & Goldberg [97] describe as non-synonymously redundant. That is, the redundant genotypic variants of a function are not connected, so the potential for neutral walk is inhibited and local optima are not alleviated. The result is that higher mutation rates are required to escape local optima, defying heritability. That is, the Random Boolean Network representation resulted in a search space that lacked the *adequacy* property that was defined in this chapter. EBDDIN's search space, in contrast, was shown to be *completely adequate*.

A minimal population size was used in this chapter to eliminate exploration through population diversity, always exploiting the fittest parents seen so far. Experiments with larger populations resulted in a drop in performance due to reduced exploitation. However, this is not to say that population diversity cannot be important, and it is not enough to avoid the trade-off between exploration and exploitation. Not only is it important to maintain and breed the fittest seen so far while exploring for higher fitness, the population must explore and maintain evolvability: the *potential* to acquire increased fitness. In chapter 9, larger populations are shown important for the evolution of evolvability, and

exploitation is maintained there using a greedy form of steady-state selection, in which only the fittest phenotypes are bred.

This work has not addressed needle-in-haystack, deceptive, or long-path search spaces (or landscapes). Admittedly, these types of 'landscapes' are of some academic interest. However, such landscape are often considered independently of the operators applied (see Jones [57] for a discussion). For example, some researchers contrive a troublesome landscape for a hill-climber simply for the purpose of showing how a GA crossover, or such, is able to do better. The present author takes the view such approaches have limited value. Moreover, this chapter's approach was to make the search space navigable by construction, for which the idea of contriving difficult to navigate landscapes is antipodean. For this purpose, a search space model based on accessibility, that is not independent of the variation operators applied, can be more useful than the conventional landscape depiction of a search space.

7.7 Summary

This chapter investigated the potential for exploration from neutrality as opposed to exploration from higher mutation rates. A search space model based on set relationships was introduced to reason about search in a search space in terms of accessibility. The notion of *adequacy* was introduced to reason about the potential for gradualism in a space with respect to modality. Using two versions of EBDDIN, one in which the cost of neutral offspring could be circumvented, and one in which it could not, experiments were conducted which showed that neutrality was a better source of exploration than higher mutation rates. Where the cost of evaluating neutral offspring could be avoided, the balance between exploration and exploitation was shown to be avoidable. The findings also suggest a positive correlation between robustness and evolvability.

Chapter 8

Emergent π and the evolution of evolvability

BDDs have become the data structure of choice for representing discrete functions in many applications. However, BDDs are not a common representation within EC for Boolean functions or other domains that might be encoded in BDD form. The hitherto difficulties in designing effective methods to evolve BDDs combined with the variable ordering (π) problem poses a significant challenge which is yet to be overcome.[1] This chapter addresses this challenge and extends the EBDDIN approach to exhibit good variable orderings as an emergent property.[2] Evolvability is shown to be correlated with the quality of π, and the emergence of good π reasoned to demonstrate the evolution of evolvability in a static environment, the viability of which is disputed. A maintained and sometimes accelerating pace of evolution is demonstrated as a result.

[1] These limitations were discussed in chapter 4.

[2] A shorter version of this chapter was presented as a paper at PPSN X [28].

8.1 Introduction

The variable ordering problem

The variable ordering problem is prominent for all BDD applications. If a good variable ordering can be found, the BDD representation of a function will often be simple and efficient to manipulate [17]. However, the variable ordering problem is NP-complete in both optimal and approximate solutions [14, 107]. Furthermore, Krause [65] has argued theoretically that synthesising even an approximating function in the BDD representation is hard, and further suggested that the variable ordering must be optimised during the synthesis procedure.

The extended version of EBDDIN presented in this chapter optimises the variable ordering alongside function. It is elegant in its construction and can exhibit near optimal orderings as an emergent property. Most of the previous approaches to evolving BDDs have employed only a static variable ordering and have therefore been limited to functions for which a good variable ordering is known in advance [27, 34, 103, 118, 128]. For most practical applications, however, good variable orderings cannot be known in advance so the variable ordering must be optimised along with functional fitness. Only Droste [35] has attempted to address this previously with a distributed hybrid approach, combining his earlier BDD-based GP with existing heuristics for variable reordering.

The evolution of evolvability

The evolution of evolvability is a phrase coined by Dawkins [24] for the postulated phenomenon that evolvability[3], the capacity to evolve, is a selectable trait and so is itself subject to the evolutionary forces of variation, selection and the resulting adaptation.[4] Selection for evolvability, however, is not direct; evolvability is selected for indirectly by what Dawkins refers to as *second-order selection*. That is, individuals with good evolvability traits are not selected for

[3]The reference here is to the variation component of evolvability.
[4]These and related issues were introduced and discussed thoroughly in section 3.1.

on the basis of of their ability to survive and reproduce, but by the fact that the offspring of individuals with good evolvability traits will typically be fitter than individuals without, regardless of the fitness of the parents.

Much of the literature on the subject suggests that the evolution of evolvability necessitates a dynamic environment. Both Lipson [74] and Reisenger [94] have argued that for evolvability to evolve, a dynamic environment is necessary. Turney's [117] model of the evolution of evolvability also depends on a highly dynamic fitness function. This chapter employs EBDDIN as a computational model of the evolution of evolvability within the context of a static environment, demonstrating that a dynamic fitness function is no necessary to facilitate the evolution of evolvability.

In chapter 6 the variation component of evolvability was shown to be represented within EBDDIN as pleiotropic patterns that help to align variation with fitness. However, while the patterns that emerged could be easily reasoned to facilitate evolvability, the depiction of the evolution of such patterns was less than lucid. In this chapter, a more lucid depiction of the evolution of evolvability is presented in terms of emergent variable orderings (π). Better π better facilitate favourable pleiotropic patterns (section 6.5), so evolvability is correlated with π. It is by observing how π evolves that provides a lucid picture of how evolvability is evolving.

Section 8.2 extends EBDDIN to include dynamic variable reordering. Section 8.3 shows how evolvability is correlated with the quality of π when π is fixed in EBDDIN. Section 8.4 looks at the emergence of good π when EBDDIN is extended to exploit dynamic variable reordering. Discussion is then presented in section 8.5 regarding the implications.

8.2 Evolving the BDD variable ordering

So far in this thesis, the variable ordering (π) of the population has been static: each individual has the same π which is unchanging. The significance and

difficulties of choosing a good π were discussed in section 2.3 and again in the introduction to this chapter. Therefore, it is clear that any effective approach to synthesizing functions in the BDD representation must also facilitate dynamic variable reordering. This section extends EBDDIN to facilitate dynamic π.

All heuristic approaches to dynamic variable ordering are built on the procedure for swapping adjacent variables without affecting function. Swapping non-adjacent variables and other manipulations of the variable ordering is achieved by repeatedly swapping adjacent variables to achieve the desired manipulation. Rudell [98] describes an efficient implementation method for swapping adjacent variables. Variable swapping has time complexity proportional to the number of nodes associated with the two adjacent variables, so can generally be done in reasonable time.

To extend EBDDIN to facilitate dynamic variable reordering, the N3 mutation is introduced. N3 is simply Rudell's procedure for swapping adjacent variables in the ordering while maintaining function: it is a neutral mutation. Figure 8.1 shows the basic procedure.

N3 is the most simple of variable reordering operators. Other operators can be built on top of N3. A jump mutation would see a variable jump from one position to another, possibly quite distant. Given that the variable ordering is just a permutation, any established permutation operator could be implemented under EBDDIN. However, significant changes to the variable ordering can be costly and unpredictable. Moving variables too much is computationally costly in terms of time, and an OBDD can grow in complexity exponentially. Furthermore, swapping variables at several levels is lightly to greater disrupt any emergent bias pleiotropy that was shown significant for the variational component evolvability in chapter 6. For these reasons only the N3 mutation is employed for mutating the variable ordering, and this suffices for the purposes of this chapter.

8.3 Evolvability and π

The aim of this section is to reinforce the relationship between evolvability and the ROBDD complexities[5] induced by π. The complexity of ROBDD induced by a given π for a given problem is referred to as the *Implied Solution Complexity* (ISC) of π, or of an π-(R)OBDD. For example, the ISC of $\pi = [0, 1, 2, 3, 4, 5]$ for the 6-bit multiplexer problem (6-mux) is 7; this is the number of nonterminals in the $[0, 1, 2, 3, 4, 5]$-ROBDD solution to 6-mux. The reverse ordering has an ISC of 29. Similarly, any $[0, 1, 2, 3, 4, 5]$-(R)OBDD has an ISC of 7 for 6-mux, regardless of its actual fitness for 6-mux.

So, the objective here is to investigate how differing π, categorised by their ISC values, influence evolvability. To achieve this, EBDDIN is run *without* dynamic variable ordering (i.e., no N3 mutations) for selected ISC categories using a $(1 + 5)$ ES. The Average Evaluations to a Solution (AES) performance measure is then taken as an indication of the degree of evolvability associated with each ISC category. The actual π under each ISC category are generated randomly. The results are plotted in figure 8.2. As can be seen clearly, for all problems tested, the trend associated with increasing ISC is increasing AES (poorer evolvability)[6]. Furthermore, the trend of increasing AES is greater than linear in ISC, and appears to be approaching exponential. What is concluded from these results is that better π, that is, π having lower ISC values, are associated with much greater evolvability.

Knowing that evolvability is associated with low ISC values, however, appears of little use if there is no prior knowledge about which π have low ISC values. For functions such as the multiplexer and adder, optimal π are well-known so a good π can be fixed in advance of running the EA. However, in general, it is not possible to tell in advance which π have low ISC.

[5]Section 2.3 defined ROBDD *complexity* as the number of unique subfunctions (or nonterminals) present. In contrast, an unreduced OBDD has *size* rather than complexity because nonterminals may represent subfunctions which are not unique.

[6]Higher values of $\frac{1}{AES}$ indicates better evolvability of the system.

8.4 Emergent π

This section argues that good π are an emergent property of the extended EBDDIN with dynamic variable ordering. What is meant by 'emergence' in this respect is that there is no explicitly introduced incentive in EBDDIN for inducing individuals with below average ISC value. That is, there is no aspect of the fitness function, secondary size-related fitness objective, or mutation-related incentive that explicitly encourages propagation of π with low ISC. Indeed, only the N3 mutation, the swapping of adjacent variables, can influence ISC directly, and the location point for N3 in the genotype is always chosen randomly by variable. Good π arise solely as the logical consequence of being associated with subspaces of genotype space that are more evolvable. That is, individuals possessing π with lower ISC values are more successful reproductively, and so propagate their favourable evolvability characteristics.

The problems investigated here are the 11-mux (11 inputs, 1 output), 20-mux (20 inputs, 1 ouput) and the 4-bit adder with carry out (8 inputs, 5 outputs). The fitness functions employed on both problems are negated counts of erroneous output bits, so maximum fitness = 0. Optimal ISC is 15, 32 and 29 respectively, and worst ISC is 509, 131,069 and 105 respectively; expected ISC, E, established by frequency sampling, is approximately 49, 564, 72 respectively. A $(10, 16)$ ES is employed for the mux-n problems, and a $(15, 50)$ ES for the adder, so no parents are carried to subsequent generations; no clones are bred either.[7] A mutation bound of 1, the minimal, is used for all experiments, as this has been found likely to be the most favourable [29]. The populations are initialised to worst π for mux-n, and randomly for the adder. The results are shown in figures 8.3 and 8.4. Note that the vertical scales for ISC are inverted so that correlation with fitness is more easily interpreted, and ISC may be plotted only within the range of primary interest. An interpretation of the results is presented in the remainder of this section.

[7] These parameters are not optimised. A smaller population was chosen for mux-n problems so that the effects of drift and the loss of the optimal phenotype are exhibited.

For both mux problems (figure 8.3), expected ISC is exceeded, and near optimal ISC reached, early on in each run. It is near the optimal where ISC appears most stable. However, there are periods where ISC undergoes temporary relapse, but is soon recovered. In the inset of figure 8.3(a) this can be seen in more detail, a correlation between ISC and fitness being apparent. An increase in ISC appears to be followed by a drop in fitness or slowing in fitness increase, while a drop in ISC appears to be followed by an increase in fitness or rate of increase. While a drop in ISC is accounted for by inherent selection for evolvability, the converse, an increase in ISC (drop in evolvability), can only be the result of random genetic drift, where mutants with high relative fitness but high ISC (low evolvability) saturate the population temporarily; this behaviour is not unexpected in a small population. In addition, for 20-mux, it can be seen that the near optimal evolvability is reached long before fitness is optimised, the population genotype appearing to exhibit foresight in predicting the optimal π.

