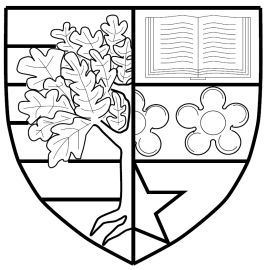
Towards the Control of Cell States in Gene Regulatory Networks by Evolving Boolean Networks

Nadia Solime Taou



submitted for the degree of
Doctor of Philosophy

Heriot-Watt University
School of Mathematical and Computer Sciences

July 2018

Abstract

Biological cell behaviours emerge from complex patterns of interactions between genes and their products, known as gene regulatory networks (GRNs). More specifically, GRNs are complex dynamical structures that orchestrate the activities of biological cells by governing the expression of mRNA and proteins. Many computational models of these networks have been shown to be able to carry out complex computation in an efficient and robust manner, particularly in the domains of control and signal processing. GRNs play a central role within living organisms and efficient strategies for controlling their dynamics need to be developed. For instance, the ability to push a cell towards or away from certain behaviours, is an important aim in fields such as medicine and synthetic biology. This could, for example, help to find novel approaches in the design of therapeutic drugs. However, current approaches to controlling these networks exhibit poor scalability and limited generality. This thesis proposes a new approach and an alternative method for performing state space targeting in GRNs, by coupling an artificial GRN to an existing GRN. This idea is tested in simulation by coupling together Boolean networks that represent controlled and controller systems. Evolutionary algorithms are used to evolve the controller Boolean networks. Controller Boolean networks are applied to a range of controlled Boolean networks including Boolean models of actual biological circuits, each with different dynamics. The results show that controller Boolean networks can be optimised to control trajectories in the target networks. Also, the approach scales well as the target network size increases. The use of Boolean modelling is potentially advantageous from an implementation perspective, since synthetic biology techniques can be used to refine an optimised controller Boolean network into an *in vivo* form, which could then control a genetic network directly from within a cell.

Acknowledgements

I would like to express my gratitude to my supervisors Dr. Michael Lones and Prof. David Corne, particularly to Dr. Michael Lones for all his advice and guidance throughout the years of my study. Thanks to Dr. Marta Vallejo for her friendship, support and advice, I cannot thank you enough for all you have done for me, since my first day in this lab. Thanks to my friends for all their encouragement.

Finally, I am specially very grateful and thankful to my parents, my siblings and to Heiko for their enduring love, support and encouragement. You never give up on me, I could not have made it this far without you.

Declaration

Some of the research within this thesis has previously been published by the author Taou et al. (2016a), Taou et al. (2016b), Taou et al. (2018) and Taou and Lones (2018). All work reported in this thesis as original is so to the best knowledge of the author. All research or work from other researchers which have contributed to this thesis have been referenced and acknowledged as appropriate.

Table of Contents

List of Tables

List of Figures

Hypothesis

This research takes inspiration from the idea that artificial gene regulatory networks can exhibit a wide range of useful properties and behaviours found in biological systems, and that their dynamical behaviours can be controlled by designing appropriate control strategies. More specifically, it is asserted that:

- Gene regulatory networks are biological processes that govern (i.e. control) the activities of biological cells in living organisms based on internal and external environmental factors.
- Gene regulatory networks' properties, functions and behaviours are the result of an evolutionary process.
- Artificial gene regulatory networks can be designed and optimised using evolutionary algorithms.
- A number of computational approaches, such as artificial immune systems and artificial biochemical networks, have been developed by taking inspiration from natural networks. These computational models can capture useful biological properties, for instance robustness, adaptability and self organisation, when they are evolved to carry out control tasks in complex systems.

Following these assertions, it is hypothesised that artificial gene regulatory networks can be optimised to perform efficient control in computational models of gene regulatory networks.

Chapter 1

Introduction

1.1 Overview

Biological cells are the fundamental structural and functional units of all living organisms. They interact with each other and with their environnent. In general these interactions are governed by biological networks, for instance protein-protein interaction networks and protein-mediated networks of biological reactions, which regulate the behaviours of biological cells. Biological networks, because of their ubiquity, their profound effects upon human lives and livelihood, and their ability to control and respond to complex non-linear dynamics, have been widely studied. There has been an increasing interest in recent years in controlling the dynamical behaviour of biological networks (Gates and Rocha, 2016). However, complex biological networks have a range of emergent properties, such as fault tolerance, adaptability and robustness, which make them intrinsically difficult to control (Motter, 2015).

Much of the existing research in controlling computational models of biochemical networks has focused on analytical methods, typically making use of conventional control theoretic approaches. This has resulted in control methods that can be applied to certain types of complex biochemical networks, for instance those with particular relationships between their structure and dynamics (Gates and Rocha, 2016), and those with restricted topologies (Cheng and Qi, 2009). The existing

control methods are computationally expensive, require certain knowledge (which is not always available) about the target networks and cannot efficiently control large set of nodes because of their complexity (complex mathematical equations). With these disadvantages there is a need to develop more general and efficient control strategies for controlling biological networks.

This thesis aims to explore how computational models of biochemical networks can be used to design more biologically realistic controllers which can efficiently control gene regulatory networks.

1.2 Artificial Gene Regulatory Networks

Artificial gene regulatory networks (AGRNs) are computational models which mimic genetic interactions occurring within biological cells. They can be classified into two different categories. The objective for the first is to model gene regulation processes to better understand their functionality. The second aims to develop abstract models which are able to capture biological behaviours and properties of gene regulation, and apply these for use within artificial systems (or *in silico*).

AGRNs comprise a set of interconnected nodes. These model genes as abstract computational components (or units) which are able to take inputs, process them and generate an output. These networks are capable of capturing particular properties of the biological systems from which they are inspired, for instance adaptability, robustness and self-organisation. Because of their functionality, AGRNs are becoming popular models within various areas such as optimisation and the control of systems which can express complex non-linear dynamics (Lones et al., 2010), (Lones et al., 2014).

1.3 Evolutionary Algorithms

Evolutionary algorithms are biologically-motivated computational techniques which are used to optimise data structures for solving a wide range of computational problems. In this thesis evolutionary algorithms, especially genetic algorithms, are

used to evolve solutions to the problem of biological control. Genetic algorithms are adaptive metaheuristic search algorithms based on the evolutionary principles of natural selection and genetics, which are used to artificially evolve possible solutions towards a particular target. Specifically, genetic algorithms are used to optimise artificial gene regulatory networks, in the form of Boolean networks, within this thesis.

1.4 Evolvability

Evolvability is the relative ability of a system to exhibit appropriate change. In biological organisms, evolvability is defined as the relative probability that arbitrary genetic change can lead to an improvement in the organism's behaviour during adaptive evolution. This is due to a continual evolutionary process where genetic variations integrate new functions and increase their adaptability to their environment (Marijuán et al., 2013). Evolvability can be seen in different areas, for example, biological organisms are designed in a compartmental style, which enable individual compartments to be evolved separately reducing the propagation of genetic errors within the entire organism (Conrad, 1990). The work reported in this thesis is widely inspired by the control structures of biological cells. But also, by their evolvability.

1.5 Boolean Networks

A Boolean network is a type of artificial gene regulatory network. Different variants of Boolean networks such as random Boolean networks and scale free Boolean networks have been used in this thesis. It is a generic analogue of gene regulatory networks which can reproduce their properties and functionalities. The advantages of using Boolean networks in this work are their relative ease to be refined into synthetic biology circuits, and their speed of execution (in the context of fitness evaluation).

1.6 Contributions

This work makes the following principle contributions to knowledge:

- The development and understanding of a new computational control method for controlling artificial gene regulatory networks using evolutionary algorithms.
- The demonstration that artificial gene regulatory networks can be used to solve complex control problems.
- The demonstration that random Boolean networks can be optimised to control a number of Boolean network variants.
- The demonstration that the proposed control method can be used to control steady states (or attractors) in Boolean models of real biological networks.
- The usage of multi-objective evolutionary algorithms to minimise the number of interventions required to exert control.
- The realisation that the control approach has the advantage to scale well as the controlled network's size increases.

1.7 Thesis Structure

This thesis is organised into three main sections. Chapter 2 introduces the biological background on which this thesis is based. Chapters 3-4 describe the field of artificial gene regulatory networks and the evolutionary algorithms used to evolve the controller networks to carry out control tasks. Chapters 5-7 present the main contributions of this thesis. They describe the application of controller Boolean networks to a range of control tasks, and the conclusions that can be drawn from this work. More precisely:

Chapter 1 is the introduction of this thesis.

Chapter 2 describes the fundamental biological structures and their functions.

Chapter 3 reviews the different kinds of evolutionary algorithm.

Chapter 4 reviews artificial gene regulatory network models, their abilities to capture biology and their computational properties. It also contains a review of other related work.

Chapter 5 Presents methodology, experimental results and analysis of the control of the dynamics of Boolean networks using Boolean networks.

Chapter 6 Presents experimental results and analysis of application of the controller Boolean networks to the control of dynamics within Boolean models of actual biological networks.

Chapter 7 Summarises the work conducted in this thesis, drawing conclusions, discussion and proposing future ideas for research.

1.8 List Of Publications from this Work

- **N. S. Taou**, D. W. Corne and M. A. Lones, "Towards Intelligent Biological Control: Controlling Boolean Networks with Boolean Networks."
  Proceedings EvoComplex 2016, LNCS 9597, Porto, March 2016

- **N. S. Taou**, D. W. Corne and M. A. Lones, "Evolving Boolean Networks for Biological Control: State Space Targeting in Scale Free Boolean Networks."
  Proceedings IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) 2016, October 2016

- **N. S. Taou** and M. A. Lones, "Towards In Vivo Genetic Programming: Evolving Boolean Networks to Determine Cell States." accepted to EuroGP 2018

- **N. S. Taou**, D. W. Corne and M. A. Lones, "Investigating the use of Boolean

Networks for the Control of Gene Regulatory Networks."

Journal of Computational Science, April 2018

Chapter 2

Structures And Functions Of Biological Systems

This thesis presents a method of controlling artificial gene regulatory networks (AGRNs). A gene regulatory network (GRN) is a group of genes and their products that interact with each other and with other components of biological cells. GRNs play a crucial role within living organisms. Designing control strategies for the control of trajectories in GRNs requires to understand the underlying biological processes, functionalities and structures. A biological system's activities can be observed and described at different levels in the organism, from the interactions between chemical components known as biochemicals (lower levels) up to interactions at high levels of complex structures such as cells, biochemical networks, and interactions with all the non-living factors and processes in the ecosystem known as the abiotic environment. Abiotic factors such as pollution can affect biological systems and cause diseases. The aim of this chapter is to show how highly complex biological behaviours emerge from interactions between small fundamental elements. The chapter is divided into two. The first part describes fundamental biological elements and processes which are related to gene regulation. The second part presents a general overview of the structures, properties and behaviours of GRNs.

2.1 Genes

Mendel (1865) defined a gene as a physical and functional unit of hereditary information in every living organism. A gene is an extension of a polypeptide chain encoding at least one protein, a portion of DNA that can be transcribed by RNA. In human beings, gene size varies from a hundred DNA bases to more than 2 million bases (Sarkar and Plutynski, 2010). Each organism is made of genes held in its genetic structure. The Human Genome Project has estimated that humans have between 20,000 and 25,000 genes. In all organisms genes are very similar in their structure. This enable genes which are not generated naturally within the organism to be artificially introduced and they will still work perfectly (Lai et al., 2002).

Gene genetic structure differs notably between prokaryotes and eukaryotes, even though they are structured similarly, especially a portion of DNA indicating a sequence of amino acids. The major difference between these two categories of organisms is that prokaryotes which are single cells (unicellular) do not contain any internal membrane structures (or organelles) such as nucleus but rather contain nucleoid (a cell that has double stranded DNA) and one single chromosome (see figure 2.1). While eukaryotes are multi-cellular organisms which contain membrane-bound organelles, such as the nucleus and chromatin which is used to wrap up DNA into the nucleus of a cellular organism. The absence of nucleus reduces the complexity in prokaryotes. DNA in prokaryotes are arranged in a structure of operons, in which a group of genes are all transcribed together in a single regulatory prompter (Dworkin, 2006), (Fletcher and Hickey, 2012), (Miller, 1980), is showing an example of the lac operon (Purdom and Anderson, 2008).

Also, in prokaryotes, a single mRNA can encode various proteins, which can be produced independently by a ribosome. Operons have many advantages and one of them is their ability to achieve regulatory circuits through single DNA transcription. But, under the influence of environmental perturbations, the transcription of all the genes together by operons might not be optimal. Eukaryotes structure and functionalities are different than prokaryotes genetic operation because, their genome

contains a large number of non-proteins coding DNA. Also, eukaryotes ribosomes are only synthesised at the beginning of the mRNA strand, implying that eukaryotes mRNA can only be transcribed from a single gene. The relation between the number of non-proteins coding DNA and protein coding within the genome in prokaryotes is linear (Ahnert et al., 2008), while in eukaryotic cells this is not the case because gene regulation is very complex.