The results for the adder also exhibit the emergence of π with better than expected ISC (figure 8.4). The population is this time initialised to random π rather than worst. While better than expected ISC is reached in around 500 generations, ISC appears to remain erratic within a wide range of values whose average is a long way off the optimal of 29, but better than the expected of 72. One run (top inset) does approach the optimal ISC early on, but this is quickly lost and never recovered like it was in for 11-mux. However, in contrast to the single run shown for 11-mux, optimal fitness is maintained during this loss evolvability. This is perhaps accountable, in part, to the larger population which counters the loss of parents in subsequent populations.

The apparent difference in the emergence of low ISC π between mux and the adder problem is now discussed. The terrain of ISC values under the N3 mutation is likely to be significant here. Both mux and adder problems are known to have many local optima under direct ISC optimisation using N3. However, the fact that the objective of the fitness function used here is optimised function, not optimised ISC, allows genetic drift to move the search away from

# perturbations	1	5	10	15
11-mux	0.0960	0.4860	1.0350	1.4980
4-bit adder	3.0960	11.3100	17.9850	22.7310

Table 8.1: ISC robustness to N3 perturbations. The column headers indicate the number of successive perturbations applied to a π with optimal ISC. The values below reflect the corresponding increase in ISC for the two problems, averaged of 1000.

becoming trapped in what would otherwise be ISC local optima. To give an indication of the comparative ISC terrain, optimal ISC were perturbed for 11-mux and the 4-bit adder, and the corresponding increases in ISC recorded. The results are shown in Table 8.1. 11-mux is clearly much more robust to perturbations than 4-bit adder, which suggests a much smoother ISC terrain for the former. The range of ISC values is 15-509 and 29-105 respectively, which enhances confidence in this conclusion. The frequencies of ISC values may also be a factor. Thus, for the adder, the population appears to become ISC-localised due to rugged ISC terrain, which is difficult to navigate under the present scheme.

The pace of adaptation

The promise offered by the evolution of evolvability is now elucidated by the presentation of some impressive fitness curves using the 11-bit multiplexer as the target function. The curves are plotted alongside ISC to provide an indication of how evolvability and fitness correspond. A $(7, 12)$ ES is employed and the population initialised with individuals having worst possible ISC. See figure 8.5.

What is impressive about these fitness curves is that they maintain an almost linear fitness increase for a prolonged period, sometimes even accelerating. It is the evolution of evolvability that facilitates this behaviour. The simultaneous increase in evolvability indicated by the emerging lower ISC values maintains the rate of fitness improvement against a search space becoming increasingly sparse in superior solutions. The fact that such curves can be induced at all

is something not readily seen in EC, and it is indicative of the potential of the evolution of evolvability within EBDDIN to align variation with fitness.

8.5 Discussion

Evolutionary phenomena are often disputed on the basis that they would have required foresight. The need for foresight in the construction of complex organisms is one of the founding tenets for the anti-Darwinian *Intelligent Design* (ID) [125] movement. Dawkins [25] addressed the issue of foresight with his famous *blind watchmaker* analogy on natural selection's ability to solve problems apparently requiring foresight:

> "All appearances to the contrary, the only watchmaker in nature is the blind forces of physics, albeit deployed in a very special way. A true watchmaker has foresight: he designs his cogs and springs, and plans their interconnections, with a future purpose in his mind's eye. Natural selection, the blind, unconscious, automatic process which Darwin discovered, and which we now know is the explanation for the existence and apparently purposeful form of all life, has no purpose in mind. It has no mind and no mind's eye. It does not plan for the future. It has no vision, no foresight, no sight at all. If it can be said to play the role of watchmaker in nature, it is the blind watchmaker."

The results presented in this chapter, if regarded superficially, appear to imply the need for foresight. It can be seen in figure 8.3b particularly that the genotype is quickly molded into one most evolvable for the fitness function. Near optimal π are reached relatively quickly whilst fitness optimisation is still in its infancy. This is no trivial matter as the variable ordering problem is NP-complete in both optimal and approximate solutions [14, 107]. How can it be that the genotype is able to determine in advance what the most evolvable configurations of π are for that particular fitness function? Yet, there is no foresight here, no

design in the manner in which variables are swapped. It is simply a matter of random variation and differential reproduction incrementally discovering and propagating better π until the optimal is reached.

The pressure to evolve is present everywhere, in all environments, regardless of whether those environments are static or dynamic. If evolvability can evolve, then the pace and extent of adaptation will be enhanced. This has clearly happened in nature, and biological populations are highly evolvable. The findings of this chapter and of chapter 6 indicate that genotypes in EBDDIN are able to self-organise at different levels of the representation's organisation so as to respond more favourably to random genotypic mutation. This was shown in this chapter by way of emergent π, and in chapter 6 by way of pleiotropic patterns that enhance evolvability. The consequence for the pace and extent of evolution in EBDDIN was shown to be dramatic (figure 8.5). An appreciation of the properties of EBDDIN that facilitate these dynamics may provide insights into how evolvability and its emergence can be achieved more generally in EC.

8.6 Summary

EBDDIN has been extended to incorporate dynamic variable reordering. Good π were shown to be correlated with high evolvability. Good, sometimes optimal π will emerge when dynamic variable reordering is employed. This emergence occurs as a logical consequence of better π being associated with greater evolvability: individuals having better π will produce fitter offspring, on average, and will propagate those evolvability traits more readily. The pace and extent of evolution was demonstrated to benefit significantly as a result. Maintenance and even acceleration in the pace of evolution was demonstrated, and this resulted from the simultaneous evolution of evolvability alongside functional fitness. Finally, it was emphasised that the genotype is able to self-organise at different levels to facilitate evolvability, and if the properties that facilitate this can be understood in the context of EBDDIN, then this may aid in providing an under-

standing and characterisation of evolvability and its evolution more generally.

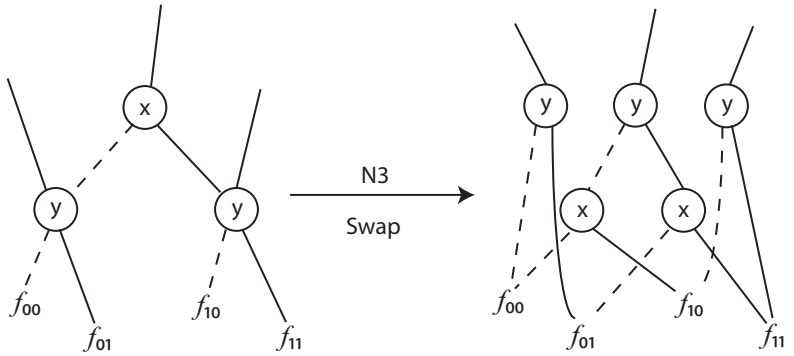


Figure 8.1: The N3 mutation: swapping adjacent variables. The two outer vertices remain unchanged. The central vertex is relabeled and its child edges redirected to two new vertices. The child edges of these two new vertices are directed to the subgraphs below so as to maintain overall function. The algebraic justification for conserving function is that:

$$\bar{x}(\bar{y}f_{00} + yf_{01}) + x(\bar{y}f_{10} + yf_{11}) = \bar{y}(\bar{x}f_{00} + xf_{10}) + y(\bar{x}f_{01} + xf_{11})$$

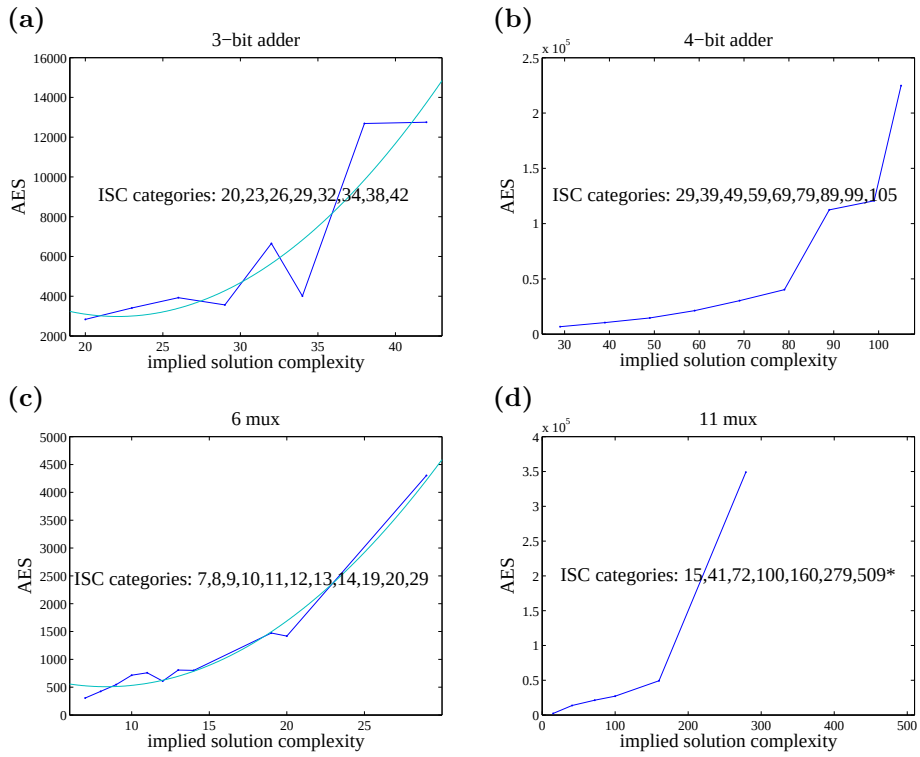


Figure 8.2: Evolvability comparison of variable orderings for selected problems. Variable orderings are categorised by their ISC value and sample categories selected to span the entire range for each problem. 100 runs are perfomed for each ISC category. All four graphs exhibit the trend of rapidly increasing AES against increasing ISC value.
* In (d) no AES value could be obtained for 509 due to to the extremely poor evolvability of this category.

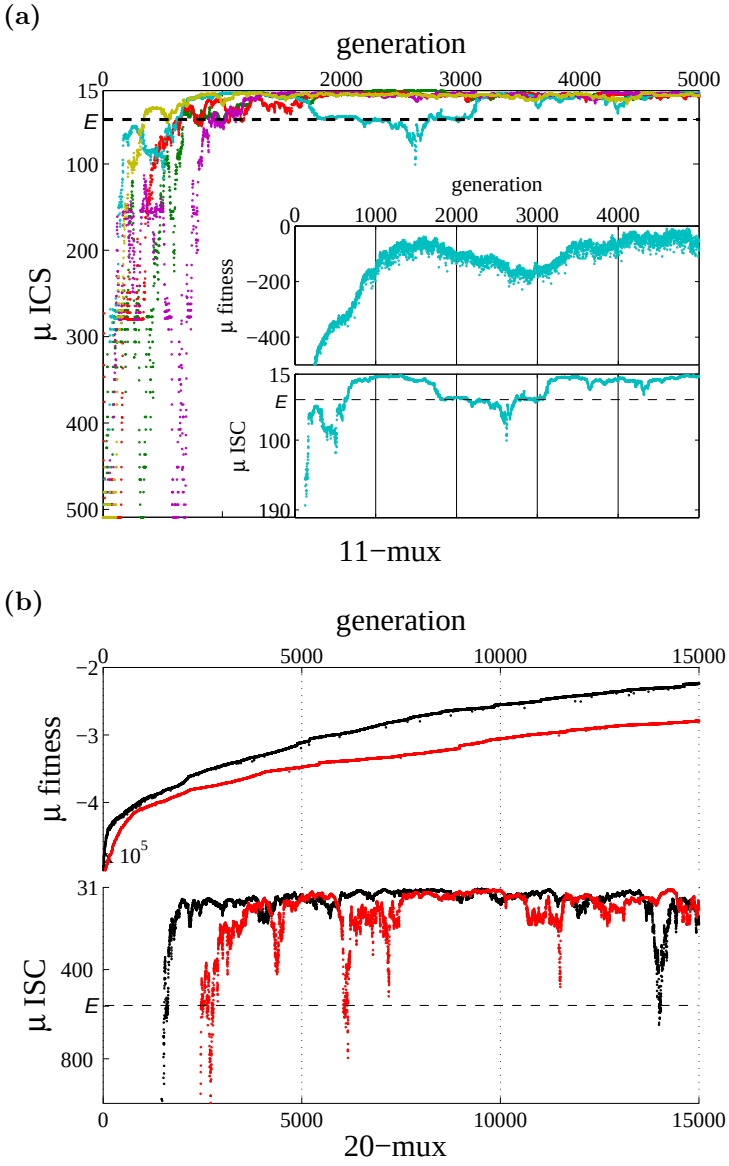


Figure 8.3: The emergence of good π. The population mean (μ) ISC and fitness values are plotted. The population is initialised to random OBDDs having worst π. (a) The main figure shows ISC values for five independent runs on 11-mux. Inset: the single run that undergoes temporary ISC relapse is shown below fitness as an indication of the correlation between ISC and fitness. Within 1000 generations all runs pass expected ISC, E, and stabilise near the optimal of 15. (b) Two runs are shown for 20-mux with fitness alongside ISC. Both runs approach optimal ISC, and remain fairly stable there, while fitness remains in the very early stages of optimisation.

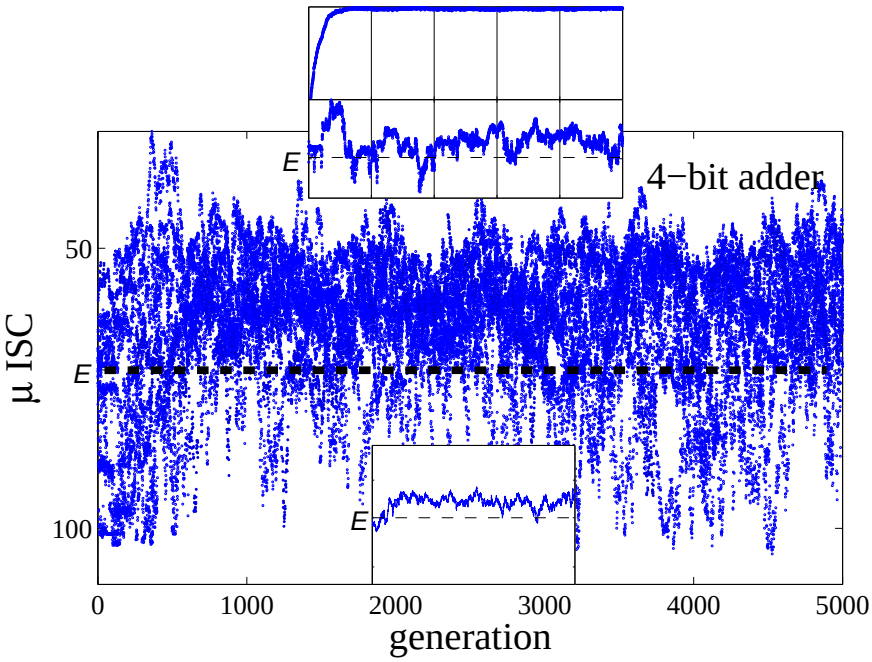


Figure 8.4: 10 runs on the 4-bit adder problem are shown with population initialised to random π. The top inset shows the a single run as ISC below fitness, and a less prominent correlation than for 11-mux. The bottom inset shows the average of all runs. ISC exceeds the expected and stabilises within 1000 generations, but the gain is, respectively, more modest than for 11-mux.

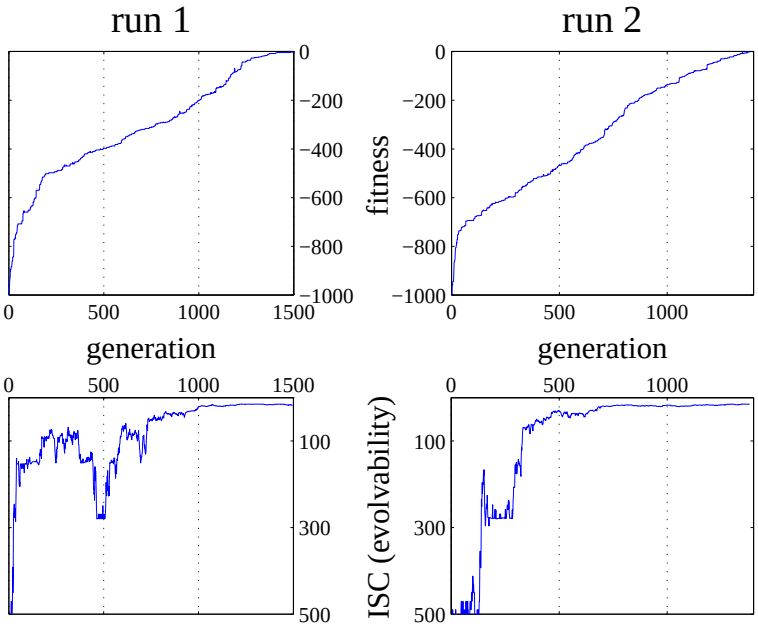


Figure 8.5: The pace of adaptation. Two fitness curves resulting from the evolution of evolvability exhibit an almost constant, sometimes even accelerating, rate of fitness increase. This is maintained for the duration of the run as high evolvability emerges in the form of low ISC.

Chapter 9

On population diversity and neutrality

EC has traditionally depicted evolutionary search as taking place on a multimodal landscape. If population diversity can be maintained, then the search will ascend the gradients of many peaks simultaneously, leaving the local optima in its wake. Accompanying this depiction are the difficulties of diversity maintenance and sensitivity to the starting conditions, both of which remain prominent issues within the EC research community.

This chapter examines the alternative and very different dynamics of evolutionary search in the neutrality-induced local-optima-free search space of EBD-DIN, focussing on the role of the population.[1] Neutrality serves to decouple genotypic variation in evolvability from fitness variation. Population diversity and neutrality work in conjunction to facilitate evolvability exploration whilst restraining its loss to drift, ultimately facilitating the evolution of evolvability. The search is found to be both tolerant of the loss of initial diversity and robust to the starting configuration.

[1] A shorter version of this chapter was presented as a paper at EuroGP '07 [30].

9.1 Introduction

Wright's landscapes, reviewed in section 3.3, have had a profound influence in EC. They have served as the foundation for hypothesising mechanisms of evolution. De Jong [58] highlights that EC often views the initial population as starting points in a parallel search process. The initial population is generated so as to be widely dispersed across the landscape: each peak may then have individuals within its sphere of influence. The problem of evolution then becomes one of having a very diverse initial population gravitate toward many peaks simultaneously, eventually converging on the highest. However, this process is highly dependent on maintaining the diversity of the initial population and highly sensitive to the starting configuration. If diversity is lost, then the population will most likely converge on a poor local optima; if the initial population is too small, or the landscape too rugged, then the higher peaks will not have individuals within their spheres of influence and will, again, converge to a poor local optima. Both of these problems are non-trivial and remain prominent issues within the EC field. Diversity is also considered essential for recombination operators because an homogenous population does not yield new solutions. Thus, the loss of the initial diversity is often considered the end evolutionary adaptation [58].

Wright's landscape concept originated in 1932. It should be considered a tribute to Wright that the concept has spawned so many theories of mechanisms of evolution. However, it must also be considered that Wright's landscapes are just wrong and misleading notions of the structure of genotype space. Instead of pursuing this possibility, much of EC has persisted with Wright's landscape depiction of the search space, using it as the foundation for hypothesising mechanisms of evolution, some of which are inconsistent with accepted evolutionary theory.[2] Hybrid approaches [7, p 35] fall into this category, in which an EA is augmented with a non-evolutionary technique in an effort to improve the search performance. The mechanism of generating gross initial diversity and

[2]Criticisms of Wright's landscape metaphor were discussed in section 3.3

necessitating its maintenance to ascend towards many peaks in a parallel adaptive search process [58] also falls into this category. Though it appears more practical than attempting to move a population from a lower to a higher peak, creating an initial population with gross diversity is clearly inconsistent with Darwinism. In nature, individuals of a population are not so grossly diverse and common descent tells us that life has diverged from a single species into many, not converged from many species into one. Any attempt by the community to theorise on mechanisms of evolution must be consistent with accepted evolutionary theory if real progress is to be made.

Wright knew nothing of molecular genetics and worked without the benefit of the contemporary theories that have followed. Some EC researchers have taken heed of such theories and begun to consider alternative notions of evolution. In particular, proponents of neutrality [59] argue a different notion of evolutionary search. Proponents of neutrality suggest neutral networks alleviate local optima and the loss of diversity is of lesser concern. For example, Ebner et al. [36] find that redundant representations increase accessibility between phenotypes through neutral walk. Harvey & Thompson [50] show that evolution can progress satisfactorily in a small, genetically converged population for an evolutionary hardware task. Miller et al. [81, 82] investigates many populations sizes for CGP and finds that smaller populations can be most efficient. Barnett [10] goes further still and argues that a non-population based approach is optimal. Studies employing RNA models have been particularly influential [53, 105, 52], the structure of RNA spaces apparently exhibiting the purported properties more readily than artificial representations. The potential of neutrality has been further recognised in [116, 130, 120], though others voice more sceptical or cautionary notes [97, 61, 110]. While it cannot be denied that it has hitherto been poorly understood how neutrality contributes toward evolvability, the neutralist depiction is consistent with contemporary evolutionary science and cannot be dismissed.

So, what is the role of population diversity in a neutrality-induced local

optima free search space? While the search space is proven to be free of local optima for this investigation (theorem 1), it is clearly not uni-modal in the intuitive sense of a hill with a single peak. The space is highly neutral and perforated by massively connected neutral networks. Should a population-based approach be considered beneficial in such a space, or should a hill-climbing approach be preferred as it would be in the intuitive idea of a search space with a single-peak? These are the questions addressed in this chapter.

The conclusions identify neutrality and population diversity working in conjunction to facilitate the evolution of evolvability in a three step process:

1. Neutral mutation creates evolvability variation whilst conserving fitness.

2. Population size facilitates evolvability exploration, but also restrains drift away from favourable evolvability characteristics.

3. Selection acts indirectly, through fitness, on evolvability, propagating the more favourable evolvability characteristics.

This depiction promotes evolvability as the principal beneficiary of genotypic variation and selection. Whilst selection acts directly on fitness, it is evolvability that is the ultimate target. This is the kind of second-order selection for evolvability discussed by Dawkins [25, 24]. Effectively, fitness becomes evolvability's selection surrogate, the expression of evolvability's latent potential, exposing to selection that which evolvability, by itself, cannot. In this, evolvability can emerge, and the evolution of evolvability be witnessed. The role of the population in this is as the vehicle for evolvability exploration, protecting from drift evolvability traits that have previously proved their worth and facilitating competition between those traits.

In contrast to the conclusions drawn in this chapter, Barnett [10] argued that a variant of hill-climber was optimal, and that a larger population was not beneficial; and Smith et al. [110] found that evolvability was not evolving during neutral evolution. The reason for these contradictory conclusions is not

examined here, but differences in the representations and operators are likely to be primarily responsible.

Sections 9.3 & 9.4 examine the effects of losing diversity, concluding the employment of *greedy selection*. Sections 9.5 & 9.6 investigate the potential of diversity and neutrality on functions contrasting in their evolvability potential.

9.2 The test functions

The investigation presented in this chapter will exploit the contrasting properties of two especially chosen target functions, the multiplexer and parity functions. The relationship between evolvability and π was investigated in chapter 8 for a number of functions. The effort required to solve a problem, in terms of the *Average number of Evaluations to a Solution* (AES), grew super-linearly in the *implied solution complexity* (ISC) of π for a fixed ordering. The multiplexer exhibited this kind of behaviour. Furthermore, under dynamic variable reordering using the N3 mutation, good π would emerge as a logical consequence of being associated with better evolvability. However, the parity function exhibits ISC linear in the number of variables and is ISC invariant under dynamic π, so will always evolve to a solution rapidly without the need for good π to emerge. Thus, evolvability is required to evolve in order to find a solution to the multiplexer problem, but this is not so for the parity problem. The contrasting properties of the parity and multiplexer functions in this respect will be exploited in the experiments discussed in the following sections.

9.3 The effects of diversity loss

This section investigates the influence on performance of constraining diversity to differing extents. Three experiments are conducted on both the 10 parity and 11-bit multiplexer functions. A population of 30 random individuals is first generated and written to disk. The results are plotted in figure 9.1.

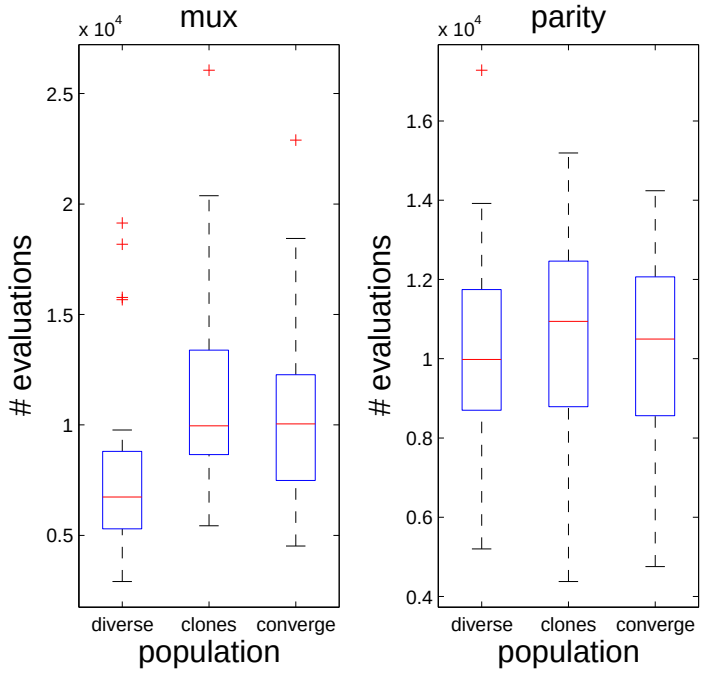


Figure 9.1: Comparison of performance with various restrictions on diversity using *box and whisker plots*. *diverse* - initial population of individuals all of which are genotypically unique; *clones* - initial population of genotypic clones; *converge* - periodically remove diversity every 50 generations by breeding only a single individual.