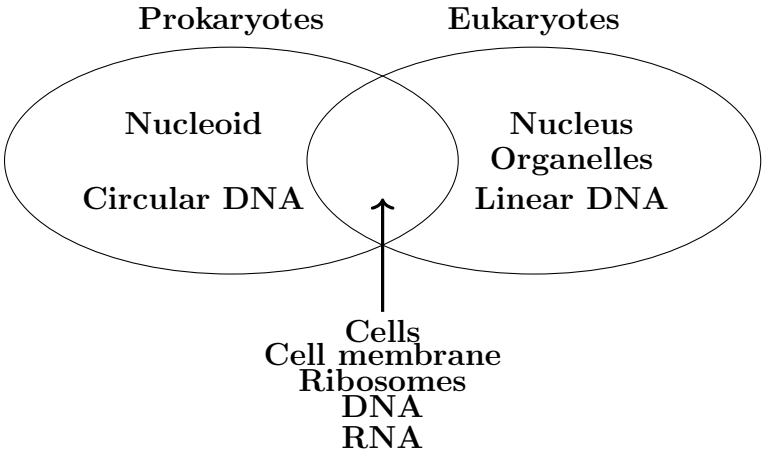


Figure 2.1: Difference and similarities of prokaryotes and eukaryotes. Adapted from (Mattick, 2001).

2.2 Cells

Cells are the fundamental biological unit of almost all living organisms. They are formed of cytoplasm, a fluid enclosed inside a cell membrane, which contains a large number of biological molecules such as nucleic acids and proteins (Alberts et al., 2002). A cell is the smallest unit of biological organisms 1 to 100 micrometres (Alberts et al., 2002) that can replicate autonomously. In humans, the number of cell is more than 10 trillion (10^{12}) (Alberts et al., 2002). The combination of cells metabolism which is the sum of all chemical reactions occurring in the cell and cell communication which is the information transmission between cells lead to emergent behaviour of an organism. Cells are biological components which are used to build organism. Cells can be classified into two main classes: prokaryotic and eukaryotic (Lodish et al., 2008).

2.3 Biological Activities

2.3.1 Gene Regulation

Gene regulation is the process through which biological organisms control (increase or decrease) the rates of gene expression in order to maintain an optimum state. Gene regulation networks vary in complexity and size and have a wide range of different levels of abstraction. Nowadays, most of the smallest gene regulatory circuits have been wholly mapped and well studied, a well known example is the lac operon (Jacob and Monod, 1961), found in the bacteria *Escherichia coli*. The lac operon is created to metabolise lactose dynamically, according to the quantity of lactose in the environment. In this process, the control mechanism is done in two ways, the first in response to lactose and the second in response to glucose. These two control strategies enable the *Escherichia coli* to control with extreme accuracy, the expression of certain genes in accordance with the environment.

Gene regulation is made of a wide range of interconnected biological structures and mechanisms. This process can be affected by biological structures such as repressors, RNA editing, RNA interference and transcription factors. These biological structures and processes are highly connected with complex epigenetic structures for example chromatin modifications, which have a important effect on gene regulation. The emergent properties of gene regulation is due to the joint work between all the biological structures at different degrees of complexity. The genome regulatory information process is hard to compile because of the wide range of other factors involved in gene regulation mechanisms such as cell signalling, metabolism and environmental changes.

2.4 Biochemical Networks

Biochemical networks arise from protein-protein and protein-mediated molecular interactions occurring within cells. They interact to form high level structures which can exhibit complex dynamical behaviours. Biochemical networks underlie

the structural and functional complexity existing within biological living organisms (Lones et al., 2010). These networks are omnipresent throughout biological processes, operating on small to large scales, through communication between elements of the same species. Research such as (Bhalla and Iyengar, 1999), have made the assumption that emergent complex properties of biochemical networks occur entirely due to the emergent property of the underpinning elements. They are also thought to be used as computational components in living cells (Bray, 1995).

In general, three kinds of biochemical network can be observed within a living organism, metabolic networks, gene regulatory networks and signalling networks. Even though they are isolated networks, they are coupled and constantly interact with each other (see Figure 2.2). The gene regulatory network changes the behaviour of metabolic and signalling networks by regulating protein production. Signalling networks deliver chemical signals to various cellular locations; this regulates the behaviours of both metabolic and genetic networks. Within a cell's components, these interactions enable the cell's metabolism to be reconfigured for various environments.

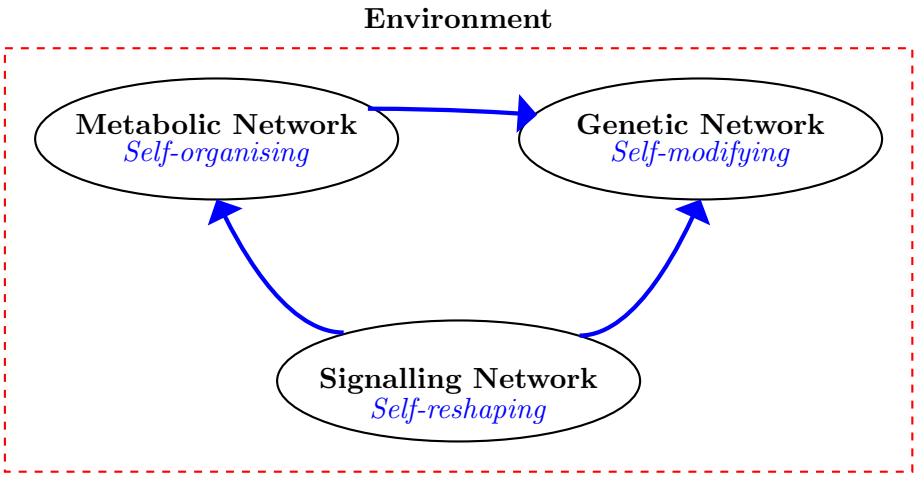


Figure 2.2: Interaction between biochemical networks, adapted from (Lones et al., 2010).

2.4.1 Gene Regulatory Networks

Gene regulatory networks are discrete dynamical networks which are the result of gene interactions within them and with their environment. The regulation of these interactions involved in protein synthesis and protein function determine gene expression levels within a cell. It is not easy to capture the regulatory nature of the

cell due to the very large number of possible steps. A protein can bind to another protein, excite or inhibit DNA exposure, it can bind to a protein's allosteric site (which allow molecules to either activate or inhibit, or turn off, enzyme activity), it can modify structure and change the environment. All these activities going on constantly, makes it hard to model. Gene regulatory networks size vary enormously (see Figure 2.3).

Gene regulatory networks exist over many abstract levels making their modelling difficult. Simple systems can be entirely understood, but when they interact with other simple systems to generate behaviours that interact with large systems, the networks become very complex and abstract, therefore difficult to model and control. However, all gene regulatory networks have specific properties such as self-organisation, self-adaptivity, dynamical behaviours, robustness and evolution over time, which arise from underlying components like genes. This evolution is described in (Shimeld, 1999) as gene duplication events and in (Bornholdt and Sneppen, 1998), (Bornholdt and Rohlf, 2000) artificial evolution is used to evolve connectivity in dynamical networks (for instance Boolean networks). Having all this information, scientists have attempted to model genetic networks using automata and to analyse them by simulation. Random Boolean networks were proposed by Stuart Kauffman as a model of gene regulatory networks (Kauffman, 1969).

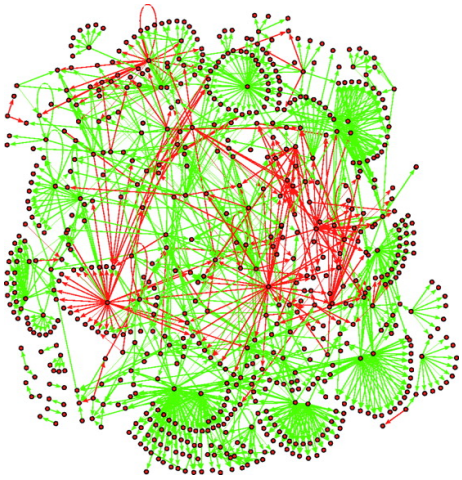


Figure 2.3: Simplified representation of the yeast regulatory network. Two types of interactions are observed, being activated (green) or repressed (red), and the dynamical components representing the gene states are taken to be binary with values 0 and 1. Adapted from (Maslov et al., 2003).

2.4.2 Metabolic Networks

Metabolic networks emerge from complex interactions between biochemicals. Their role is to facilitate the production of chemical products through a number of interdependent pathways. A metabolic pathway is formed of two intertwined flows: reaction pathways and control feedbacks. Reaction pathways consist of a group of enzymes with firmly linked specificities for one another's products and substrates. A control feedback can be internal to the pathway, positive or negative, due to external metabolic or signalling pathways. Metabolic networks contribute to the synthesis of products which cannot be found in the environment but are essential to every living organism.

A good example of this is the citrate cycle, a fundamental process of the metabolism of eukaryotes and prokaryotes, though both of them are not always used in the same way (Forst and Schulten, 1999). The citrate cycle, also know as Krebs cycle, is part of the respiratory chain which is an extensive structure, a series of reactions which transform food into energy. Also, it generates intermediates required in the amino acid synthesis. Usually most of the pathways are not separate entities since they share reactions and control between them. But, there are some pathways or parts of pathways which are independent of others, known as genetically independent pathways (which means the enzymes in this pathway define an independent genotype) (Schilling et al., 1999).

A good understanding of metabolic networks has impact in various fields. For instance, in bio-medical (or other heath-related areas) this knowledge contributes towards the design of novel therapeutic drug interventions for treating diseases, and in bio-process engineering, by creating new metabolic pathways which can be used in chemical plants (Karp and Mavrovouniotis, 1994). In addition, research in different fields such as biochemistry, molecular evolution and biophysics have shown that a better understanding of how the metabolism is evolved will help to understand metabolic networks (Kurganov et al., 1997), (Ashbrook, 1993), (Igamberdiev et al., 1999), (Forst and Schulten, 1999).

2.4.3 Cell Signalling Networks

Cell signalling networks are two-directional communication processes between cells and their environment. They drive the basic activities of cells and coordinate cell actions. For example, they take an internal signal and spread the signal outside the cell wall. Kholodenko (2006) has shown that cell signalling networks take an external signal from the environment and sense that signal by using plasma membrane receptors and receptor tyrosine kinases. These external signals are taken from the environment, into the cells, processed and then moved to gene regulatory networks in which gene expression values can adapt to ensure the cell's activity is well suited to its environment. Very often this process involves metabolic networks. Cell signalling networks are encoders, information processors and integrators rather than just a transmit and receive model. According to some research (Hoffmann et al., 2002), (Kholodenko, 2006), different spacial temporal activations of the same repertoire of signalling leads to distinct protein pathways being activated.

2.5 Biological System Properties

Biological systems are complex networks formed by groups of organs which work together to perform biological activities in living organisms. The aim of this section is to highlight and describe the underling principles and properties of these complex biological systems. Understanding biological system principles helps to build more realistic computational models of biological systems.

2.5.1 Evolution

Evolution is the process that generate changes over consecutive generations. In the biosphere, many structures such as glaciers, oceans and rocks (granite) change over time through evolutionary processes. These structures' current state is considered to have evolved. In this chapter, the focus is on the biological evolution of populations the theory of which was introduced by Darwin (1859). The theory of biological evolution explains changes that occur in traits of an individual or a species over

time, creating the current diversification of species in the world. It also describes the evolution (or progression) of positive traits in a population from one generation to another.

Usually, during the evolution process most of the positive characteristics are conserved within a population (Dawkins, 1976). This gives a strong foundation to the theory of natural selection, which is the process through which organisms well adapted to their environment try to survive and produce more offspring. The ability to reproduce depends on how good organisms are at surviving, living and passing on their genetic constitution (especially DNA) contained in their genetic structure. An organism's genetic structure is a biological encoding of its phenotype, and the best surviving phenotypes will pass on their DNA to their offspring.

For a system to be biologically evolved, three elements are usually required: the organism, its genetic representation and the processes through which changes or modifications can occur. Changes in an organism can happen in many different ways, such as mutation and genetic recombination, but the most ubiquitous in all organisms is mutation. Mutation is a change in the genetic representation of an organism (mainly in its DNA sequence), due to either errors when the DNA is copied, interactions with viruses, or physical damage in the DNA structure. Also, environmental effects such as air pollution can induce mutation.

Mutation is a mechanism that produces in the next population new phenotypes and genetic data. In (Wilson et al., 2011), (Lenski, 2010), the authors have shown that this mutation process is used by bacteria to create resistance to antibiotics in a short time. Another way to produce changes is genetic recombination, occurring generally in two principal types: vertical and horizontal gene transfer.

Vertical gene transfer or inheritance (see figure 2.4) is the process of transferring genetic material from parents to offspring, through sexual or asexual reproduction. In general vertical transfer is linked to eukaryotes and is a way of creating viable phenotypes with some genetic material from both parents. The aim of this mechanism is to produce offspring which have a high probability to survive as they have almost the same traits as their parents; however, they are not identical. The result advantage

of vertical gene transfer is that over successive generations, the latest generation will be fitter than the previous. This can be explained by the fact that individuals tend to adapt continually, becoming better optimised within their environment.

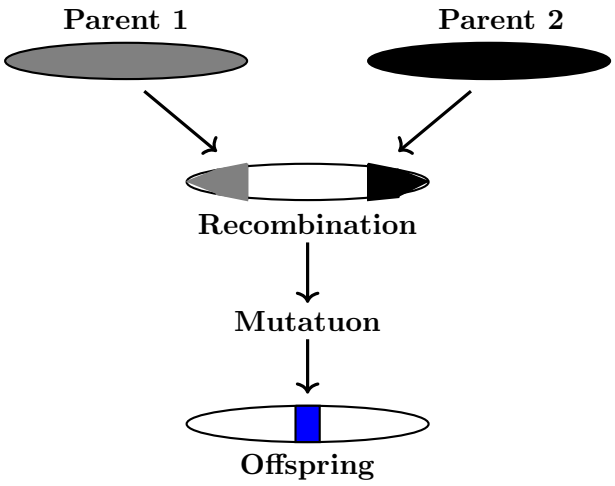


Figure 2.4: Vertical gene transfer from parents to an offspring, or child. This child is generated after sexual reproduction or recombination and mutation processes and has both parents' genetic traits.