For the first experiment, the population is read from disk and the number of evaluations required to solve the problem plotted over 30 runs. For the second experiment, the population is initialised to clones for each of the 30 runs, one run for each of the individuals on disk. For the third and final experiment, the setup is similar to that described for the second experiment but, additionally, diversity is periodically removed every 50 generations by only breeding one parent for that generation. A (15 + 30) ES is employed. The setup facilitates comparison of a population that is not prevented from maintaining initial diversity, a population that has no initial diversity, and a population that has diversity periodically eliminated.

For both problems and all configurations a 100% success rate is maintained. Furthermore, the effect on the number of evaluations required from the loss of diversity is negligible; it is slightly accentuated for mux where the population is initialised to be diverse, but this can be attributed to the higher probability of having better evolvability (i.e. lower ISC values) present in at initialisation rather than having to wait for it to emerge. These results suggests a certain uniformity in the search space and the search can be considered highly independent of the starting configuration. Furthermore, the loss of diversity accompanying a fitness improvement step can be considered benign.

9.4 Fitness conservation and generation lag

Given that temporary loss of diversity has negligible effect on performance (section 9.3), maximising selection pressure can be considered. Altenberg [2] has emphasised the importance of strong parent to offspring fitness correlation for evolvability; achieving this through neutral mutation and selecting only the fittest individuals to be parents has great appeal. This type of *greedy selection* will be denoted $(0, \lambda)$ or $(0 + \lambda)$ in the style of ES for generational algorithms. The $\mu = 0$ indicates that the number of parents is not specified explicitly but depends on the the number of individuals currently exhibiting the equal highest

fitness, which may vary between 1 and λ.

Figure 9.2 examines how a $(0, 30)$ ES compares against standard selection, and the former is found to be favourable. Thus, the benefits of not breeding suboptimal solutions outweighs any loss in genotypic diversity from fitness diversity, emphasising the potential of neutrality to decouple genotypic variation from fitness variation, the significance of which is well-recognised [52, 105, 116].

The effects of *generation lag* must also be recognised when using AES as the performance measure. Generation lag occurs at the fitness improvement step during the generation of the child population. A fitness improvement early in the production of the child population is not available for breeding until the following generation, resulting in the breeding of inferior individuals until the child population is fully populated. The cost increases with population size and must be balanced against any beneficial effects of a larger population. In the following experiments generation lag will be recognised as a consequence of a generational algorithm, or eliminated with a steady-state variant where indicated.

9.5 Evolvability diversity

This section investigates the effect of evolvability variation within the population, and postulates the existence of an *evolvability threshold*. The objective is to identify population diversity as the inducer of favourable evolvability characteristics rather than maintaining gross genotypic diversity from initialisation. Whilst the former can easily arise gradually from a population of clones through only a minor mutation severity, the latter cannot.

Recall that low ISC indicates greater evolvability and the multiplexer function exhibits significant discrepancies in evolvability as a result of variation in π (see figure 8.2(d)). The parity function, however, exhibits no such discrepancies because it is a symmetric function, exhibiting invariant ISC for all π. Figure 9.3 examines the difference in performance characteristics that result from the

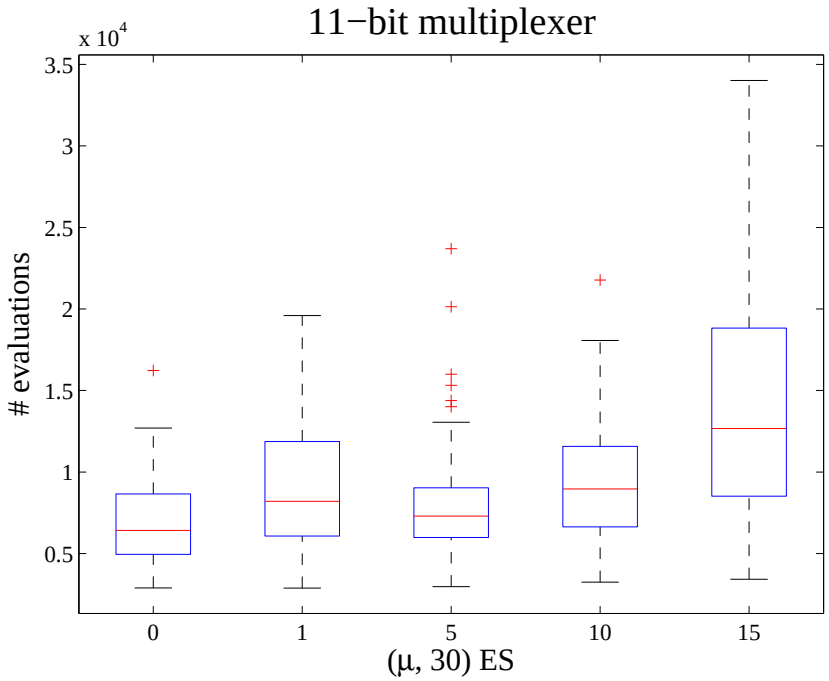


Figure 9.2: How *greedy selection* compares against standard truncation selection.

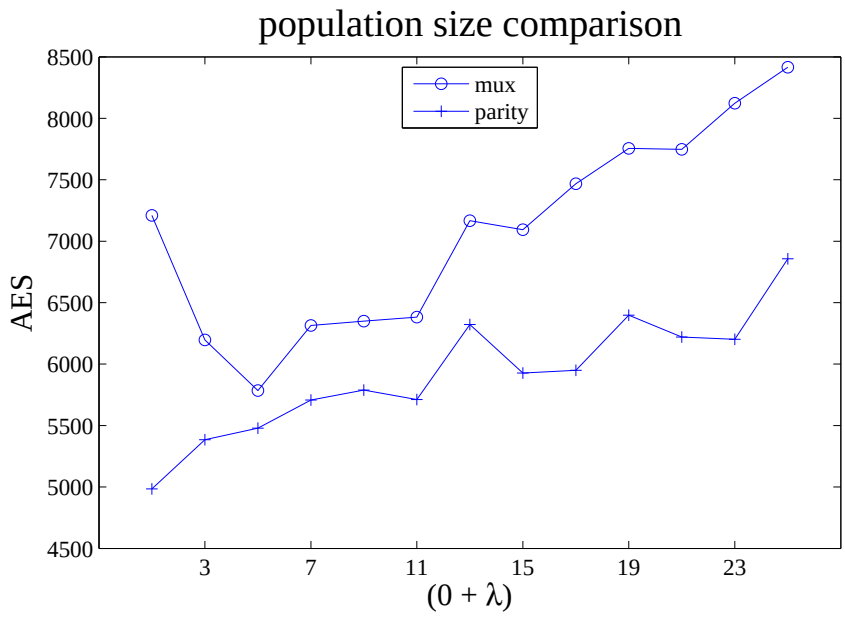


Figure 9.3: Effects of population size on performance (100 runs). Where there is potential for evolvability variation (mux), population diversity can improve performance, but generation lag taints the results.

presence of evolvability variation for different population sizes. From the figure, it is clear that the parity function exhibits an increasing AES against population size. The increase appears roughly linear and can be attributed to generation lag. For mux, a performance gain is observed up until a population size of around 5, at which point performance takes a downturn. Clearly, where there is variation in evolvability to be exploited, a larger population appears to offer some benefit, but the presence of generation lag taints the results. Thus, in the following experiments, steady-state selection is employed. *Greedy steady-state selection* chooses randomly a single parent from the set of all individuals in the population that exhibit the highest level of fitness.

Figure 9.4 shows a trace averaged over 30 runs using greedy steady-state selection. The figure shows that a larger population is better able to maintain evolvability (low ISC values), and this corresponds to more rapid fitness improvement. The distribution of ISC for 11-mux is heavily skewed, with an expectation of approximately 50, so the consequence on fitness of losing ISC is not great (see figure 8.2(d)). A function having a differing ISC distribution, however, would have adaptation much more grossly impaired.

Thus, an *evolvability diversity threshold* can be postulated. Huynen et al. [53] discuss genotypic versus phenotypic error thresholds, stating that it is the latter at which adaptation breaks down. However, the results of this section suggest adaptation can also be stifled at the evolvability threshold even whilst maintaining the phenotype. A population exhibiting a high percentage of neutral offspring is less likely to lose phenotype to such errors, but those errors are only fitness-neutral, not evolvability-neutral. Thus, evolvability properties may be lost to drift, particulary where the population size is smaller, even thought the phenotype is not lost. The diversity facilitated by a larger population essentially acts as a buffer to the loss of evolvability to neutral drift.

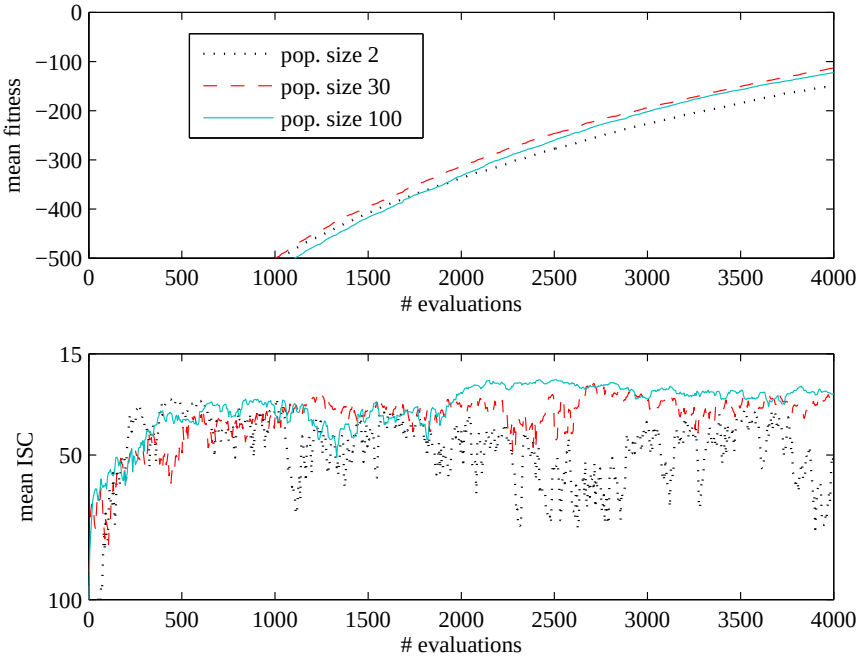


Figure 9.4: Trace on population sizes averaged over 30 runs using greedy steady-state selection (no generation lag). A larger population is better able to maintain low ISC.

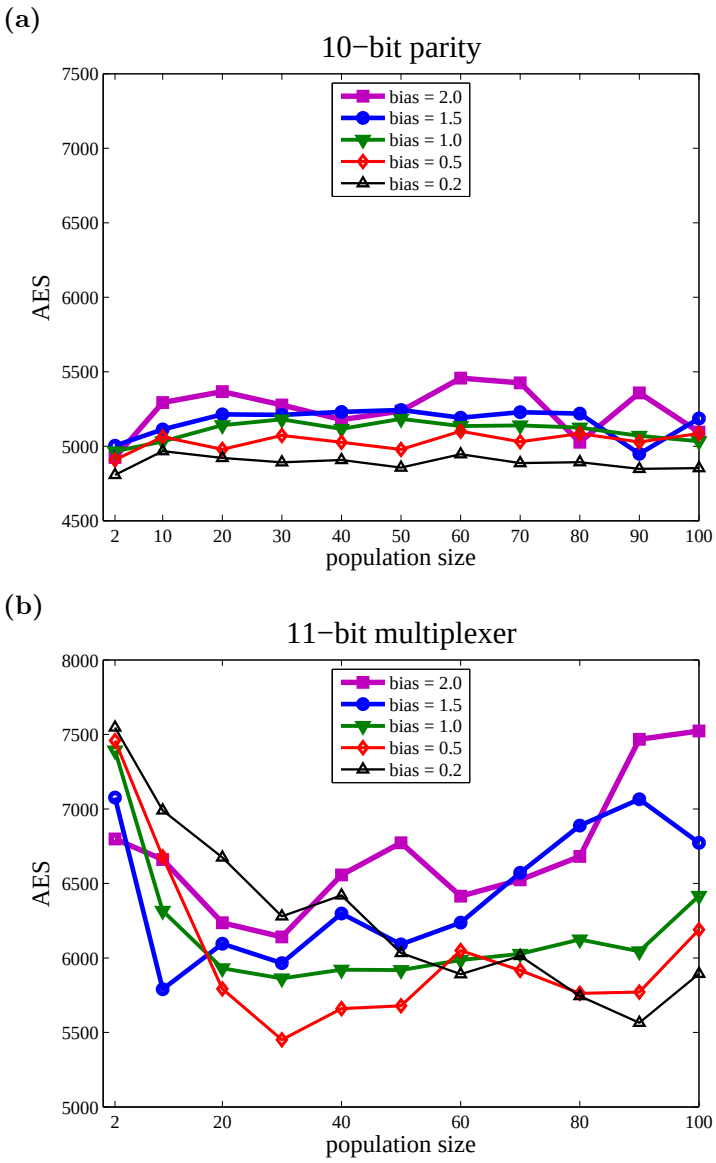


Figure 9.5: The effect on performance of population size for varying degrees of neutrality. (a) Parity - increasing population size has negligible effect on performance due to the absence of discrepancies in evolvability. More neutrality improves performance; (b) Multiplexer - increasing population size has a sometimes positive effect on performance, particularly where a higher degree of neutrality is present. See text for a full discussion.

9.6 Population size and neutrality

This section investigates the relationship between population diversity and neutrality. Figure 9.5 shows the effect of varying both population size and the degree of neutrality for both the parity and mux functions. Again, a greedy steady-state selection is employed. Results are averaged over 100 runs.

The degree of neutrality is influenced by the ratio of adaptive to neutral mutations which, in turn, is controlled by introducing a *mutation bias*: a bias of 2.0 means that it is twice as likely that the adaptive A1 mutation is chosen for application than it would be otherwise; a bias of 0.5 means that A1 is half as likely to be chosen. Thus, a lower bias means a lower ratio of adaptive to neutral mutations and, therefore, more neutral offspring. In the figures, the size of the bias is reflected by heavier lines, so lighter lines reflect the fact that comparatively more neutral drift is present. The population is initialised to clones to eliminate any diversity oriented initialisation benefit for larger populations; refer to figure 9.1 for a reminder of how this can affect performance.

For the parity function, figure 9.5(a), increasing population size has negligible effect on performance and AES remains within a relatively small range. This behaviour is expected given that there are known to be no variation in evolvability resulting from exploration of π: one individual is just as evolvable as another. However, an increase in neutral drift through a lower mutation bias improves performance independently of population size. More neutral drift means greater exploration generally and less effort re-sampling familiar areas of the space.

In contrast, for the multiplexer function, figure 9.5(b), trends against population size are discernable and the range of AES values is much greater. Increasing population size results in a decrease in AES followed by an increase in AES in most cases. Too large a population stifles neutral drift and, therefore, exploration. At the smallest population size, more neutrality consistently results in poorer performance, favourable π being more readily lost to neutral drift.

At the largest population, however, the situation is completely reversed. The population is able to tolerate greater neutrality-induced exploration while maintaining evolvability, resulting in overall better performance where neutrality is higher. This result suggests a synergy in increasing neutrality and population size. While neutrality encourages exploration and increases mutational robustness, greater diversity facilitates that exploration and makes the population robust to evolvability damage from drift.

9.7 Discussion

The aim of this chapter was to determine if and how population diversity might be beneficial in a neutrality-induced local-optima free search space. Barnett [10] had previously argued that population diversity was not beneficial in such spaces. The conclusion states that population diversity is beneficial because it facilitates the evolution of evolvability. Without diversity, any favourable evolvability traits would be lost to drift.

These results are particularly dependent on the AES performance measure and EBDDIN's capability for effortless neutral walk; the cost of evaluating neutral offspring would have to be balanced against the benefits otherwise. This advantage makes EBDDIN more akin to natural systems than less so. In nature, the evaluation of a population's fitness is highly parallel and a greater propensity for neutral offspring can be reflected in an expanding and subdividing population. In contrast, evaluation of a population in artificial evolution is typically done serially, as reflected by the AES performance measure, and the population size and number is also typically fixed. Thus, there is a *serial evaluation deficiency* associated with artificial evolution, and EBDDIN's capability to circumvent the evaluation cost of many offspring alleviates much of that deficiency.

The generality of these results may be questioned because they are conducted in a local optima free search space. However, regardless of the nature

of the search space, evolvability variation within a population is clearly significant and a prerequisite for the evolution of evolvability, a property considered fundamental in the evolution of complex systems. Indeed, who is to say that for a lineage to progress satisfactorily towards ever-greater complexity, local optima sparsity, neutrality-induced or otherwise, is not a prerequisite? Perhaps, rather than hypothesising mechanisms to navigate the multi-modal spaces that are Wright's landscapes, the EC field might better focus on transforming the search space of a problem with neutrality into one more susceptible to gradual evolution? This research provides insights into the role of population diversity in such spaces and will aid those researchers and practitioners adopting such an approach.

9.8 Summary

This chapter investigated the role of population diversity in a search space that is free of local optima by way of neutrality. Using EBDDIN and two contrasting functions, one invariant in evolvability, and the other not, the dynamics of how evolvability evolves under different population sizes was analysed. Where evolvability is invariant, population diversity offers no benefit. However, where evolvability is variable, population diversity encourages the evolution of evolvability, facilitating neutral evolvability variation whilst restraining the loss of favourable evolvability traits to drift. The difficulties of diversity maintenance and sensitive to the starting configuration that are the legacy of Wright's adaptive landscapes have been avoided. Thereby, it is argued that evolutionary search in search spaces constructed so as to be free of local optima by way of neutrality is a promising future direction for EC research.

Chapter 10

Conclusion

This thesis has introduced a new approach to evolving BDDs which exploits the neutrality inherent in the representation: *Evolving Binary Decision Diagrams using Inherent Neutrality* (EBDDIN). The focus has been on using EBDDIN as a computational model to develop an understanding of evolvability and its evolution within that context. Significant insights have been gleaned into the properties and mechanisms that can facilitate evolvability. These understandings may lead to a deeper understanding of evolvability more generally.

10.1 Answers to the thesis questions

In the introductory chapter of this thesis, questions that the thesis addresses were posed as an indication of what the thesis was about. Those questions are now restated along with summarised answers .

Is there a better approach to exploiting the BDD data structure for artificial evolution? To address this question, chapter 5 introduced the EBDDIN approach, which exploits the neutrality inherent in the BDD representation. The EBDDIN approach was developed further throughout the rest of thesis, and extended to facilitate dynamic variable reordering in chapter 8. Many of the practical issues regarding EC are addressed by EBDDIN. The number of

fitness evaluations are reduced by exploiting explicitly neutral mutations. The cost of each fitness evaluation for Boolean fitness functions is dramatically reduced by the method of *difference evaluation*. Redundancy is easily removed by exploiting the ability to reduce BDDs to their canonical form. In comparisons of performance against other approaches to evolving BDDs, the EBDDIN approach proved superior against all of the tested benchmarks.

How might evolvability be represented within the genotype and what properties make one genotype more evolvable than another? Chapter 6 addressed this question. It was shown that pleiotropic patterns can emerge in which genotypic features (i.e. edges) represent a biased fraction of adapted and maladapted traits (i.e. fitness cases). That is, the genotypic representation of adapted and maladapted traits begins to separate under normal evolutionary forces. Modularity was shown to be important and reasoned to ensure that the number of edges influencing the set of adapted traits does not have to be proportional to the number of adapted traits. Mutation is then able to focus, perturbing maladapted traits while leaving relatively unperturbed the adapted traits even when fitness is high and the proportion of adapted traits approaches saturation.

What are the properties and mechanisms that facilitate the emergence and evolution of evolvability? Answers to this question were produced throughout the thesis. Chapter 6 highlighted that it was the massive redundancy of the OBDD representation that permitted pleiotropic patterns to exist that favour evolvability. Chapter 5 argued that it was the massively connected neutral networks that permit such pleiotropic patterns to be discovered through neutral evolution. Chapter 7 showed the benefits of maximising exploitation through minimal mutation, leaving exploration to be achieved through neutrality. Chapter 8 demonstrated that the genotype can configure itself at a secondary level to facilitate the pleiotropic pattern witnessed in chapter 6. Configurations (i.e. variable orderings) that better facilitate favourable pleiotropic patterns emerge as a logical consequence of being associated with better evolvability. Chapter 9 examined the role of the population in search space that is free of local optima

by way of neutrality. It was shown that the population diversity facilitated the evolution of evolvability.

What is the role of neutrality in evolutionary search? Neutrality has been emphasised throughout the thesis, and the type of neutrality investigated exploited functional redundancy exclusively. It was shown to facilitate exploration without loss of fitness, not only in terms of higher fitness regions, but also in terms of pleiotropic patterns and higher-level configurations (i.e. variable orderings) that are important for achieve the variation component of evolvability. Neutral networks are so pervasive so as to completely alleviate local optima for the class of Boolean fitness functions (chapter 5). Chapter 6 showed that pleiotropic patterns could evolve through neutral mutation. Chapter 7 showed that neutrality was a better source of exploration than increased mutation rate, also implying a positive correlation between evolvability and mutational robustness. Finally, chapter 9 showed that neutral evolvability variation within a population permits the evolvability traits of such individuals to compete through their offspring, ultimately facilitating the evolution of evolvability.

10.2 Contributions

The contributions of the thesis are now listed. Clarifying comments are also given, along with pointers to the most relevant chapters where appropriate.

1. **A new and improved approach to evolving BDDs that also facilitates dynamic variable reordering.**

   Previous approaches to synthesising functions in the BDD representation using EC have not proved particularly successful. The EBDDIN approach proves superior in the number of fitness evaluations required on a number of benchmark functions against the compared approaches [128, 103, 118]. EBDDIN also facilitates dynamic variable reordering.

2. **A new EA to be exploited by the EC community.** The field of EC

is broad, exploiting many representations and approaches, some of which have become common place or standardised, but each of which comes with its own limitations and range of applicability. In EBDDIN, EC researchers and practitioners have another EA to draw upon.

3. **An investigation into how evolvability is represented within the genotype.**

   Chapter 6 investigates this within the context of EBDDIN and characterises evolvability as being represented by pleiotropic patterns which constrain the phenotypic effects of random genotypic mutation. Mutation is then able to perturb maladapted traits while leaving adapted traits relatively unperturbed. Modularity plays an important role by facilitating control of the relative exposure of adapted traits to variation.

4. **A computational model demonstrating the evolution of evolvability within the context of a static environment.** Chapters 8 & 9 demonstrate that structural configurations (i.e. OBDD variable orderings) emerge as a logical consequence of better facilitating evolvability. This supports Dawkins [24] claim that evolvability is a selectable trait and is selected for in evolution indirectly. The impact on an evolutionary run is shown to be dramatic, sometimes achieving a linear rate of fitness increase with a static fitness function for almost the duration of the run. Reisinger et al. [94] and Turney [117] have previously argued that a dynamic fitness function was necessary for the evolution of evolvability.

5. **An investigation into the role of neutrality in evolutionary search.**

   Neutrality is shown to be a crucial, and the most important, source of exploration within the context of EBDDIN and the problem domain employed. Neutrality is shown to completely alleviate local optima for the class of Boolean fitness functions (chapter 5). Chapter 7 shows that neutrality is a better source of exploration than increased mutation rate, and is cost-free under EBDDIN, allowing the trade-off between exploration

and exploitation to be circumvented. Chapters 5, 6 & 9 also shows that neutral variation in evolvability traits is important for the evolution of evolvability.

6. **An investigation into the role of population diversity in a search space that is free of local optima by way of neutrality, which shows diversity to facilitate the evolution of evolvability.**

   Chapter 9 shows that although the search performance is tolerant of the absence of initial diversity, the diversity that results from gradual mutation is important and facilitates the evolution of evolvability by permitting the evolvability traits of individuals to compete via their offspring. Selection is thus seen as acting indirectly on evolvability traits, and by this, favourable evolvability traits are propagated. Barnett [10] had previously argued that population diversity was not beneficial in such a space, and that a minimally-sized population was optimal.

7. **An investigation into the the role of mutation rate in a search space that is free of local optima by way of neutrality.**

   Chapter 7 investigates this with EBDDIN, and the notion of search space *adequacy* is introduced. Gradual genotypic mutation is shown to provide the best performance over *completely adequate* search spaces for the problems investigate. It is further argued that the balance typically assumed necessary between exploration and exploitation can be avoided by utilising cost-free exploration from explicitly neutral mutations. The most gradual mutation maximises heredity not only of fitness, but also of favourable evolvability traits.