Horizontal gene transfer (or transposition) is the process of exchanging genetic material between different organisms (see figure 2.5). It occurs mainly in prokaryotes, but can also occur in multicellular eukaryotic organisms (Ros and Hurst, 2009). This process can occur through three main mechanisms: transformation, transduction and conjugation. Transformation is a modification of a cell's genetic material due to the uptake and incorporation of small DNA fragments by bacteria. Transduction is the insertion of foreign DNA from one bacterium into another (the host bacterium) via a bacteriophage.

Finally, conjugation is the transfer of genetic data between two bacteria in direct contact. This is done through sexual reproduction, therefore it requires cell to cell contact. Conjugation is especially used by bacteria to transfer DNA with another organism. However, bacteria can integrate DNA directly from the environment. Horizontal gene transfer happens on a smaller time scale compared to vertical gene transfer. Multiple DNA exchange can be observed in bacteria using horizontal gene transfer over one cell division (or one generation). Research such as (Blount et al., 2008), (Cooper et al., 2003), have shown that horizontal gene transfer is an efficient technique to optimise small and less complex organisms.

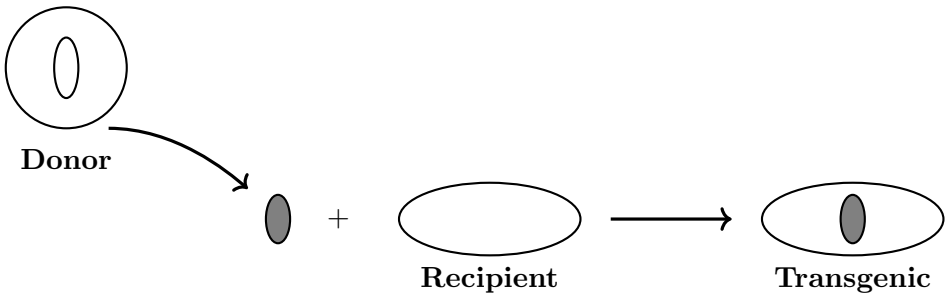


Figure 2.5: Horizontal gene transfer.

2.5.2 Evolvability

Evolvability is an organism's capacity to evolve through natural selection towards future environments (Kirschner and Gerhart, 1998). It is also the ability of an evolutionary system to create adaptive genetic variation (Conrad, 1979), (Colegrave and Collins, 2008). In general systems which are considered to be evolvable are capable of managing change without any major failures. Furthermore, these evolvable systems are able to preserve the existing genetic information (or traits). This shows the ability of evolvability to transfer modification and pass on genetic material from one generation to another. Evolvability can be considered to be an evolved property, since evolvable organisms can easily adapt to new environments and integrate changes in order to survive compared to non-evolvable entities.

2.5.3 Robustness

Robustness is the characteristic that enables a system to preserve its functions and resist internal and external perturbations (Kitano, 2004), (Stelling et al., 2004), (Félix and Wagner, 2008). Like evolvability, robustness is a biological property and both are highly linked to one another (Kitano, 2007). Robustness emerges from complex interactions between networks, organisms, systems and behaviours occurring in real life. This can be seen in a large number of systems and at different degrees of abstraction. In this thesis biologically inspired computational networks, for instance Boolean models of gene regulatory networks, will be created and they have to be both evolvable and robust like real biological networks. Also, in these models a particular effort is made to stay faithful to their biological underpinning. Many

different researchers (Lones, 2004), (Kitano, 2004), (Gershenson et al., 2005) have demonstrated that computational models can be developed to be evolvable and robust.

A number of biological system properties such as decoupling, modularity and redundancy are thought in be fundamental participants to robustness (Lones, 2004), (Kitano, 2004), (Gershenson et al., 2005), (Kitano, 2007). Modularity (formed by modules) is the process used to minimise the effect of perturbations on the entire system (Kitano, 2004). It can often be seen in biological networks such as genetic, metabolic, neural and signalling networks (Newman, 2006). Modules are functioning elements which are detachable from other components. They can be identified by observing functional, topological and evolutionary criteria (Hintze and Adami, 2008). Although modules are important in modularity mechanisms, the failure of modules does not imply a failure of the system. This enables therefore modularity, to be a positive evolutionary property.

Redundancy is the mechanism in which perturbation can affect the organism without units of its functionality. It enables the evolution of a system without any major failure, by allowing redundant components to replace failed ones. However, this does have the disadvantage of extra resource requirements (Kitano, 2007). In biology, decoupling is the mechanism whereby the phenotype of an organism is the result of an indirect representation of that organism. It divides biological activities into two parts: low level variations and high level functionalities (Kitano, 2004). A well known example of this process is the decoupling of phenotype and genotype (Lones, 2004), (Kitano, 2004), (Kitano, 2007). Decoupling helps to allow modification without affecting the system.

2.5.4 Emergence of complex behaviours

Complex biological system behaviours such as evolvability, robustness, complexity and self-organisation are very difficult to implement *in silico*, because they are abstract behaviours. The origin of these abstract behaviours is still not well known and their implementation or transfer into computational models remains a challenging

problem. A number of researchers in complex systems have noted the existence of an emergent property from their constituent elements and the interactions between these elements (Banzhaf, 2004), (Bull, 2012), (Reil, 1999).

This suggests that it is possible to produce desired behaviours without directly coding for them, assuming that they come about through emergence. Gene regulation networks have other properties such, as time and variation in interconnected elements over time which are hard to implement in computational models. This challenging issue can be explained by the fact that biological connections between components of a systems are either connected for a certain time or are not connected at others (**?**), (Holme and Saramäki, 2012). This is an important issue since it brings many restrictions to what can be modelled and at which level of abstraction. Current computational models are not a perfect replication (or copy) of real-life gene regulation.

2.6 Summary

This chapter provides details about biological systems and their functions. It first presents a description of the low level biological components which are the basis of every biological organism, and looks at key biological structures and their sub-circuits. Secondly, it highlights biological component functions and their interactions, particularly the fact that the interactions between biological circuits generate higher order functional structures.

Also, the chapter describes different kinds of biological mechanisms and processes such as gene regulation that occur in living organisms. Then, shows connections between the biology and the computational models discussed in this thesis, by presenting abstract concepts such as emergence of biological properties, evolvability and robustness. These abstract concepts provide a fundamental framework in which computational models used in this thesis are built in order to accurately capture highly complex behaviours within computational simulations.

In the next chapter, artificial gene regulatory networks are discussed, and their architectures and biological inspiration are highlighted.

Chapter 3

Computational Models of Gene Regulatory Networks

The aim of this chapter is to describe techniques used to design computational models of gene regulatory networks. Its also highlights the properties these models have and the advantages and the disadvantages of using them. This will provide a fundamental basis on which to build upon these models in the next chapters.

Gene regulatory networks (GRNs) are complex dynamical structures that orchestrate the activities of biological cells. They play a crucial role in different processes through which biological systems control their dynamics behaviours, their growth, their interactions within the organism and with their environment. Computer scientists inspired by biological complexity were able to design computational models of gene regulatory networks, also known as artificial gene regulatory networks (AGRNs), which can mimic gene's interactions in a cell. A large number of them were built with different degrees of complexity. They are mainly developed to model gene interactions in biological systems in order to better understand them (Karlebach and Shamir, 2008), (Ribeiro et al., 2006). These network models are relatively specific, with many details to make them as similar as possible to their real world analogues. Other AGRNs are designed to be abstract models, which aim to display important biological properties, without any exact modelling details (Aldana et al., 2007), (Kuyucu, 2010), (Lones et al., 2010), (Lones et al., 2011), (Lones, 2016). These

AGRN models are mostly used for modelling the steady state behaviours of biological systems (to study them) or to look at general biological system properties.

This chapter is divided into two sections. In the first section, different types of Boolean networks will be described. These networks were chosen because they are a fair representative of most of the existing AGRNs and also because of their ability to represent each region of the time and space map. The time and space map are a representation of the type of data AGRNs use and how they process these data in terms of time. Time and space can be either continuous or discrete variables in the course of a simulation. Depending on the type of variables a given model uses, the properties of the network can vary. The second section reviews previous research on modelling, controlling and evolving Boolean networks. It also presents some work on implementing Boolean networks in cells and how AGRNs have been used as controllers to solve problems.

3.1 Boolean Networks

Boolean networks (BNs) were originally introduced by Kauffman (1969) as a very simple binary-state computational model of gene regulatory networks. They also have a simple mapping to the kind of digital or numerical circuit models that are often used in synthetic biology. They were inspired by self organisation and stability properties found in randomly generated networks and from Von Neumann's work on cellular automata (Von Neumann et al., 1966), (Burks, 1969). A BN is a discrete-time non-linear dynamical system represented as a directed graph $G(N, E)$ composed of nodes, or vertices, N and edges E (Kauffman, 1969), (Kauffman, 1993). They exist in the discrete time and space domains. Because of their simplicity, BNs have been occasionally criticised (Harvey and Bossomaier, 1997), especially regarding their ability to accurately capture the quantitative dynamics of regulatory circuits.

Despite this criticism, BNs remain very popular, for example they have been used successfully to capture the structure and dynamics of real biochemical networks (Kauffman et al., 2003), (Albert and Othmer, 2003), (Saez-Rodriguez et al., 2007), (Davidich and Bornholdt, 2008), (Veliz-Cuba and Stigler, 2011), (Dallidis and Karafyl-

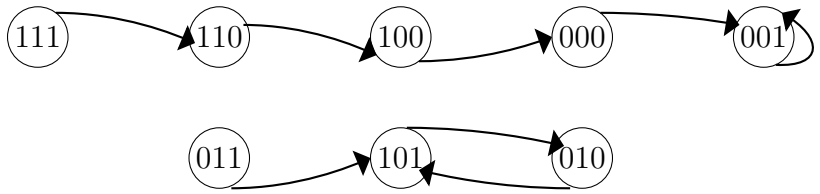


Figure 3.2: State transition graph corresponding to the Boolean network shown in Figure 3.1.

lidis, 2014), (Kaushik and Sahi, 2015), (Saadatpour and Albert, 2013), (Fumiã and Martins, 2013), (Poret and Boissel, 2014). In addition, BNs have been considered as a more general model of complex networks, and studies of their dynamical behaviour have brought significant insight into the properties of real world networks (Aldana, 2003). Three forms of BN are considered in this work: deterministic random Boolean networks, scale-free boolean networks and asynchronous random Boolean networks. Random Boolean networks nodes have uniform connectivity, while scale-free Boolean networks capture the power law distribution of connectivity within biological GRNs.

3.1.1 Random Boolean Networks

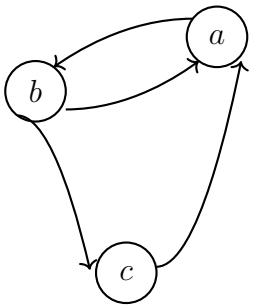


Figure 3.1: An example of a Boolean network with three nodes.

A random Boolean Network (RBN) is a Boolean network which is randomly sampled from a set of possible Boolean networks. This means that, in the network, node inter-connections and Boolean functions associated with each node are randomly generated (Gershenson, 2004), (Drossel, 2008) (see Figures 3.1–3.3). An RBN is formed by a set of N nodes (or genes) called Boolean states which represent the activity level of a gene. Genes can be either active (one) or inactive (zero). The connectivity K in RBN indicates the number of inputs from other genes that are needed by a gene to update its own activity level. From this, a state transition table

abc (t)	abc (t+1)
000	001
001	001
010	101
011	101
100	000
101	010
110	100
111	110

(a) Truth table

$$a^{(t+1)} = b^t$$
$$b^{(t+1)} = a^t \wedge c^t$$
$$c^{(t+1)} = \neg a^t$$

(b) Boolean functions

Figure 3.3: Functions and truth table used by the Boolean network shown in Figure. 3.1.

is generated (see Table 3.3a), showing all the potential combinations for a gene's next state according to its inputs.

The time evolution of a RBN is expressed by a set of Boolean functions f_i, $i = 1, 2, 3, ...$ Each RBN node has a binary state s which is updated synchronously according to its Boolean function and the states of the k input nodes that are connected to it. Formally, $s(t+1) = f_i(s(t))$, where s is a set of network states $s \in \{0,1\}^N$, $t = 0, 1, 2, 3, 4, ...$ is the discrete time, $f_i : \{0,1\}^N \rightarrow \{0,1\}$. Since a RBN is deterministic $s(t+1)$ is only determined by $s(t)$. The possible number of Boolean functions is 2^{2^k}, and the state space is finite and equal to 2^N in size. Each node has $\frac{N!}{(N-K)!}$ possible ordered options for K different connections and the number of possible networks (Harvey and Bossomaier, 1997) is shown in equation 3.1.

$$\left(\frac{2^{2^k} N!}{(N-K)!} \right)^N \tag{3.1}$$

Since the state space is finite, states must eventually be repeated, leading to temporal structures called attractors. An attractor formed by one state is called a point attractor, and when it is formed by at least two states it is known as a cyclic attractor.