8. **Gradual evolution in a search space that is free of local optima by way of neutrality present as a viable and more plausible alternative to problematic evolution on multi-modal landscapes. This is demonstrated on Boolean functions within the context of EBDDIN.**

The thesis adds to a growing body of evidence that a desirable pace and extent of evolution may require a search space structured differently to that envisaged by Wright [127]. The problems of evolution on Wright's landscapes have been well-studied, and no general evolutionary mechanism for successfully navigating them has been discovered. This thesis shows that gradual evolution in a search space that is free of local optima by way of neutrality can exhibit characteristics with a greater consistency with biological evolution.

No claim is made that it is always possible to formulate a given problem to be free of local optima by way of neutrality to permit effective gradual evolution. The claim is only that it may be possible with the right choice of representation and operators and other components, and should be given serious consideration as an alternative to problematic evolution on multi-modal landscapes.

10.3 Implications

The aim of this thesis has not been to argue that gradual, neutrality-induced local-optima-free search is always possible, or always fruitful. Indeed, it has been shown in the thesis that even in such a search space, evolvability is poor where the target function does not have a compact ROBDD representation. This thesis has demonstrated that this kind of evolutionary search is *possible* given the right set of circumstances, and that it can avoid the problems inherent with evolution on multi-modal landscapes.

The insights into the mechanisms and properties of EBDDIN that facilitate evolvability were elucidating. The role of neutrality in facilitating exploration while conserving fitness was shown to be crucial to the search process. Population diversity was shown to facilitate the evolution of evolvability in conjunction with neutrality. Gradualism and modularity were also shown to play crucial roles. Moreover, the relationships and interdependencies between these

concepts in facilitating evolvability provides even greater insight. The deep understanding provided within the limited context of EBDDIN will yield a more general understanding of evolvability, and aid in the design of EAs attempting to facilitate evolvability and the evolution of evolvability.

The search characteristics demonstrated within this thesis have a clear potential for facilitating open-ended evolution. Hitherto attempts at demonstrating open-ended artificial evolution eventually grind to a halt [114], and this may well be because of properties of the search space and fundamental flaws in the way it is depicted. Any model of evolution that exhibits properties and characteristics closer to that of natural systems offers great potential for advancement. Robust, scalable search that is insensitive to the starting configuration and accepting of absence of initial diversity, and the periodic loss of diversity, is precisely what open-ended evolution would appear to demand. That neutrality and minimal mutation severity are important components of the model adds to its credibility. That the neutral networks have similar properties to RNA spaces gives further testimony to the credibility of the model, as does the ability of the model to demonstrate directed phenotypic variation from random mutation in a manner not inconsistent with the theory facilitated variation. A successful demonstration of open-ended artificial evolution may well come from a model with similar properties and characteristics.

10.4 Evaluation

The success of this thesis is best evaluated by the insights provided into the properties and mechanisms that can facilitate evolvability and its emergence. Hitherto, evolvability and the mechanisms that facilitate its emergence have been poorly understood by the EC community. This thesis has made a small step in redressing that lack of understanding. The philosophical test of this thesis will be whether the reader will be encouraged to accept the concept of gradual evolution in a neutrality-induced local-optima-free space as a viable

alternative approach to evolution on multi-modal landscapes. The practical test of this thesis will be whether the understanding of the concepts, properties and mechanisms that have been shown to facilitate evolvability within EBDDIN will be employed for practical applications and designed into other systems of artificial evolution.

10.5 Future directions

Extensions and enhancements

The problem domain considered in this thesis has been significant, but limited. It would be useful to examine how the EBDDIN approach performs on other types of fitness function. BDDs are sometimes used in AI for planning problems [55], such as robot control.

In addition, EBDDIN, as developed in this thesis, is based on the most basic form of BDD and multi-rooted BDD. There are many BDD variants for representing different types of function, and having different properties. A promising future direction, to extend the problem domain of the approach, investigate other BDD variants on a range of problems.

Given the apparent scalability of the approach on parity problems (see section 5.7), a mathematical characterisation of the properties of the representation (i.e. neutrality, redundancy, etc.) that facilitate evolvability would provide a more formal understanding. This may then be used to generalise the properties, and apply that knowledge in the wider field. This is no trivial matter, however. The mathematical analysis of EAs is recognised as extremely difficult, and the OBDD representation is extremely complex. Furthermore, it is not only the dynamics of fitness that needs to be considered, but the dynamics of evolvability, which complicates the matter further. However, the difficulties should be seen as a worthy challenge.

Developmental EBDDIN

A promising future direction for EC generally is developmental EAs [11, 67]. This involves complex processes which grow the phenotype from the genotype with a view to creating greater complexity of the phenotype, and is how nature is able to generate the great complexity of the human mind and body from just a few thousand genes. It will be interesting to investigate if the evolvability principles revealed in this thesis could be transferred into a developmental version of EBDDIN for growing complex phenotypes.

Application ot classification

A classifier learns to classify data by observing already classified data, i.e. training data. The objective is to capture features of the training data that determine how unseen cases should be classified. Examples of approaches to classification are: decision trees, neural networks, GA, support vector machines, Bayesian methods, k-nearest neighbor algorithms, and many others. Some key issues for classifiers are:

1. Avoiding over-fitting of the training data. Algorithms are prone to accurately learning to classify the training data, but fail to generalise to unseen cases.

2. Avoiding local optima. Most real-world problems exhibit multi-modal landscapes under most approaches, making greedy approaches particularly subject to the trapping in local optima.

3. Which data features are important for classification?

These problems are prevalent in current approaches to classification. An approach derived from EBDDIN will address these issues:

1. EBDDIN provides freedom from local optima for these types of fitness functions.

2. By the principle of Occam's razor, the simplest BDD can be considered to be the best generalising. This point was recognised by Droste [34] in his approach to evolving BDDs.

3. Data features that are not important may not appear in the graph, or can be pruned.

Application to the Verifying Evolutionary Algorithm

The Verifying Evolutionary Algorithm (VEA) is a concept that seeks to bring together EC and methods of automated formal verification (e.g. probabilistic model-checking [102]). EC is recognised for producing novel solutions to problems where manual design methods fail, but criticised because those solutions are poorly understood so cannot be trusted for safety critical applications, for example. On the other hand, automated formal verification techniques are recognised for their power to prove the correctness of systems that are difficult to design manually, such as concurrent systems. A successful coupling of these technologies will facilitate the automated generation of verifiably correct and novel solutions to difficult problems that are trusted for any application.

The first application of model-checking in EC appears to be that of Greenwood & Song [47]. A proof of concept for the VEA [31] has been developed by the author of this thesis using probabilistic model-checking [102] and a matrix genotypic representation. Later work by Johnson [56] also recognised the potential of the concept using standard model-checking. The type of problems tackled in both cases were simple, and scaling to more difficult problems appeared the most pressing challenge. Model-checking algorithms often make use of BDDs to represent state-spaces, so they are a natural representation for the VEA. Coupled with this, the EBDDIN approach may prove beneficial in scaling to more challenging problems.

Evolvable Hardware

As highlighted by Sakanashi et al. [103], BDDs are a useful representation for *evolvable hardware* systems. The verification of such systems is dependent on the representation used (i.e. truth table, finite-state machine, BDDs, etc). Because BDDs can represent functions efficiently, and be verified efficiently in comparison to other representations, they are a natural choice. Hitherto, methods to evolve BDDs have had limited success, succumbing to the problems of scalability and sensitivity to the starting conditions, which are evident in much of EC. With the EBDDIN approach, however, these obstacles may be overcome to some extent.

Chapter 11

Glossary

BDD See *Binary Decision Diagram.*

Binary Decision Diagram (BDD A Boolean function representation originally introduced by Lee [70] and further by Akers [1]. A BDD is a rooted, directed, acyclic graph. Contemporary usage usually refers to the variant *Ordered Binary Decision Diagrams.*

Box and whisker plot A diagrammatic depiction of the the spread of a data set. The first quartile (Q_1), second quartile or median (Q_2), and third quartile (Q_3) are represented by the box ends and a central line for Q_2. The whiskers indicate the end points of the data set, though data elements that are a distance greater than $1.5 \times Q_3 - Q_1$ of the box ends are considered outliers and are indicated by a cross. See figure 9.1 for an example.

Character complex A group of phenotypic traits, or characters, which have become integrated through the genotype-phenotype mapping process to serve some primary function.

Constructional fitness A term defined by Altenberg [2] to refer to blocks of code in GP which proliferate based on their probability of increasing fitness of the genotypes they reside in.

EBDDIN Is the the acronym for *Evolving Binary Decision Diagrams using Inherent Neutrality.* EBDDIN is an evolutionary algorithm characterised by explicitly neutral mutations on a BDD genotypic representation which, when applied on their own, are capable of exploring many (or all) redundant variants of a given function through neutral walk, and this will be true for all functions.

EC See *evolutionary computation*.

Evolutionary Computation (EC) The branch of computer science that seeks to mimic and exploit concepts from natural evolution for artificial evolution.

Evolvability Broadly, the capacity to evolve: it includes both a variational component and a selection component [60]. The variational component is the property of the genotype that determines how it responds to random mutation in terms of generating phenotypic variation: it is a selectable trait [24]. The selectional component relates to how parents are selected to breed.

Exploitation versus exploration Is the concept that search algorithms need to both explore new areas of the search space and exploit solutions that have already been discovered. In the context of evolutionary algorithms, exploitation is usually achieved by a minimal rate of mutation and selecting only the fittest individuals in a population to breed; exploration is usually achieved by an increased mutation rate and less discerning selection of parents. Thus, increasing exploitation and exploration simultaneously is usually seen as antagonistic, and a balance, or trade-off, between the two is usually considered necessary [79, p.45].[1]

Facilitated variation A theory that claims to fill a gap in Darwinian theory, by explaining how constrained phenotypic variation is generated by random genotypic variation [60].

Neutral theory A theory pertaining to molecular evolution which claims that most genotypic change is selectively neutral [59].

OBDD See *Ordered Binary Decision Diagram*.

Ordered Binary Decision Diagram (OBDD) A Boolean function representation developed by Bryant [16]. It is distinguished from the more general BDDs developed by Lee [70] and Akers [1] by imposing an ordering restriction of the appearance of Boolean variables along all paths through the graph.

Phenotypically isolated A genotype that can not be mutated with a single application of a variation operator to exhibit a different phenoype.

[1] One of the contributions of this thesis is to show that a trade-off is not always necessary where cost-free neutrality can be exploited for exploration.

Pleiotropic utility The fraction of *pleiotropy* for a given edge or vertex that is correct with respect to the fitness function.

Pleiotropy In biology, when a single gene influences multiple phenotypic traits. In the context of this thesis, pleiotropy where an edge or vertex of an OBDD genotype is responsible for processing multiple fitness cases. It is quantified by the number of fitness cases processed by a given edge or vertex.

Robustness More precisely, *mutational robustness to perturbation*. The degree to which a genotype can tolerate mutational perturbation and still produce a phenotype having high fitness.

Second-order selection A type of selection postulated by Dawkins [24] for determining how evolvability traits are selected for. This type of selection acts not on the traits that determine the fitness of the individual, but on the traits that determine the fitness of the descendants of the individual. If individuals A and B have the same fitness but A has better evolvability traits, then A's offspring will typically be fitter and its evolvability traits will be propagated more readily.

Utility See *pleiotropic utility*.

Variable ordering problem Is the problem of determining a variable ordering to minimise the size of an ROBDD. This problem is very hard (NP-complete) in both optimal and approximate solutions [14, 107].

Bibliography

[1] Sheldon B. Akers. Binary Decision Diagrams. *IEEE Transactions on Computers*, C-27(6):509–516, June 1978.

[2] Lee Altenberg. The evolution of evolvability in genetic programming. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press, 1994.

[3] Lee Altenberg. Modularity in evolution: Some low-level questions. In Diego Rasskin-Gutman and Werner Callebaut, editors, *Modularity: Understanding the Development and Evolution of Complex Natural Systems*. MIT Press, Cambridge, MA, USA, 2004.

[4] Henrik Reif Andersen. An introduction to Binary Decision Diagrams. Notes for course 49285: Advanced Algorithms, Technical University of Denmark, 1997.

[5] P. J. Angeline and J. B. Pollack. Evolutionary module acquisition. In D. Fogel and W. Atmar, editors, *Proceedings of the Second Annual Conference on Evolutionary Programming*, pages 154–163, La Jolla, CA, USA, 25-26 1993.

[6] Wolfgang Banzhaf. Genetic programming for pedestrians. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, page 628, University of Illinois at Urbana-Champaign, 17-21 1993. Morgan Kaufmann.

[7] Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, and Frank D. Francone. *Genetic Programming: An Introduction*. Morgan Kaufmann, 1998.