During execution, three complex dynamics regimes can be observed in RBNs: *ordered, chaotic* and *critical* (Stepney, 2009). Ordered RBNs have attractors with a relatively short period, repeating the same series of states over and over again. Chaotic RBNs have attractors with long periods, they appear random, even though

they are deterministic. Critical RBNs also have attractors with long periods, but they appear to have a complex internal order which has been associated with computation. In general, the number of attractors grows with the number of nodes (Kauffman, 1969), (Kauffman, 1993), (Bilke and Sjunnesson, 2001). RBNs with $K < 2$ tend to be ordered; those with $K > 2$ tend to be chaotic; critical dynamics tend to be found when $K = 2$ (Gershenson, 2004).

In addition, RBNs exhibit high levels of robustness to a number of perturbations such as gene deletion and gene insertion (Aldana et al., 2007). These properties show that robust and complex systems can be developed based on randomly ordered networks. This concept can be seen throughout the study of biochemical networks and in connectionism which is a way to model emergent processes from the activity of interconnected networks of simple, and non-linear components (Lones et al., 2013c).

3.1.2 Scale Free Boolean Networks

RBNs typically have a fixed connectivity k. Real world complex networks, by comparison, tend to have a scale-free distribution of connectivities. A scale free Boolean network (SFBNs) is a Boolean network with a scale-free distribution, or more precisely a connected graph composed of a set of N nodes $\{N_1, N_2, ..., N_N\}$ and connectivities (k), or degree, which exhibits a power law distribution $P(k) \sim k^{\gamma}$ (Aldana, 2003), (Barabási and Bonabeau, 2003), (Clauset et al., 2009), (Cohen et al., 2003), (Serra et al., 2003) (see Figure. 3.4). $P(k)$ is the probability distribution that an arbitrary node of the network is connected to n other nodes, and γ is the scale free exponent, or scaling parameter. Scale free exponents often lie in the range $2 < \gamma < 3$; however, there are some exceptions.

SFBNs can be constructed by adding nodes incrementally to an existing network (*growth mechanism*) and by creating new connections to existing nodes with a *preferential attachment mechanism* i.e. new nodes will prefer to connect to more connected nodes. The probability p that a new node will be connected to a given node N_i depends on the number of existing connections, k_i, that node N_i has. The mathematical expression of this probability is: $p \sim \frac{k_i}{\sum_d k_d}$, where k_i is the connectivity

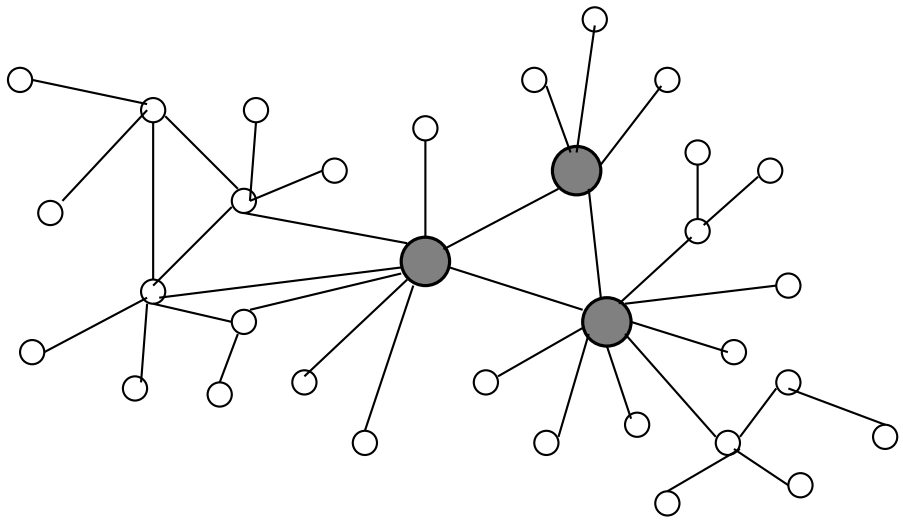


Figure 3.4: A scale free Boolean network, showing three hubs in grey.

of node N_i and d is the index denoting the sum over all network nodes. These two mechanisms explain the existence of *hubs*, which are nodes having connections with many other nodes in the network.

Each node, N_i, has a binary state, either 0 or 1, and is connected to k_i other nodes of the network $\{N_{i1}, N_{i2}, ..., N_{ik_i}\}$, where k_i is randomly chosen from a probability distribution $p_{inp}(k)$. $p_{inp}(k)$ is defined as follow: $p_{inp}(k) = [(\sum_{n=1}^{\infty} k^{-\gamma})n^{\gamma}]^{-1}, \gamma > 1$. At each time step a Boolean function $F_i(N_{i1}, N_{i2}, ..., N_{ik_i})$ taken from a set of Boolean functions $\oint\{F_1, F_2, ..., F_N\}$ is assigned to N_i, such that for each state of k_i other nodes, $F_i = 1$ with probability p and $F_i = 0$ with probability $1 - p$. Each node of the network is updated synchronously as follows: $N_i(t+1) = F_i(N_{i1}(t), N_{i2}(t), ..., N_{ik_i}(t))$ and the entire network $\chi(t)$ is updated at time t with this dynamical equation: $\chi(t+1) = \oint(\chi(t))$, where $\chi(t) = \{N_1(t), N_2(t), ..., N_{(t)}\}$. SFBNs are more robust to external perturbations than Boolean networks (Aldana, 2003). Aldana (2003) showed that for most real scale free networks $\gamma \in [2, 2.5]$.

3.2 Probabilistic Boolean Networks

A probabilistic Boolean network (PBN) is a variant (a stochastic version) of BNs (see figure 3.6). It is a stochastic model, which has more than one possible Boolean function for every node (Shmulevich et al., 2002). It is a dynamical system represented as a directed graph composed by a set of N nodes, $N = \{g_1, g_2, \cdots, g_n\}$, a connectivity

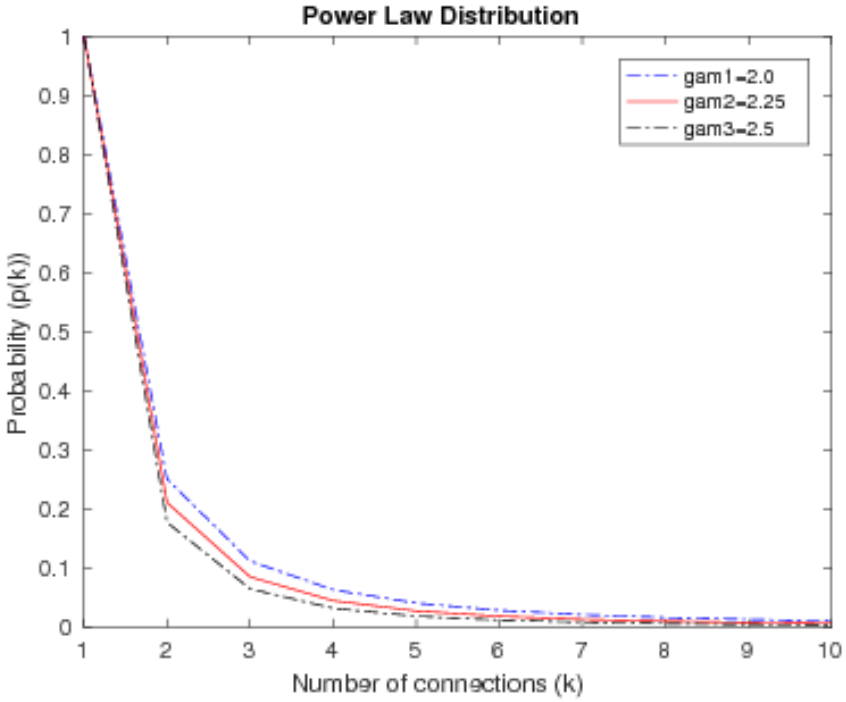


Figure 3.5: Power Law Distribution ($k = [1 - 10]$, $\gamma_1 = 2.0$ $\gamma_2 = 2.25$, $\gamma_1 = 2.5$). The plot shows the distribution of connectivities in scale free networks for three different values of γ. A large number of nodes have only a few connections, and a small number of nodes (hubs) have a large number of connections.

k (or number of inputs), and a set of $\oint_i$ Boolean functions (or predictor functions) $\oint_i = \{\varphi_j^{(i)}\} = \{\varphi_1^{(i)}, \varphi_2^{(i)}, \cdots, \varphi_{h(i)}^{(i)}\}$; where n is the number of genes, $j = 1, 2, \cdots, h(i)$, $h(i)$ is the number of possible Boolean functions corresponding to every node g_i and $\varphi : \{0,1\}^n \to \{0,1\}$. Each node g_i is assigned a set of Boolean functions $\oint_i$. The probability of selecting $\varphi_j^{(i)}$ as a Boolean function is $d_j^{(i)}$, $0 \leq d_j^{(i)} \leq 1$, $\sum_{j=1}^{h(i)} d_j^{(i)} = 1$ with $i = 1, 2, \cdots, n$. The realisation of PBN at a given time t is determined by a vector of Boolean functions. The number of vector Boolean functions is equal to the number of possible realisations.

Let γ_j be a set of jth possible realisation and M the number of possible realisations. $\gamma_j = (\gamma_{j(1)}^{(1)}, \gamma_{j(2)}^{(2)}, \cdots, \gamma_{j(n)}^{(n)})$, $1 \leq r \leq M$, $1 \leq r(i) \leq h(i)$ and where $\gamma_{j(i)}^{(i)} \in \oint_i$, $\gamma_j : \{0,1\}^n \to \{0,1\}^n$, $i = 1, 2, \cdots, n$. Each possible realization maps at least one of the φ_j Boolean functions. If the selection of the Boolean function for each node is independent the PBN is said to be an independent PBN, the probability of choosing jth BN $\mathcal{P}_j$ is given by $\mathcal{P}_j = \prod_{i=1}^{n} d_{j_i^{(i)}}$, $i = 1, 2, \cdots, n$. There are at most $M = \prod_{i=1}^{n} h(i)$ different possible realizations of BNs. If $h(i) = 1$, for all $i = 1, 2, \cdots, n$, then $M = 1$ and the PBN is reduced to a classical Boolean network.

The given probability values used in this work are $p_1 = 0.3$ and $p_2 = 0.7$. These values were chosen in order to observe the effect of the control method on PBN dynamics when a small and a large number of nodes are updated at a given time step.

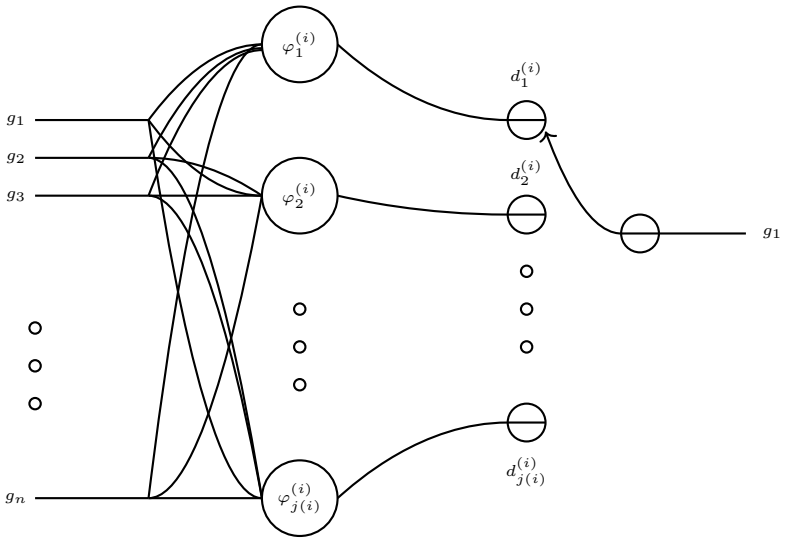


Figure 3.6: A building block of a probabilistic Boolean network adapted from (Shmulevich et al., 2002). On this figure the *wiring* of the inputs to each Boolean function appear to be general, however, in reality, each Boolean function has only a few input variables.

3.3 Related Literature

3.3.1 Boolean Modelling of Biological Networks

Modelling biological processes using quantitative and continuous mathematical models such as differential equations has brought important insights to systems biology (Le Novère, 2015), (Akutsu). However, these models are often inefficient when simulating larger biological networks. This has promoted interest in discrete-valued models such as BNs. The use of binary states and Boolean functions makes BNs especially cheap to simulate on a computer. Numerous studies have demonstrated that, despite their apparent simplicity and high level of biological abstraction, these models are often able to capture the qualitative dynamics of biological processes. For example, Kauffman et al. (2003) developed a BN model of the yeast transcription network, Albert and Othmer (2003) used BNs to successfully model the GRN

underlying pattern formation in drosophila, Dallidis and Karafyllidis (2014) have modelled the quorum sensing circuits of *Pseudomonas aeruginosa* and Kaushik and Sahi (2015) developed a BN model of the GPR142 biological pathway in type 2 diabetes.

A number of studies have applied Boolean models to cancer analysis, both by considering specific pathways (Saadatpour and Albert, 2013), (Davidich and Bornholdt, 2008), (Fumiã and Martins, 2013) and through more abstract systems-level studies (Huang et al., 2005), (Huang et al., 2009). Many of these studies have carried out an attractor analysis of the resulting models in order to gain insights into the biological system's stable states (Albert and Othmer, 2003), (Huang et al., 2005), (Davidich and Bornholdt, 2008), typically associating these with phenotypes. In (Poret and Boissel, 2014), the authors went a step further and identified nodes whose state would effect the accessible attractors; this can help in identifying potential drug targets for preventing the expression of pathological phenotypes. Discrete models such as BNs have been shown to be equivalent to continuous models when only the steady states of the system are considered (Veliz-Cuba et al., 2012); however, it should be borne in mind that BNs are not appropriate when a detailed quantitative understanding of a process is required. For a review of Boolean modelling in biology, see (Saadatpour and Albert, 2013).

3.3.2 Controlling Boolean Networks

The development of control interventions for complex dynamical systems is an important topic, with potential applications in diverse areas such as sociology, economics, drug discovery and treatment of diseases of the immune system (Kitano, 2002), (Kitano, 2004). Many results have been obtained for the control of linear systems, while for nonlinear systems control a fewer number of practical results have been obtained (Azuma and Imura, 2006). This indicates that useful and efficient control strategies need to be developed for complex biological systems such as GRNs. A significant focus of the recent research in the field of control systems is the development of executable computational models whose behaviours can be

controlled and analysed in order to understand the complex dynamics of their real world analogues, which are generally non-linear systems. Several mathematical and computational models have been designed for modelling complex and non-linear biological systems.

BNs have been used to model many real biological networks, for example the regulatory networks in the mammalian cortical area (Giacomantonio and Goodhill, 2010), the mammalian cell cycle (Fauré et al., 2006), and T-cell large granular lymphocyte leukemia (Saadatpour et al., 2011), (Chaos et al., 2006), (Mendoza and Xenarios, 2006), (Klamt et al., 2006), (Davidich and Bornholdt, 2008), (Li et al., 2004). It is clear that BNs cannot model all details of biological networks because of their simplicity, however they do give a good and reliable approximations of the non-linear biological functions (Amaral et al., 2004). Also, BNs have been used to explain how perturbations can affect biological system's natural behaviours and their consequences (Albert and Thakar, 2014).

Finding strategies to control BNs is therefore an important and challenging problem. The control problem is typically defined in terms of leading a BN's trajectory towards a particular point in its state space, ideally by manipulating the state of a minimum group of nodes and with the aim of reaching the target state in a minimal period of time. Like with the complex networks that they model, BNs have a number of characteristics that make them hard to control, including non-linear dissipative dynamics, multiple stable states and high dimensionality (Motter, 2015). A number of previous works on the control of BNs have been conducted (Akutsu et al., 2007), (Cheng and Qi, 2009), (Shi-Jian and Yi-Guang, 2011), (Kobayashi and Hiraishi, 2012a), (Kim et al., 2013), (Motter, 2015), (Gates and Rocha, 2016), (Veliz-Cuba et al., 2014), (Zhang et al., 2007), (Garg et al., 2007), (Albert and Barabási, 2000), (Drossel et al., 2005), (Kauffman, 1993), (Samuelsson and Troein, 2003), (Aldana, 2003), (Amaral et al., 2004), (Abul et al., 2004), (Kobayashi and Hiraishi, 2011), (Kobayashi and Hiraishi, 2012b), (Kyozuka et al., 1997).

Many of these use control theoretic approaches. For instance, pinning control methods have been used to stabilise the dynamics of BNs, allowing particular

phenotypic states to be maintained (Li, 2016). In (Cheng and Qi, 2009), (Cheng et al., 2010), (Cheng and Zhao, 2011) Cheng et al. suggested the semi-tensor product of matrices approach for controlling BNs. Fornasini and Valcher (2014), based on the concept of infinite horizon, proposed a method for the optimal control of BNs. Laschov and Margaliot (2011) and Chen et al. (2015) considered Mayer-type optimal control techniques to control BNs. Also, in (Li and Sun, 2011), (Li and Sun, 2012), the authors used semi-tensor of matrices to control BNs with time-variant and time-invariant delays in states.

However, in general the control of BNs is known to be NP-complete (Akutsu et al., 2007), meaning that optimal control techniques can only be applied to networks of limited size, though polynomial-time algorithms have been developed for BNs with constrained topologies such as tree structures (Akutsu et al., 2007). In order to express the non-deterministic dynamics of BNs, Kobayashi and Hiraishi (2012a), suggested a novel mathematical method for controlling BNs using inputs, based on the Petri net framework. They have reduced the control problem of Petri nets to an integer programming problem. Based on the discrete-time dynamics Cheng and Qi (2009) were able to control BNs by using two types of inputs: free Boolean sequence and input Boolean network. They have also solved the problem of observability for free Boolean sequences by choosing necessary and sufficient conditions. In (Shi-Jian and Yi-Guang, 2011), the authors presented a method to control random BNs, which used periodic functions and the average sensitivity of Boolean functions of the nodes. This method periodically freezes a fraction of the network based on average sensitivity of Boolean functions of the nodes and the probability. Then numerical analysis was used to estimate the performance of the proposed control method and simulation was used to illustrate the effectiveness of this method.

To an extent, the control problem can be made easier by identifying nodes that have dominant roles within the network (such as hubs in scale-free networks) and focusing control interventions on these nodes (Liu et al., 2011), (Kim et al., 2013). This works well for certain kinds of networks, but in general it has been shown that dynamics can not be determined by structure alone, and therefore that methods

based on structural analysis will not always be effective (Gates and Rocha, 2016). In the same way Clark et al. (2017) attempted to solve the problem of computing a minimum-size subset of control nodes which can be used to force a given biological network towards a desired attractor in BNs. They found that the network topology and its nodes, threshold dynamics are central in solving the input selection problem. For example, in the case of a cactus topology, they solved the input selection problem in polynomial time. Networks with nested canalyzing dynamics could also be addressed using polynomial time algorithms. The control process of BNs can be affected by the size of the network. When the size of the network increases, the control problem becomes more difficult to solve. To tackle this issue, a number of studies were done using BNs with different sizes. Hou et al. (2016) used integer linear programming (ILP) to choose the minimum number of driver nodes to carry out theoretical analyses on the average size of the minimum set of driver nodes, both in six different BNs models of real gene regulatory networks.

Liu and Barabási (2016) were able to control a regulatory network model of the mammalian circadian rhythms in mice using the feedback vertex set. This network is made of 21 nodes (see (Mirsky et al., 2009) for more details). Gao et al. (2013) have used algebraic approach and matrix semi-tensor product theory to control two types of GRNs: the protein-nucleic acid interactions network with size 6 and the cAMP receptor of *Dictyostelium discoideum* network formed of 8 nodes. Using a modified configuration model, authors in (Darabos et al., 2007) attempted to control four SFBNs with connectivity $k = 4$ and size $[50, 100, 150, 200]$. In (Murrugarra et al., 2016), using an algebraic approach, the authors controlled an algebraic BN representation of the $p53 - mdm2$ network and blood T cell lymphocyte granular leukemia survival signalling network which have respectively 16 and 60 nodes. To perform the control task, they have used computational algebra techniques such as Gröbner basis to find controllers. Kim et al. (2013) achieved to control a certain number of computational models of biological networks, by finding a minimal set of nodes as the control kernel that can perform the control task. They have also developed a general algorithm for identifying this control kernel. Table 3.1 represents

a summary of previous work on controlling BNs.

Table 3.1: Summary of some selected work on controlling Boolean networks (BNs)

First author, year	Approach
Cheng and Qi (2009)	Semi-tensor product of matrices
Cheng et al. (2010)	Discret-time dynamics techniques
Cheng et al. (2010)	Semi-tensor product of matrices
Cheng and Zhao (2011)	Semi-tensor product of matrices
Laschov and Margaliot (2011)	Mayer-type techniques
Shi-Jian and Yi-Guang (2011)	Periodic function and average sensitivity of Boolean functions
Zhao et al. (2011)	Concept of infinite horizon
Li and Sun (2011)	Semi-tensor matrices
Li and Sun (2012)	Semi-tensor matrices
Kobayashi and Hiraishi (2012a)	Petri net framework
Kim et al. (2013)	minimum-size subset (control kernel)
Fornasini and Valcher (2014)	Concept of infinite horizon
Chen et al. (2015)	Mayer-type techniques
Li (2016)	Pinning control
Murrugarra et al. (2016)	Algebraic approach (Gröbner basis)
Clark et al. (2017)	Combinatorial algorithms

3.3.3 Evolving Boolean Networks

In addition to modelling biological GRNs, a number of studies have shown that GRN models can be used to carry out complex computational and control behaviours that are to some degree analogous to their biological activities (Lones, 2016). Typically this is done by optimising the model using an evolutionary algorithm, and includes a number of approaches that have used BNs (Dubrova et al., 2008), (Roli et al.),

(Goudarzi et al., 2012), (Zanin and Pisarchik, 2011). For example, Bull and Preen evolved BNs to solve digital design problems such as multiplexing in synchronous and asynchronous systems (Bull and Preen, 2009).

Another notable work is that of Roli et al., who evolved BNs to control robotic behaviours. Mesot and Teuscher (2005) demonstrated that BNs can achieve better performance than CAs on tasks that measure the capacity of distributed models to perform global density classification. In (Goudarzi et al., 2012), the authors showed that BNs can be used to solve information processing problems, showing that network learning and generalisation can be optimised according to the complexity of the task and the quantity of information provided.

3.3.4 Implementing Boolean Networks in Cells

Part of the justification for using BNs in this study is the potential for implementing them as optimised control systems within biological cells. One benefit of BNs, in this respect, is that they are relatively amenable for implementation in biological cells using existing synthetic biology approaches. A key focus of synthetic biology has been on implementing digital circuits within cells, the idea being that this will allow more conventional computing approaches to be readily refined into biological systems. However, these approaches also have direct relevance to BNs, since both digital circuits and BNs are comprised of Boolean logic functions that can be implemented as logic gates. Synthetic biology has demonstrated that logic gates can be implemented in various biochemical forms, including proteins, RNA and DNA (Purcell and Lu, 2014), (Singh, 2014), (Shi et al., 2014). It is also possible to assemble these logic gates into circuits, though it remains challenging to implement large circuits due to crosstalk between logic gates (Purcell and Lu, 2014). Other authors have considered the potential for using synthetic biology to implement control systems: in (Cury and Baldissera, 2013), for example, the authors discuss how conventional control approaches may be refined into biological forms and used to control a cell's GRN.

Another benefit of using BNs is that, because they are relatively abstract, they are less likely to be susceptible to the "reality gap" that is often found in computational

modelling. This gap occurs when a model is optimised under simulation, but then does not function correctly when used in a real world setting. This is caused either by over-fitting to the simulation, or by noise in the real world system. Since BNs have few parameters, they are less likely to over-fit than continuous-valued models of GRNs. Since relatively large signal differences are required to cause binary state changes, they are also likely to be less affected by noise. In this respect, the value of a Boolean approach has previously been demonstrated in the field of robotics, where Boolean network controllers were found not to be susceptible to the reality gap (Roli et al.).

3.3.5 Using Artificial Gene Regulatory Networks as Controllers

There have been a number of studies which used artificial gene regulatory networks to control other systems such as robots (Cussat-Blanc and Pollack, 2012), (Bentley, 2003), (Zahadat et al., 2010), (Cussat-Blanc et al., 2012), (Bentley, 2004), (Lones et al., 2013b). An early example of this is the work of Quick et al. (2003), who used AGRNs controllers to control robots behaviours, by continually coupling the controllers to the environment in which the robots are embodied. Bentley (2004), used fractal gene regulatory networks to train a robot to avoid a series of obstacles in its environment. Taylor (2004) developed AGRN controllers for the control of a group of underwater robots. These AGRN controllers were evolved using genetic algorithms and the results showed that they were able to successfully carry out control in a simple clustering task.

Kumar (2005) used a model of GRNs in combination with a spatially distributed evolutionary algorithm to evolve simulated robot controllers for solving obstacle avoidance problems. Using the idea of optimising fractal GRNs, Zahadat et al. (2010) were capable of controlling modular robots in a distributed way. Results from their proposed control method are better than the results from previous learning methods Trefzer et al. (2010) explored in two different case studies; simulated and real robots, the ability of an AGRN controller to control the robot behaviours in order to avoid

obstacles in its environment.

Another notable use of artificial gene regulatory networks is the work of Cussat-Blanc and Pollack (2012), who designed a developmental model in which prototype robotic blocks are controlled by an AGRN. By evolving in parallel the AGRN and a hormonal system, they were able to produce different virtual robots with target properties, for instance symmetry and regularity. Cussat-Blanc et al. (2012) designed a AGRN-based controller for robots present in video games, showing that GRN-based controllers can be optimised to teach a robot how to handle and manage simultaneously four conflicting and cooperative continuous actions. AGRNs are used to control robot swarm behaviours in a dynamic environment in Yao et al. (2014).

AGRNs are not only used in robotics. For instance, Turner et al. (2013) have used AGRNs inspired by biological epigenetics to control a system of coupled inverted pendulums. In (Lones et al., 2013b), the authors used artificial biochemical networks (ABNs), which include AGRNs, to diagnose Parkinson's disease, using the ABNs to perform classification of time series data. AGRN characteristics such as self-organization and cell differentiation, made them also suitable to designing digital circuits in (Zhan et al., 2009). Table 3.2 summarises the previous work described in this section (Section 3.3.5).

3.4 Summary

This chapter presents some of the many different models which have been developed by taking inspiration from gene regulatory processes. The main objective of designing these kind of models is to attempt to model biological gene regulatory networks and also capture the emergent properties of biological gene regulation. There has been a large amount of research conducted in modelling gene regulation and previous work on capturing emergent biological properties has shown interesting and promising results. All models described in this chapter capture important emergent dynamics which are not clearly coded within the models. This means that these models capture the principle of emergence. The models detailed in this chapter can be used to do both biological modelling and carry out computational tasks, such as control.