[8] L. Barnett. Ruggedness and neutrality - the NKp family of fitness landscapes. In C. Adami, R. K. Belew, H. Kitano, and C. Taylor, editors, *ALIFE VI, Proceedings of the Sixth International Conference on Artificial Life*, pages 18–27. ALIFE, The MIT Press, 1998.

[9] L. Barnett. Ruggedness and neutrality: The nkp family of fitness landscapes. In C. Adami, R. Belew, H. Kitano, and C. Taylor, editors, *Proceedings of the Sixth International Conference on Artificial Life*, pages 17–27. MIT Press, 1998.

[10] Lionel Barnett. Netcrawling - optimal evolutionary search with neutral networks. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 30–37, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.

[11] Peter Bentley and Sanjeev Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 35–43, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.

[12] Jochen Bern, Christoph Meinel, and Anna Slobodová. Efficient obdd-based boolean manipulation in cad beyond current limits. In *DAC '95: Proceedings of the 32nd ACM/IEEE conference on Design automation*, pages 408–413, New York, NY, USA, 1995. ACM.

[13] Tobias Blickle and Lothar Thiele. Genetic programming and redundancy. In J. Hopf, editor, *Genetic Algorithms within the Framework of Evolutionary Computation (Workshop at KI-94, Saarbrücken)*, pages 33–38. Max-Planck-Institut für Informatik (MPI-I-94-241), 1994.

[14] B. Bollig and I. Wegener. Improving the variable ordering of OBDDs is NP-complete. *IEEE Transactions on Computers*, 45(9):993–1002, 1996.

[15] Daniel Brand. Verification of large synthesized designs. In *ICCAD '93: Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design*, pages 534–537, Los Alamitos, CA, USA, 1993. IEEE Computer Society Press.

[16] Randal E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, August 1986.

[17] Randall E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, September 1992.

[18] Shan-Tai Chen, Shun-Shii Lin, Li-Te Huang, and Chun-Jen Wei. Towards the exact minimization of bddsan elitism-based distributed evolutionary algorithm. *Journal of Heuristics*, 10(3):337–355, 2004.

[19] Steffen Christensen and Franz Oppacher. An analysis of Koza's computational effort statistic for genetic programming. In James A. Foster, Evelyne Lutton, Julian Miller, Conor Ryan, and Andrea G. B. Tettamanzi, editors, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002*, volume 2278 of *LNCS*, pages 182–191, Kinsale, Ireland, 3-5 April 2002. Springer-Verlag.

[20] M. Collins. Finding needles in haystacks is harder with neutrality. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantu-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llora, Spiros Mancoridis, Martin Pelikan, Guenther R. Raidl, Terence Soule, Andy M. Tyrrell, Jean-Paul Watson, and Eckart Zitzler, editors, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 2, pages 1613–1618, Washington DC, USA, 25-29 June 2005. ACM Press.

[21] U. Costa, D. eharbe, and A. Moreira. Advances in BDD reduction using parallel genetic algorithm. In *Proceedings of the 10th International Workshop on Logic Synthesis (IWLS'01).*, 2001.

[22] Nichael L. Cramer. A representation for the adaptive generation of simple sequential programs. In John J. Grefenstette, editor, *Proceedings of an international conference on Genetic Algorithms and their applications.*, pages 183–187. Lawrence Erlbaum Associates, 1985.

[23] Charles Darwin. *The Origin of Species by Means of Natural Selection.* John Murray, London, first edition, 1859.

[24] Richard Dawkins. The evolution of evolvability. pages 201–220, 1989.

[25] Richard Dawkins. *The Blind Watchmaker.* Penguin Books, London, 1991.

[26] Richard M Downing. Investigating the search properties of cgp, and the cgp framework. Master's thesis, School of Computer Science, University of Birmingham, September 2003.

[27] Richard M. Downing. Evolving Binary Decision Diagrams using implicit neutrality. In David Corne, Zbigniew Michalewicz, Marco Dorigo, Gusz Eiben, David Fogel, Carlos Fonseca, Garrison Greenwood, Tan Kay Chen,

Guenther Raidl, Ali Zalzala, Simon Lucas, Ben Paechter, Jennifier Willies, Juan J. Merelo Guervos, Eugene Eberbach, Bob McKay, Alastair Channon, Ashutosh Tiwari, L. Gwenn Volkert, Dan Ashlock, and Marc Schoenauer, editors, *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2107–2113, Edinburgh, UK, 2-5 September 2005. IEEE Press.

[28] Richard M. Downing. Evolving binary decision diagrams with emergent variable orderings. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund Burke, Juan J. Merelo-Guervos, L. Darrell Whitley, and Xin Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *LNCS*, pages 798–807, Reykjavik, Iceland, 9-13 September 2006. Springer-Verlag.

[29] Richard M. Downing. Neutrality and gradualism: encouraging exploration and exploitation simultaneously with binary decision diagrams. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 615–622, Vancouver, 6-21 July 2006. IEEE Press.

[30] Richard M. Downing. On population size and neutrality: facilitating the evolution of evolvability. In Marc Ebner, Michael O'Neill, Anikó Ekárt, Leonardo Vanneschi, and Anna Isabel Esparcia-Alcázar, editors, *Proceedings of the 10th European Conference on Genetic Programming*, volume 4445 of *Lecture Notes in Computer Science*, Valencia, Spain, 11 - 13 April 2007. Springer.

[31] Richard Mark Downing. Research progress report 2: Formally verified solutions from evolutionary algorithms. Technical report, School of Computer Science, University of Birmingham, April 2004.

[32] Richard Mark Downing. Evolving binary decision diagrams using implicit neutrality. In *The 2005 IEEE Congress on Evolutionary Computation, CEC 2005*, pages 2107–2113, Edinburgh, Scotland, UK, September 2005. IEEE Press.

[33] Rolf Drechsler and Nicole Göckel amd Bernd Becker. Learning heuristics for OBDD minimization by evolutionary algorithms. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – PPSN IV*, pages 730–739, Berlin, 1996. Springer.

[34] Stefan Droste. Efficient genetic programming for finding good generalizing boolean functions. In John R. Koza, Kalyanmoy Deb, Marco Dorigo,

David B. Fogel, Max Garzon, Hitoshi Iba, and Rick L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 82–87, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.

[35] Stefan Droste, Dominic Heutelbeck, and Ingo Wegener. Distributed hybrid genetic programming for learning boolean functions. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, volume 1917 of *LNCS*, pages 181–190, Paris, France, 16-20 September 2000. Springer Verlag.

[36] 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 CEC2001*, pages 1–8, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.

[37] Marc Ebner, Mark Shackleton, and Rob Shipman. How neutral networks influence evolvability. *Complexity*, 7(2):19–33, 2001.

[38] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.

[39] Aniko Ekart and S. Z. Nemeth. Selection based on the pareto nondomination criterion for controlling code growth in genetic programming. *Genetic Programming and Evolvable Machines*, 2(1):61–73, March 2001.

[40] Larry J Eshelman. *Genetic algorithms*, chapter B1.2. The Handbook of Evolutionary Computation. The Insitute of Physiscs, 1997.

[41] R.A Fisher. *The Genetical Theory of Natural Selection*. Oxford University Press, 1930.

[42] Laurence J. Fogel. *Intelligence Thorugh Simulated Evolution: Forty years of evolutionary programming*. Wiley, 1999.

[43] Richard Forsyth. BEAGLE A Darwinian approach to pattern recognition. *Kybernetes*, 10:159–166, 1981.

[44] Chris Gathercole and Peter Ross. An adverse interaction between crossover and restricted tree depth in genetic programming. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*,

pages 291–296, Stanford University, CA, USA, 28–31 July 1996. MIT Press.

[45] Nicholas Geard, Janet Wiles, Jennifer Hallinan, Bradley Tonkes, and Ben Skellett. A comparison of neutral landscapes - nk, nkp and nkq. In David B. Fogel, Mohamed A. El-Sharkawi, Xin Yao, Garry Greenwood, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 205–210. IEEE Press, 2002.

[46] S. Gould and N. Eldredge. Punctuated equilibrium comes of age. *Nature*, 366:223–227, November 1993.

[47] G. W. Greenwood and X. Song. How to evolve safe control strategies. In *EH '02: Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware (EH'02)*, page 129, Washington, DC, USA, 2002. IEEE Computer Society.

[48] Steven Gustafson. *An Analysis of Diversity in Genetic Programming*. PhD thesis, School of Computer Science and Information Technology, University of Nottingham, Nottingham, England, February 2004.

[49] Tomas F. Hansen. Is modularity necessary for evolvability? remarks on the relationship between pleiotropy and evolvability. *Biosystems*, 69(2-3):83–94, 2003.

[50] I. Harvey and A. Thompson. Through the labyrinth evolution finds a way: A silicon ridge. In T. Higuchi, M. Iwata, and L. Weixin, editors, *Proc. 1st Int. Conf. on Evolvable Systems (ICES'96)*, volume 1259 of *LNCS*, pages 406–422. Springer-Verlag, 1997.

[51] J.H. Holland. *Adaptation in Natural and Artificial Systems*. M.I.T. Press, 1st edition, 1992.

[52] Martijn A. Huyen. Exploring phenotype space through neutral evolution. *Journal of Molecular Evolution*, 43(3):165–169, September 1996.

[53] Martijn A. Huynen, Peter F. Stadler, and Walter Fontana. Smoothness within ruggedness: the role of neutrality in adaptation. *Proc. Natl. Acad. Sci. (USA)*, 93:397–401, 1996. SFI preprint 95-01-006, LAUR-94-3763.

[54] Nick Jakobi. Encoding scheme issues for open-ended artificial evolution. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – PPSN IV*, pages 52–61, Berlin, 1996. Springer.

[55] R. Jensen. Efficient bdd-based search for planning, 2002. Thesis proposal - citeseer.ist.psu.edu/jensen01efficient.html.

[56] Colin Johnson. Genetic programming with fitness based on model checking. In Marc Ebner, Michael O'Neill, Anikó Ekárt, Leonardo Vanneschi, and Anna Isabel Esparcia-Alcázar, editors, *Proceedings of the 10th European Conference on Genetic Programming*, volume 4445 of *Lecture Notes in Computer Science*, pages 114–124, Valencia, Spain, 11 - 13 April 2007. Springer.

[57] Terry Jones. *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, University of New Mexico, 1995.

[58] Kenneth De Jong. *Evolutionary Computation: A Unified Approach.* MIT Press, 2006.

[59] Motoo Kimura. *The neutral theory of molecular evolution.* Cambridge University Press, 1983.

[60] Marc W. Kirschner and John C. Gerhart. *The Plausibility of Life: Resolving Darwin's Dilemma.* Yale, 2005.

[61] Joshua D. Knowles and Richard A. Watson. On the utility of redundant encodings in mutation-based evolutionary search. In H.-G. Beyer J.-L. Fernández-Villacañas H.-P. Schwefel J.-J. Merelo Guervós, P. Adamidis, editor, *Parallel Problem Solving from Nature - PPSN VII, 7th International Conference, Granada, Spain, September 7-11, 2002. Proceedings*, number 2439 in Lecture Notes in Computer Science, LNCS, page 88 ff. Springer-Verlag, 2002.

[62] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA, 1992.

[63] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs.* MIT Press, Cambridge Massachusetts, May 1994.

[64] John R. Koza, Forrest H. Bennett III, David Andre, and Martin A. Keane. *Genetic Programming III: Darwinian Invention and Problem Solving.* Morgan Kaufmann, 1999.

[65] Matthias Krause, Petr Savický, and Ingo Wegener. Approximations by OBDDs and the variable ordering problem. In *Proc. 26th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1644 of *LNCS*, pages 493–502, 1999.

[66] Ulrich Kühne and Nicole Drechsler. Finding compact bdds using genetic programming. In *EvoWorkshops*, pages 308–319, 2006.

[67] Sanjeev Kumar and Peter Bentley, editors. *On Growth, Form and Computers*. Academic Press, 2003.

[68] W. B. Langdon and R. Poli. Fitness causes bloat. Technical Report CSRP-97-09, University of Birmingham, School of Computer Science, Birmingham, B15 2TT, UK, 24 February 1997.

[69] W. B. Langdon and Riccardo Poli. *Foundations of Genetic Programming*. Springer-Verlag, 2002.

[70] C.Y. Lee. Representation of Switching Circuits by Binary-Decision Programs. *Bell Systems Technical Journal*, 38:985–999, July 1959.

[71] Wolfgang Lenders and Christel Baier. Genetic algorithms for the variable ordering problem of binary decision diagrams. In *FOGA*, pages 1–20, 2005.

[72] R. E. Lenski, C. Ofria, R. T. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423(6936):139–144, May 2003.

[73] Richard E. Lenski, Jeffrey E. Barrick, and Charles Ofria. Balancing robustness and evolvability. *PLoS Biology*, 4(12):e428+, December 2006.