In addition, this chapter reports previous work done on controlling, evolving and implementing Boolean networks, the computational model of gene regulatory network used in the following chapters.

The following chapter, describes the methods for artificially evolving the computational networks.

Table 3.2: Summary of some selected work on controlling dynamical systems using artificial gene regulatory networks (GRNs)

First author, year	Approach	Problem
Quick et al. (2003)	Artificial GRN, continuous valued	Simulated robot control and temperature regulation
Bentley (2004)	Fractal GRN	Control of simulated hexopod robot behaviours to avoid obstacles
Taylor (2004)	Artificial GRN	Control of underwater robotic swarm
Bentley (2005)	Fractal GRN	Software fault-tolerance
Kumar (2005)	Artificial GRN, continuous valued	Control of simulated and real robot behaviour to avoid obstacle
Zhan et al. (2009)	Artificial GRN, discrete valued	Electronic circuit design
Joachimczak and Wróbel (2010)	Artificial GRN, operons	Control of real time foraging behaviour
Krohn and Gorse (2010)	Fractal GRN	Control of single and joint inverted pendulum
Nicolau et al. (2010)	Artificial GRN, based on Banzhaf (2003) with small changes	Control of an inverted pendulum
Trefzer et al. (2010)	Artificial GRN, discrete valued	Control of simulated and real robot behaviour to avoid obstacle
Zahadat et al. (2010)	Fractal GRN	Control of a modular robot
Cussat-Blanc et al. (2012)	Artificial GRN, based on Nicolau et al. (2010)	Control of intelligent agents in video games
Cussat-Blanc et al. (2012)	Artificial GRN	Control of coupled inverted pendulums
Lones et al. (2013b)	Artificial GRN and artificial metabolic networks	Diagnosis of Parkinson's disease
Yao et al. (2014)	Artificial GRN	Control of robot swarm in dynamic environment

Chapter 4

Evolutionary Algorithms

This chapter is focused on evolutionary algorithms (EAs) (Eiben and Smith, 2008), (Bäck, 1996), (Bäck et al., 1997) a form of evolutionary computation that mimic the natural process of Darwinian evolution in order to solve complex non-linear computational problems.

In an EA, a population of candidate solutions is generated and iteratively evolved to search over the solution space of a problem (see figure 4.2). Each time a new solution is generated, it is evaluated using a fitness function that measures its objective value (or fitness). This value is then used to select between solutions in the population, using the solutions with the best fitnesses to generate new solutions, whilst removing solutions with poor fitness from the population. New solutions are generated using mutation and recombination operators; mutation operators make small changes to existing solutions, recombination operators join parts of existing solutions to make new solutions. EAs are global optimisers (meaning they are relatively insensitive to local optima) and have much in common with other global optimisers, such as particle swarm optimisation (Lones, 2014), (Poli et al., 2007).

There are various types of EAs. This chapter describes some of the most used and well known evolutionary algorithms, underlines their biological inspiration and their evolvability. EAs are a core aspect of the work in this thesis, as they will be the main tools used to evolve artificial gene regulatory networks models and are therefore essential to capture complex emerging properties.

4.1 Genetic Algorithms

Genetic algorithms (GAs) are one of the most widely used EAs. They were introduced by Holland (1975) as a form of evolutionary computation. Originality GAs were suggested to assess and observe evolvability and emergence, then later became an optimisation tool. At first GAs were used to optimise binary strings, but since then have been used on other data structures. Often, GAs have a distinct genotype and phenotype. The genotype represents the data structure which will be evolved and the phenotype is its computational behaviour. Each individual is evaluated with respect to its phenotype, and genetic operators such as crossover, mutation and selection are applied to its genotype.

In general, GAs use crossover, mutation and selection operators as genetic operators. Crossover, also known as the recombination operator, is a computational model of vertical gene transfer. There are some GAs which use crossover based on horizontal gene transfer (Harvey, 2009). Two types of crossover are commonly used in GAs, $n-$point crossover and uniform crossover. With $n-$point crossover children are created using specified portions from each parent and, in uniform crossover, crossover points are generated with a certain probability of passing information on to a child (see figure 4.1). In this thesis, uniform crossover is used.

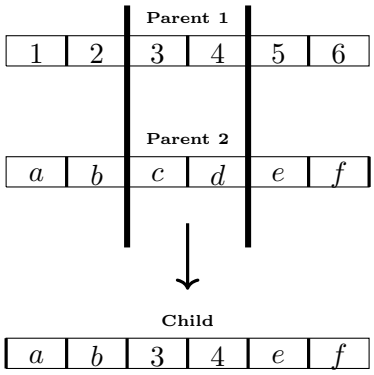


Figure 4.1: An example of n-point crossover, where $n = 2$ showing how children are generated using crossover.

Usually, GAs use three selection strategies: fitness proportional, ranking selection and tournament selection. Rank selection scores all individuals according to their fitness in the population, removing the absolute difference between each individual's

fitness. This means that the difference between many extremely close fitness scores can be multiple ranks and an individual with a fitness higher than the rest of the population fitness will always be one rank higher than the next best solution. This can potentially be a disadvantage for this solution (Bäck et al., 2000). Fitness proportional selection, also called roulette wheel selection is used in GAs to select the best solutions, by mapping the fitness of each individual using a scaling function. The probability p_i for an individual i to be selected is defined as $p_i = \frac{f_i}{\sum_{j=1}^{\eta} f_j}$, where f_i is the fitness of individual i and η the population size. Individuals with high fitness are more likely to be selected (Goldberg and Deb, 1991). Tournament selection selects randomly a number of individuals which compete in a tournament and the winner is chosen to be a parent. The evolutionary pressure can be modified by changing the size of the tournament. When the size of tournaments is large, it is very hard for weaker individuals to be selected since it increases the selection pressure; whereby for small size of tournaments make it easier by decreasing the selection pressure (Goldberg and Deb, 1991).

Mutation is used in GAs to promote genetic diversity from one generation of the population to the next. It depends on the representation of the genomes and its corresponding data structures (genotype). A genome can be represented by either binary numbers or real numbers. When the genome uses binary numbers, the mutation operator will flip each of the bits with a certain probability based on the mutation rate. For a genome using real numbers, a new number can be chosen either randomly among a set range, or from a distribution centred around the original number. Mutation is commonly applied to all individuals, except when the selection strategy uses elitism, in this case elite individuals will be copied verbatim to the next generation (Bäck et al., 2000).

A GA's execution starts with a randomly generated initial population of size η; this step is called initialisation. Then a fitness is assigned to individuals within the population. Fitness is a term used to measure how well a task has been achieved. Then, the parents are chosen based on a selection strategy, and the children are generated using crossover. These steps are repeated until there is a new population

of size η. The new population is subjected to the mutation operator, and after, the new population becomes the current population and all the steps after initialisation are repeated for a certain number of generations, or until the stopping criteria are met; see figure 4.2.

GAs have various important characteristics which enable them to evolve GRNs. For instance, a GA does not have any requirements about the individuals which will be evolved, all it needs is the data structures of the individuals and the fitness evaluation method. Also crossover and mutation used to create individuals at each generation are comparable to real life biology.

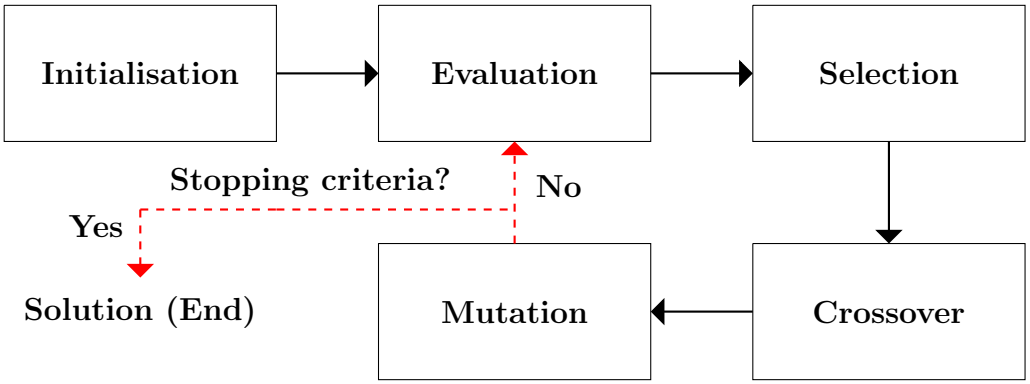


Figure 4.2: Evolutionary algorithm general framework

4.2 Genetic Programming

Genetic programming (GP) is an evolutionary algorithm used to create a program (Akutsu et al., 2007). In standard GP, a program is represented as a tree structure (see figure 4.3). In general, GP requires a set of symbols, known as the terminal and non-terminal sets, which are used to create the tree, but also a fitness function to find the fitness of the program. An initial population is randomly generated by putting together elements of the terminal and non-terminal sets into tree structures. GP genetic operators are similar in function to GA ones. Crossover (or recombination) is done by exchanging sub-trees between parents to generate two children (see figure 4.3). Mutation replaces a sub-tree with a randomly created sub-tree.

There are different types of GP such as linear genetic programming (LGP) and Cartesian genetic programming (CGP) which do not represent their programs

using tree structures. Cartesian GP was proposed by Miller and Thomson (2000) to represent electronic circuits; but later it has been used to represent general programmable structures. It represents a program as a graph structure encoded as a set of integers. Brameier and Banzhaf (2007) introduced linear GP, which uses a list of instructions to describe a program.

GP has been successfully used to solve a large number of problems including controlling robotic behaviours (Lazarus and Hu, 2001). However, GP has some weaknesses in terms of evolvability. For instance, sub-tree crossover often does not perform effective recombination (Lones, 2004). This is due to the loss of content when sub-tree are exchanged between programs. It is also due to the inherit lack of evolvability in program encodings developed by humans, which do not react well to the application of material evolving computation, a trend which has grown in recent years within the GP community (Lones, 2016).

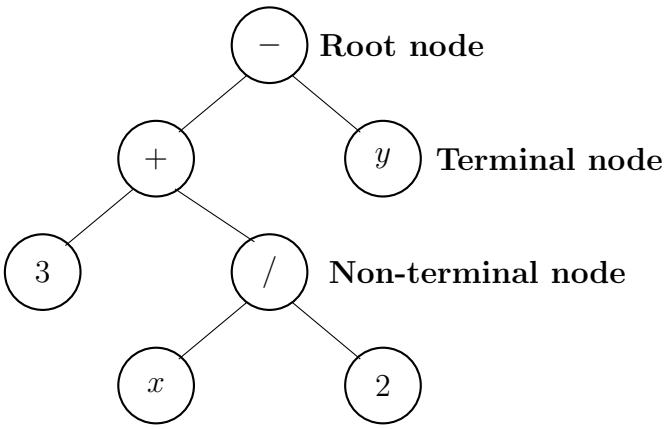
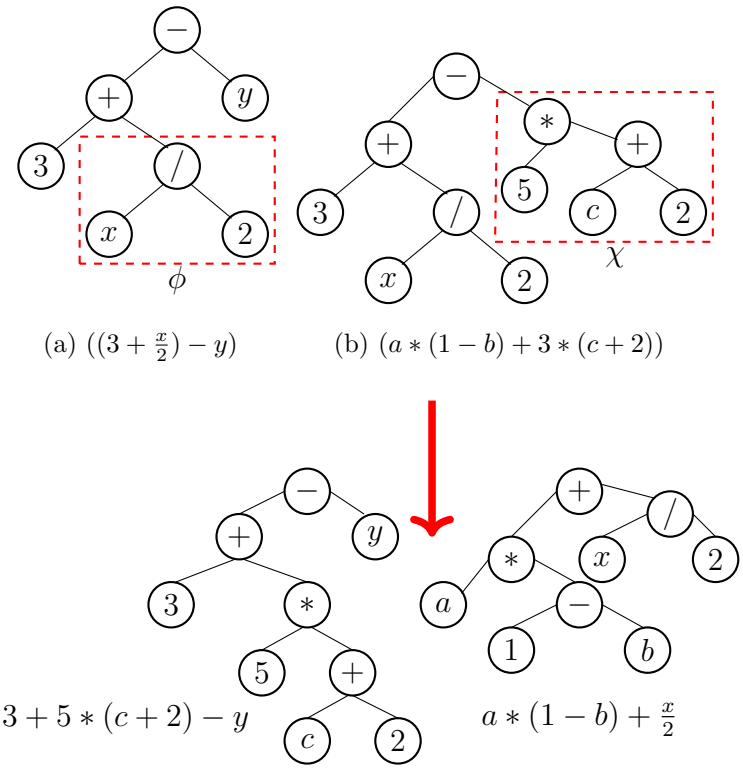


Figure 4.3: An example of a tree in genetic programming. This tree represents the equation $(3 + \frac{x}{2}) - y$. A set of inputs can be provided and iterated several time through the program to find the fitness of this program.



(a) $((3 + \frac{x}{2}) - y)$ (b) $(a * (1 - b) + 3 * (c + 2))$

(c) Children obtained after swapping ϕ and χ.

Figure 4.4: An example of the crossover operation with genetic programming trees. Two sub-trees are selected and swapped between the parents to create the children.

4.3 Evolutionary Strategies

Evolutionary strategies (ESs) was introduced by Ingo Rechenberg, Hans-Paul Schwefel and Peter Bienert (Bäck et al., 2000). At the beginning, ESs had a single parent and performed mutation to generate a child. If the child has a better fitness than its parent then it becomes the parent and the old parent is removed. This process is similar to a hill climbing algorithm. Since their introduction in 1960, many ESs have been developed (Beyer and Schwefel, 2002). The first selection strategy generates more than one child for a given parent, and all the worst individuals are destroyed in other to keep the population constant. The second selection strategy discards parents with respect of their fitness. In addition, to these two selection strategies, a crossover operator was added in many models adding the ability to generate children with more than one parent (Bäck et al., 2000).

4.4 Evolutionary Programming

Evolutionary programming (EP) is a population based algorithm for optimisation which takes inspiration from a restricted view of evolution (Fogel et al., 1964). Individuals in the population are finite state machines with fixed structures. In EP, a recombination operator is not used, because each individual is treated as a key element whose structure is not divided into sub-units. The fitness for an individual is measured by putting this individual in an environment and giving a group of symbols as input, the output is compared to the next input symbol. Through this process error can occur and is accumulative over all input symbols. The best individuals are chosen and are mutated to generate children, once all the population has been assigned a fitness value. Then the best parents and children are selected to become the next generation.

The absence of a crossover operator does not allow individuals to take elements and behaviours from other individuals of the population. Moreover, EP requires individuals to be finite state machines which can be very restrictive for phenotypes. It have been criticised for its slow convergence time. However, since its first use there have been various improvements especially in mutation strategies which have been proved by Yao and Liu (1997) to increase the convergence.

4.5 Multi-Objective Evolutionary Algorithms

Most real-world optimisation problems have multiple objectives, usually in conflict with each other, making these problems hard to solve. Often multi-objective optimisation problems are considered as single objective problems and the rest of the objectives are taken as constraints (Deb, 2001). Multi-objective evolutionary algorithms (MOEAs) were developed to be suitable tools for solving multi-objective problems, since EAs have particular properties that are useful for this kind of optimisation. MOEAs handle mutli-objective optimisation tasks, by trading-off between these conflicting objectives. Over the past decades, several MOEAs were designed such as Non-Dominated Sorting Genetic Algorithms II (NSGA II) (Deb et al., 2000),

(Deb et al., 2002), (Coello et al., 2007), Multi Objective Genetic Algorithms (MOGA) (Murata and Ishibuchi, 1995), Vector Evaluated Genetic Algorithms (VEGA) (Schaffer, 1985), Pareto Archived Evolution Strategy (PAES) (Knowles and Corne, 1999) and Strength Pareto Evolutionary Approach algorithms (SPEA, SPEA-2) (Zitzler et al., 2001), (Zitzler and Thiele, 1999).

4.5.1 Multi-Objective Optimisation

Real-world optimisation problems usually involve multiple conflicting objectives and highly complex search spaces, which prevents simultaneous optimisation of each objective. To address this, multi-objective optimisation techniques are needed. Evolutionary algorithms have various characteristics that make them useful for exploring multiple solutions at once, and consequently multi-objective evolutionary algorithms (MOEAs) are often used for such problems. Multi-objective optimisation problems (MOOPs) have solutions which explore trade-offs in different ways. These are called Pareto optimal solutions (or non-dominated solutions), where none of the objectives in the search space can be improved without decreasing in value one or more other objectives. In general MOOP comprises a set of n parameters known as decision variables, a set of b objective functions, and finally a set of m constraint functions. The set of feasible decision vectors is defined by the objective and constraint functions. MOOPs can be formulated in mathematical terms as follows:

$$\begin{aligned} maximise \quad & h = f(z) = (f_1(z), f_2(z), \cdots, f_b(z)) \\ subject \quad to \quad & c(z) = (c_1(z), c_2(z), \cdots, c_m(z)) \leq 0 \\ where \quad & z = (z_1, z_2, \cdots, z_n) \in Z \\ & h = (h_1, h_2, \cdots, h_b) \in H \end{aligned}$$

z is the decision vector, h is the objective vector, Z and H are called respectively the decision space and the objective space. The main objective of a multi-objective

optimisation algorithm is to find solutions in the Pareto optimal set and this requires to investigate solutions at the extreme ends of the objective function space (**?**). In this thesis NSGA II was used and is described in the following section.

4.5.2 Non-Dominated Sorting Genetic Algorithms II

Non-Dominated Sorting Genetic Algorithms II (NSGA II) is one of the most known multi-objective genetic algorithms (MOGA) (Deb et al., 2000), (Deb et al., 2002), (Coello et al., 2007). NSGA II sorts and ranks each individual of the population according to dominance criteria. An individual is considered to dominate others if it is better in at least one objective, and not worse in the remaining objectives. Every individual of the population achieving this performance will become part of the first non-dominated front, also know as the Pareto front, see figure 4.5. This process is repeated and the previous non-dominated front is excluded, to generate a second non-dominated front and so on.

Another important operator used in NSGA II in addition to non-domination rank is the crowding distance. Crowding distance is a measure of density of individuals (which means how close an individual is to its neighbours) within a non-dominated front. Crowding distance is created to provide a uniform distribution of individuals through a non-dominated front. Each individual in the population will have a non-domination rank and crowding distance. The process, known as partial order in NSGA II, is when an individual a is greater than individual b if it has a better or equal rank, or has a better crowding distance (Deb et al., 2000), (Deb et al., 2002), (Coello et al., 2007) see figure 4.5. NSGA-II gives a more realistic view of evolution, as individuals are better in achieving some tasks than others and NSGA II describes this well. Nevertheless, from a real world biological point of view, it uses a forced elitism, assuring that the fittest individuals are copied onward to the next generations. By doing so, NSGA-II decreases diversity of the population.

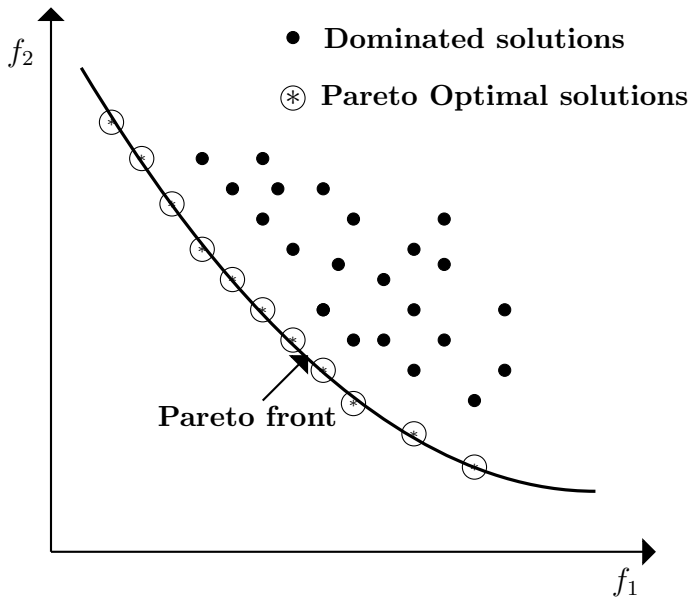


Figure 4.5: An illustration of non-dominated Pareto front and non-dominated solutions, where objective f_1 is minimised and f_2 is maximised.

Evolutionary algorithms	Representation	Genetic operators	Elitism selection	Determinism selection
Genetic Algorithms	Binary values, real values	Crossover, mutation	Optional	No
Genetic Programming	Real values, integers, (tree structures)	Crossover, mutation	Optional	No
Evolutionary Strategies	real values, strategy parameters	Crossover, mutation	Yes	No
Evolutionary Programming	Real values, finite state machine	Crossover, mutation	Yes	Yes

Table 4.1: A summary of the evolutionary algorithms described in this chapter. This table is adapted from (Knowles and Corne, 2000).

4.6 Summary

This chapter presents four main categories of evolutionary algorithms, which are each inspired by biological evolution. Table 3.1 shows a summary of these algorithms. Evolutionary strategies have no requirement in terms of phenotypes, and usually do not take a large perspective of population dynamical behaviours. Often they use a single individual to do several clones within a population. Classical genetic programming and evolutionary programming algorithms require fixed representations, respectively tree structures and finite state machines. Genetic algorithms do not require a fixed representation for genotypes (i.e. data) and phenotypes (i.e. function). Also, they have a more biologically-faithful population-based architecture.

Chapter 5

Controlling Boolean Networks using Boolean Networks

Complex networks, such as societies, economies and ecosystems, are omnipresent in the real world and have important effects upon people's lives. Therefore, the control of their complex dynamical processes is a growing interesting and challenging problem for scientists (Gates and Rocha, 2016). The difficulties faced in controlling these complex networks are due to a large number of properties that make them particularly hard to control (Motter, 2015). The complex networks chosen to be studied and controlled in this thesis are GRNs. They are biochemical networks that involve genes and their protein products, especially the transcription factors that allow a gene to regulate another gene's expression. GRNs are fundamental to the behaviour of biological organisms, and control both the internal functions of individual biological cells and the overall development of multicellular organisms. In recent years, there has been a concerted effort to characterise and map GRNs of various organisms. However, there has been relatively little work and advancement in the control of GRNs.

The work within this chapter describes how BNs (Kauffman, 1969) can be evolved using EAs to control computational models of GRNs. In particular, EAs are used to discover BNs that can control the dynamics of other BNs. A number of parameters and conditions have been considered: the ability of EAs to optimise BNs, the general

ability of BNs to control other BNs, and the effect that topology has on both the difficulty of the control problem and the ability of the controllers. In addition, multiobjective evolutionary algorithms (MOEAs) are used to explore trade-offs between the effectiveness of control and its ease of realisation, focusing on minimising the number of interventions required to apply control. MOEAs have also been used to observe the trade-offs between four parameters: control efficacy, the number of interventions, the number of controller and controlled time steps. Several kinds of network are considered: deterministic and non-deterministic random Boolean networks and scale free Boolean networks (SFBNs). The chapter is divided into two main sections: the first describes the methods used to run experiments and the second presents results.

5.1 Evolving Controllers

The control method developed in this thesis is used to manipulate trajectories around the state space in order to control the dynamics of a given system. The approach does not require explicit knowledge of the underlying dynamics of the systems, that are controlled. It has been shown in previous work (Lones et al., 2010), (Lones et al., 2014) that artificial biochemical networks, for instance artificial gene regulatory networks, have the ability to control the dynamics of a system without having concrete information about the structure of the state space. The following sections describe the use of different variants of Boolean networks to control trajectories in other randomly generated Boolean networks about which little information is known.

5.1.1 Design of Experiments

In this work, different types of artificial gene regulatory networks are evolved using techniques of artificial evolution to perform control tasks. These evolutionary methods need to be flexible in order to manage various representations of executable structures. Therefore, genetic algorithms will be used when networks are evolved. This choice was made because genetic algorithms and implementations such as NSGA-II for multi-objective problems have been successfully applied to a wide range of computational

problems (Sivanandam and Deepa, 2007), (Mitchell, 1998). In addition, they have been used in recent years to evolve artificial biochemical networks (Lones et al., 2014), (Lones et al., 2010).

Using artificial biochemical networks to control complex systems is a challenging problem for them to solve, however, previous work, (Lones et al., 2010), (Lones et al., 2013a), (Lones et al., 2014) demonstrated that it is achievable. In this work, Boolean networks, a type of artificial gene regulatory networks are applied to control a range of other Boolean networks. Many BNs have ordered dynamics and some such as scale free Boolean networks and RBN with $k > 3$ tend to be chaotic. The fact that most of the Boolean networks do not express chaotic dynamics does not make them necessarily easier to control than chaotic Boolean networks.

Algorithm 1 explains the process of applying a controller network to a given control task. This technique uses a closed loop controller and can be applied when tasks are dynamical and are updated in discrete time (meaning that at each discrete time step the task will update as the same time step as the controller network).

Algorithm 1 Execution of the network during a control task

1: Initialise control task
2: **for** a predefined number of iterations **do**
3: Map task variables onto input genes ▷ interventions
4: Execute network
5: Map network output back to the task ▷ feedback
6: Update the task
7: **end for**

5.1.2 Genetic Algorithms and Parameters

Throughout this thesis, two types of genetic algorithms will be used to evolve BNs: the classical genetic algorithm, and the multi-objective genetic algorithm NSGA II. This work does not aim to develop improved optimisation algorithms, and so both algorithms use standard formulations that have been widely studied and applied (Bäck et al., 2000), (Goldberg and Deb, 1991), (Deb et al., 2002), (Coello et al.,

2007). Both algorithms have similar general characteristics and depending on the kind of task, either single or multi-objective, the corresponding algorithm is chosen. In NSGA-II, a selection strategy, which is rank based, will be used, rather than tournament selection and elitism. The genetic operators which will be used are recombination (or crossover) and mutation. The crossover operators is $n-$point crossover. The mutation operator changes a given value within the network to a random value within the possible range for the variable.

All controller networks, deterministic and non-deterministic RBNs and SFBNs, are represented as an array of nodes, each comprising a Boolean function number (between 0 and 2^{2^k} for RBNs), an initial state, and a set of input nodes, where each input is indicated by its position within the node array. It has been shown in (Gershenson, 2004) that a RBN's capacity for computation is maximal when it is in the critical regime; therefore, a value of $k = 2$ is used for RBNs, meaning that each node has precisely two inputs. The connectivity of each node in an SFBN is determined by sampling the power law distribution; for controllers, the number of connections for a particular node can change via mutation, so long as the power law distribution is maintained.

The coupling terms indicate the nodes in the controlled network (deterministic and non-deterministic RBN and SFBN) whose state will be changed by the controller network (i). These are the control interventions (C_I) and the nodes in the controlled network whose state will be copied back to the controller network, the feedback connections (C_F); see Figures 5.1, 5.2 and 5.3. Inputs to the controller networks which are fed back from the controlled network are always delivered by over-writing the states of nodes at the beginning of its node array. Control outputs are always read from the state of nodes at the end of the array. The number of coupling terms is uniformly sampled from and bounded to the range $[1, 5]$. The mutation operator can add, remove or modify coupling terms.