[74] Hod Lipson, Jordan B. Pollacl, and Nam P. Suh. On the origin of modular variation. *Evolution*, 56(8):1549–1556, 2002.

[75] Sean Luke. *Issues in Scaling Genetic Programming: Breeding Strategies, Tree Generation, and Code Bloat*. PhD thesis, Department of Computer Science, University of Maryland, A. V. Williams Building, University of Maryland, College Park, MD 20742 USA, 2000.

[76] Sean Luke and Liviu Panait. Fighting bloat with nonparametric parsimony pressure. In Juan J. Merelo-Guervos, Panagiotis Adamidis, Hans-Georg Beyer, Jose-Luis Fernandez-Villacanas, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VII*, number 2439 in Lecture Notes in Computer Science, LNCS, pages 411–421, Granada, Spain, 7-11 September 2002. Springer-Verlag.

[77] Ernst Mayr. *What Evolution is*. Phoenix, 2002.

[78] Nicholas Freitag McPhee and Justin Darwin Miller. Accurate replication in genetic programming. In L. Eshelman, editor, *Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95)*, pages 303–309, Pittsburgh, PA, USA, 15-19 July 1995. Morgan Kaufmann.

[79] Zbigniew Michalewicz and David B. Fogel. *How to Solve It: Modern Heuristics*. Springer-Verlag, New York, 2000.

[80] Julian Miller. What bloat? cartesian genetic programming on boolean problems. In Erik D. Goodman, editor, *2001 Genetic and Evolutionary Computation Conference Late Breaking Papers*, pages 295–302, San Francisco, California, USA, 9-11 2001.

[81] Julian F. Miller. An empirical study of the efficiency of learning boolean functions using a cartesian genetic programming approach. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1135–1142, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.

[82] Julian F. Miller, Dominic Job, and Vesselin K. Vassilev. Principles in the evolutionary design of digital circuits-part II. *Genetic Programming and Evolvable Machines*, 1(3):259–288, July 2000.

[83] Julian F. Miller and Stephen L. Smith. Redundancy and computational efficiency in cartesian genetic programming. *IEEE Transactions on Evolutionary Computation*, 10(2):167–174, April 2006.

[84] Julian F Miller and Peter Thomson. Cartesian genetic programming. In R. Poli, W. Banzhaf, W. B. Langdon, J. Miller, P. Nordin, and T. C. Fogarty, editors, *Proceedings of the Third European Conference on Genetic Programming (EuroGP-2000)*, volume 1802 of *LNCS*, pages 121–132, Edinburgh, Scotland, 2000. Springer Verlag.

[85] M E J Newman and Robin Engelhardt. Effects of neutral selection on the evolution of molecular species. *PROC.R.SOC.LONDON B*, 265:1333, 1998.

[86] Jens Niehaus and Wolfgang Banzhaf. More on computational effort statistics for genetic programming. In Conor Ryan, Terence Soule, Maarten Keijzer, Edward Tsang, Riccardo Poli, and Ernesto Costa, editors, *Genetic Programming, Proceedings of EuroGP'2003*, volume 2610 of *LNCS*, pages 164–172, Essex, 14-16 April 2003. Springer-Verlag.

[87] Peter Nordin. A compiling genetic programming system that directly manipulates the machine code. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 14, pages 311–331. MIT Press, 1994.

[88] Peter Nordin and Wolfgang Banzhaf. Complexity compression and evolution. In L. Eshelman, editor, *Genetic Algorithms: Proceedings of the*

Sixth International Conference (ICGA95), pages 310–317, Pittsburgh, PA, USA, 15-19 1995. Morgan Kaufmann.

[89] Michael O'Neill and Conor Ryan. *Grammatical Evolution: Evolutionary Automatic Programming in a Arbitrary Language*, volume 4 of *Genetic programming*. Kluwer Academic Publishers, 2003.

[90] Shipra Panda and Fabio Somenzi. Who are the variables in your neighborhood. In *ICCAD '95: Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design*, pages 74–77, Washington, DC, USA, 1995. IEEE Computer Society.

[91] William B. Provine. *Sewall Wright and Evolutionary Biology*. Series: Science and its Conceptual Foundations. The University of Chicago Press, Chicago, 1986. Series Editor: David L. Hull.

[92] N. J. Radcliffe and P. D. Surry. Fitness variance of formae and performance prediction. pages 51–72. Morgan Kaufmann, 1995.

[93] Rudolf A. Raff. *The Shape of Life: Genes, Development, and the Evolution of Animal Form*. University Of Chicago Press, 1996.

[94] Joseph Reisinger, Kenneth O. Stanley, and Risto Miikkulainen. Towards an empirical measure of evolvability. In *Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO-2005)*, Washington, D.C., June 25-29 2005. ACM Press.

[95] Bernd Becker Rolf Drechsler. *Binary Decision Diagrams: Theory and Implementations*. Kluwer Academic, 1998.

[96] Justinian P. Rosca and Dana H. Ballard. Genetic programming with adaptive representations. Technical Report TR 489, Rochester, NY, USA, 1994.

[97] Franz Rothlauf and David E. Goldberg. Redundant representations in evolutionary computation. *Evolutionary Computation*, 11(4):381–415, 2003.

[98] R. Rudell. Dynamic variable ordering for ordered binary decision diagrams. In *Proceedings of the International Conference on CAD (ICCAD)*, pages 42–47, 1993.

[99] Rolf Drechsler Rüdiger Ebendt, Görschwin Fey. *Advanced BDD Optimization*. Springer, 2005.

[100] Thomas Philip Runarsson, Hans-Georg Beyer, Edmund K. Burke, Juan J. Merelo Guervós, L. Darrell Whitley, and Xin Yao, editors. *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference, Reykjavik, Iceland, September 9-13, 2006, Procedings*, volume 4193 of *Lecture Notes in Computer Science*. Springer, 2006.

[101] Günter Rundolph. *Evolution Strategies*, chapter B1.3. The Handbook of Evolutionary Computation. The Insitute of Physiscs, 1997.

[102] J. Rutten, M. Kwiatkowska, G. Norman, and D. Parker. *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems,* P. Panangaden and F. van Breugel (eds.), volume 23 of *CRM Monograph Series*. American Mathematical Society, 2004.

[103] Hidenori Sakanashi, Tetsuya Higuchi, Hitoshi Iba, and Yukinori Kakazu. Evolution of binary decision diagrams for digital circuit design using genetic programming. In *International Conference on Evolvable Systems*, pages 470–481, 1996.

[104] J. Schmidhuber. Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook. Diploma thesis, 1987. Institut f. Informatik, Tech. Univ. Munich.

[105] Peter Schuster, Walter Fontana, Peter F Stadler, and Ivo L Hofacker. From sequences to shapes and back: A case study in RNA secondary structures. *Proc. Roy. Soc. Lond. B*, 255:279–284, 1994.

[106] C. E. Shannon. A symbolic analysis of relay and switching circuits. *Transactions of AIEE*, 57:713–723, 1938.

[107] D. Sieling. On the existence of polynomial time approximation schemes for OBDD-Minimization. *LNCS*, 1373:205–215, 1998.

[108] P. W. H. Smith. Controlling code growth in genetic programming. In Robert John and Ralph Birkenhead, editors, *Advances in Soft Computing*, pages 166–171, De Montfort University, Leicester, UK, 2000. Physica-Verlag.

[109] S. F. Smith. *A Learning System Based on Genetic Adaptive Algorithms*. PhD thesis, University of Pittsburgh, 1980.

[110] Tom Smith, Phil Husbands, and Michael O'Shea. Neutral networks and evolvability with complex genotype-phenotype mapping. In J. Kelemen and P. Sosík, editors, *ECAL 2001*, volume 2159 of *Lecture Notes in Computer Science*, pages 272–281, Berlin Heidelberg, 2001. Springer-Verlag.

[111] Terence Soule. *Code Growth in Genetic Programming*. PhD thesis, University of Idaho, Moscow, Idaho, USA, 15 May 1998.

[112] Terence Soule and James A. Foster. Removal bias: a new cause of code growth in tree based evolutionary programming. In *1998 IEEE International Conference on Evolutionary Computation*, pages 781–186, Anchorage, Alaska, USA, 5-9 May 1998. IEEE Press.

[113] Bärbel M. R. Stadler, Peter F. Stadler, Günter Wagner, and Walter Fontana. The topology of the possible: Formal spaces underlying patterns of evolutionary change. *J. Theor. Biol.*, 213:241–274, 2001. SFI preprint 00-12-070.

[114] Russell K. Standish. Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3(2):167–175, 2003.

[115] Walter Alden Tackett. *Recombination, Selection, and the Genetic Construction of Computer Programs*. PhD thesis, University of Southern California, Department of Electrical Engineering Systems, USA, 1994.

[116] Marc Toussaint and Christian Igel. Neutrality: A necessity for self-adaptation. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, pages 1354–1359, 2002.

[117] Peter D. Turney. Increasing evolvability considered as a large scale trend in evolution. In Paul Marrow, Mark Shackleton, Jose-Luis Fernandez-Villacanas, and Tom Ray, editors, *Evolvability*, pages 43–46, Orlando, Florida, USA, 13 July 1999.

[118] P. van Remortel, T. Lenaerts, and B. Manderick. pages 249–254, Pasadena, California, 19-21 July. Jet Propulsion Laboratory, California Institute of Technology, IEEE Computer Society.

[119] Vesselin K. Vassilev and Julian F. Miller. The advantages of landscape neutrality in digital circuit evolution. In *ICES*, pages 252–263, 2000.

[120] Vesselin K. Vassilev and Julian F. Miller. Embedding landscape neutrality to build a bridge from the conventional to a more efficient three-bit multiplier circuit. In Darrell Whitley, David Goldberg, Erick Cantu-Paz, Lee Spector, Ian Parmee, and Hans-Georg Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, page 539, Las Vegas, Nevada, USA, 10-12 July 2000. Morgan Kaufmann.

[121] Andreas Wagner. *Robustness and Evolvability in Living Systems*. Princeton Studies in Complexity. Princeton University Press, 2005.

[122] Günter P. Wagner and Lee Altenberg. Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.

[123] James Alfred Walker and Julian Francis Miller. Evolution and acquisition of modules in cartesian genetic programming. In Maarten Keijzer, Una-May O'Reilly, Simon M. Lucas, Ernesto Costa, and Terence Soule, editors, *Genetic Programming 7th European Conference, EuroGP 2004, Proceedings*, volume 3003 of *LNCS*, pages 187–197, Coimbra, Portugal, 5-7 April 2004. Springer-Verlag.

[124] James Alfred Walker and Julian Francis Miller. Investigating the performance of module acquisition in cartesian genetic programming. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantu-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llora, Spiros Mancoridis, Martin Pelikan, Guenther R. Raidl, Terence Soule, Andy M. Tyrrell, Jean-Paul Watson, and Eckart Zitzler, editors, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 2, pages 1649–1656, Washington DC, USA, 25-29 June 2005. ACM Press.

[125] Wikipedia. Intelligent design — wikipedia, the free encyclopedia, 2007. [Online; accessed 13-July-2007].

[126] John R. Woodward. Modularity in genetic programming. In Conor Ryan, Terence Soule, Maarten Keijzer, Edward Tsang, Riccardo Poli, and Ernesto Costa, editors, *Genetic Programming, Proceedings of EuroGP'2003*, volume 2610 of *LNCS*, pages 258–267, Essex, 14-16 April 2003. Springer-Verlag.

[127] Sewall Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of the Sixth International Congress of Genetics* 1. 1932.

[128] Masayuki Yanagiya. Efficient genetic programming based on binary decision diagrams. In *1995 IEEE Conference on Evolutionary Computation*, volume 1, pages 234–239, Perth, 29 November - 1 December 1995. IEEE Press.

[129] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Trans. Evolutionary Computation*, 3(2):82–102, 1999.

[130] Tina Yu and Julian Miller. Neutrality and the evolvability of boolean function landscape. In Julian F. Miller, Marco Tomassini, Pier Luca Lanzi,

Conor Ryan, Andrea G. B. Tettamanzi, and William B. Langdon, editors, *Genetic Programming, Proceedings of EuroGP'2001*, volume 2038 of *LNCS*, pages 204–217, Lake Como, Italy, 18-20 April 2001. Springer-Verlag.

[131] Tina Yu and Julian Miller. Finding needles in haystacks is not hard with neutrality. In J. A. Foster, E. Lutton, J. Miller, C. Ryan, and A. G. B. Tettamanzi, editors, *Proceedings of the Fifth European Conference on Genetic Programming (EuroGP-2002)*, volume 2278 of *LNCS*, pages 13–25, Kinsale, Ireland, 2002. Springer Verlag.

[132] Tina Yu and Julian Francis Miller. Through the interaction of neutral and adaptive mutations, evolutionary search finds a way. *Artif. Life*, 12(4):525–551, 2006.