For all the experiments, the population size is 500 and run over 100 generations. The size of the tournament in tournament selection is 3 and the elitism size is 1. The crossover rate is 0.15 and the mutation rate is 0.06. Crossover points always

fall between node boundaries. These values were used in previous work on evolving artificial GRN models (Fuente et al., 2013), (Lones et al., 2014), and were found to be suitable choice in early experiments when a parameter sweep was carried out. Regardless to the kind of controller network used to carry out a control task, the controller network size is fixed and set to be 15. The controller network size was chosen after running several experiments using different sizes of controller networks in the range $[5 - 30]$.

The controller network fitness is a measure of the Hamming distance between the controlled network final state and the target state, after a control period of 50 time steps of the controlled network. In the process of control, two timing parameters are used. The first timing parameter determines how many steps (t_r) the controller network will execute for each step (t_d) of the controlled network, with values above 1 allowing the controller network to execute faster than the controlled network. The second parameter determines how often the controller network is executed, in terms of the number of steps of the controlled network.

Both timing parameters are uniformly sampled from and bounded to the range $[1, 50]$. Figure 5.3 shows an example of how a controller network is linearly encoded. The efficacy (or effectiveness) of a controller's interventions are measured using a fitness function that return the Hamming distance between the target state and the actual state that is reached by the end of a control period of 100 times steps of the controlled networks. This is linearly scaled to the interval $[0, 1]$, where a fitness of 1.0 indicates that the target state was reached (see equation 5.1). Let d_t be the distance from the target network's state at the end of evaluation to its target state, Net_{Size} be the target network size and fit be the fitness value. If target state is met, $d_t = 0$ therefore, $fit = 1$.

$$fit = 1 - \frac{d_t}{Net_{Size}} \tag{5.1}$$

Algorithm 2 Genetic Algorithm General Procedure

1: Initialise population P_{init}
2: Evaluate population P_{init}
3: **while** *stopping criterion not met* **do**
4: Select the best parents to produce children
5: **for** $(parent1, parent2 \in P_{init})$ **do**
6: Crossover parents from P_{init} and put in $P_{children}$
7: Mutate parents from P_{init} and put in $P_{children}$
8: **end for**
9: Evaluate children $P_{children}$ (new population)
10: $P_{init} \leftarrow P_{chidren}$
11: **end while**
12: Replace the previous population with the new population

Algorithm 3 Training BN with GA

$P \leftarrow \{\}$
for *popsize* times **do** ▷ initialize population
    $P \leftarrow P \cup \{\text{new random BN}\}$
end for
for each $p_i \in P$ **do** ▷ evaluate population
    EVALUATE(p_i)
end for
for *maxgen* times **do**
    $P' \leftarrow \{\}$
    while $|P'| < |P|$ **do** ▷ create child solutions
        $parent_1, parent_2 \leftarrow$ TOURNAMENTSELECT(p_i)
        $child_1, child_2 \leftarrow$ RECOMBINE$(parent_1, parent_2)$
        $P' \leftarrow P' \cup$ MUTATE$(child_1) \cup$ MUTATE$(child_2)$
    end while
    for each $p_i \in P'$ **do** ▷ evaluate child population
        EVALUATE(p_i)
    end for
    $P \leftarrow P'$ ▷ replace population with child population
end for
return P member with highest fitness

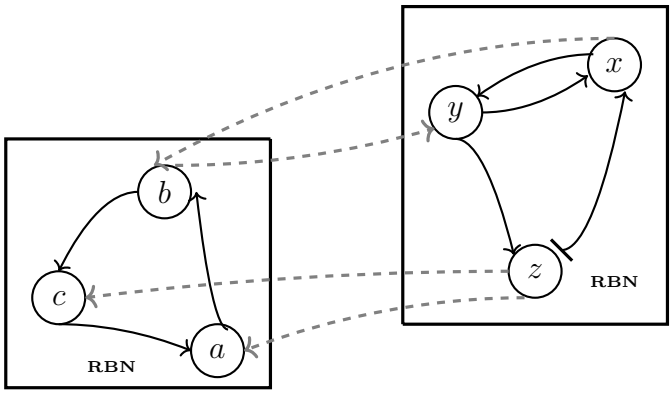
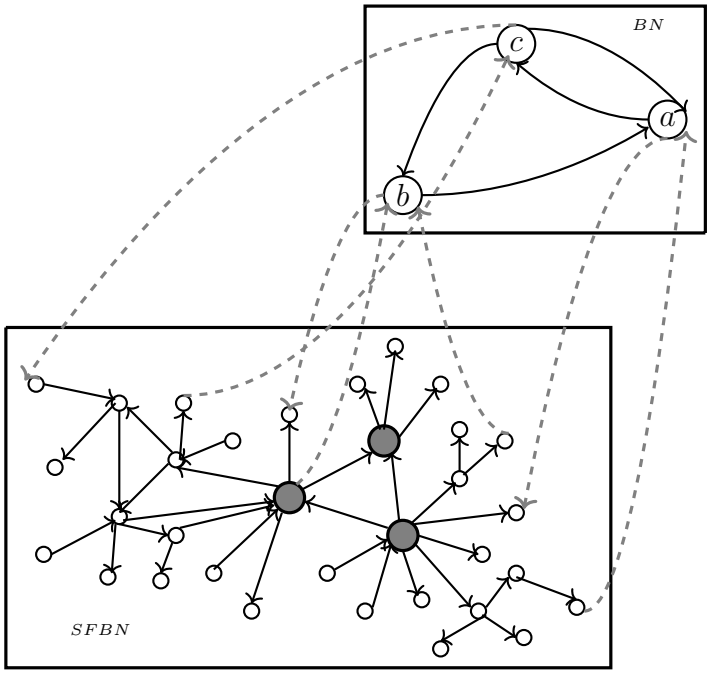
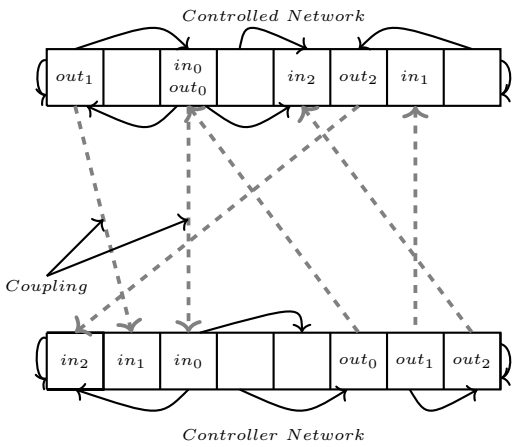


Figure 5.1: Coupled random Boolean networks.



(a) Coupled Boolean network and scale free Boolean network.



(b) Linear encoding used by the evolutionary algorithm.

Figure 5.2: Boolean network coupled to a scale free Boolean network, also showing the linear encoding used by the evolutionary algorithm. Grey dashed arrows indicate coupling between controller network and controlled network.

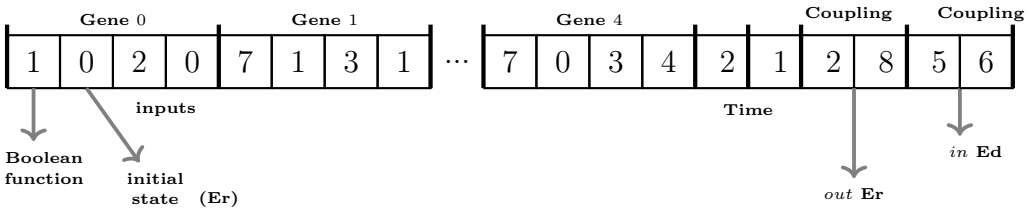


Figure 5.3: Example of a Boolean network's genetic representation (genotype). The timing and coupling terms indicate that this network is iterated twice each time it is executed, it is executed after every step of the controlled network, its control outputs (interventions) are copied to nodes 2 and 8 of the controlled network, and its feedback (*in*) inputs from the controlled network are copied from nodes 5 and 6. **Ed** and **Er** are respectively controlled and controller networks.

5.2 Control tasks

A genetic algorithm is used to generate and optimise BNs. The fitness function (see algorithm 4) measures how well a controller RBN controls other variants of BNs. In this control process, the focus is on the task of state space targeting, which means learning a control intervention that pushes a controlled BN (BN_d) to a particular point in its state space. This problem is similar to the biological problem of controlling a GRN so that it moves to and remains within a particular region of its phenotype space. All nodes in the controlled network have their expression state (S_d) set to 0 at the start of the control task, to maximise the initial distance from the target.

For simplicity and clarity, the target state is all-ones, meaning that every node in the controlled network achieves a Boolean state of 1. However, this state is no easier or harder to reach than any other arbitrary state for a particular sample of controlled BNs, and is not similar to the max-ones problem in the genetic algorithms literature. It is probable that, in practice, some controlled networks will be uncontrollable. Also it is plausible that the solution space will be hard to traverse for most controlled BNs. For example, a solution which leads the controlled network to a state of all-but-one nodes turned on is unlikely to be proximal to a solution which leads the controlled network to the optimal state.

In order to avoid bias, the controlled BNs are randomly sampled. This means that, for many of the randomly sampled networks, it will not be possible to reach the optimum. Instead, it is intended that the fitness distribution over a number of runs will give a general insight into the ability of evolved BN controllers (BN_r) to influence the dynamics of the controlled networks, and a measure of the degree to which they are able to achieve this. This gives a more general insight than looking at their ability to control particular BNs derived from the biological literature, whose topologies and dynamics may not be representative of the wider class of GRNs. Both RBNs and SFBNs are considered, for controller and controlled BNs. All nodes in the controller network have their initial expression state (S_r) randomly generated.

For completeness, each pairwise combination has been studied and analysed, i.e. deterministic RBN controllers controlling deterministic RBNs, deterministic RBN controllers controlling SFBNs, controller SFBNs controlling deterministic RBNs, deterministic RBN controllers controlling non-deterministic RBNs, non-deterministic RBN controllers controlling deterministic RBNs and controller SFBNs controlling SFBNs. Different sizes of the controlled network are considered in the range $[20, 50]$, were chosen to match the size of biological networks models used in Chapter 6. This is also done for testing proposes. For each combination of controller BN type, controlled BN type and controlled network size, 20 consecutive runs of the EA are performed, each with a (very likely unique) randomly generated controlled network. For controlled SFBNs, scale free exponents in the range $\alpha \in [2, 2.5]$ are used.

To limit the combinatorial space of experiments, the controller network size is fixed at 15 nodes (see Section 5.3.4). In early experiments, the optimisation process was found to be relatively insensitive to controller size beyond a certain threshold (Taou et al., 2016a). This may reflect a trade-off between the greater computational resources available to larger controllers and the increased size of the search space that needs to be traversed in order to optimise them. However, it is also an indication that even relatively small BNs are expressive, and are capable of generating the dynamics necessary to solve the control task. This is fortunate, since large Boolean circuits remain challenging to implement using synthetic biology principles.

Algorithm 4 Evaluating a BN on a control task

$BN_d \leftarrow$ new random BN ▷ controlled BN
$S_d \leftarrow (0, 0, 0, \ldots), S_t \leftarrow (1, 1, 1, \ldots)$ ▷ initial and target states
$t \leftarrow 0$
while t within control period **do**
    $i \leftarrow 0$
    for each $c \in C_f$ **do** ▷ feedback from controlled BN to controller BN
        $s_{r_i} \leftarrow s_{d_c}, i \leftarrow i + 1$
    end for
    for t_r times **do** ▷ execute controller BN
        UPDATE(BN_r) ▷ apply each node's update function
    end for
    $i \leftarrow |BN_r|$
    for each $c \in C_I$ times **do** ▷ apply control interventions
        $s_{d_c} \leftarrow s_{r_i}, i \leftarrow i - 1$
    end for
    for t_d times **do** ▷ execute controlled BN
        UPDATE(BN_d)
        $t \leftarrow t + 1$
    end for
end while
$correct \leftarrow 0$
for each $s_{d_i} \in S_d, s_{t_i} \in S_t$ **do** ▷ compute distance from target state
    if $s_{d_i} = s_{t_i}$ **then**
        $correct \leftarrow correct + 1$
    end if
end for
$fitness \leftarrow \frac{correct}{|S_d|}$

Algorithm 5 Training BNs with NSGA II

$P \leftarrow \{\}$
for *popsize* times **do** ▷ initialise population
    $P \leftarrow P \cup \{\text{new random } BN\}$
end for
for each $p_i \in P$ **do** ▷ evaluate population
    EVALUATE(p_i)
    compute $fitness(p_i)$ for each $p_i \in P$
end for
for *maxgen* times **do**
    $P' \leftarrow \{\}$
    $P \leftarrow$ RANK(P) ▷ NSGA-II style ranking
    $\{p_0, \ldots, p_{popsize/2}\}$ ▷ remove lower ranks
    $P' \leftarrow P$
    repeat ▷ breed child population
        $parents \leftarrow$ SELECT PARENTS(p_i)
        $P \leftarrow P'$ ▷ replace with children population
        $children \leftarrow \{\}$
        for $(p_1, p_2 \in parents)$ **do**
            $(child_1, child_2) \leftarrow$ RECOMBINE(p_1, p_2)
            $children \leftarrow$ MUTATE($child_1$)
            $children \leftarrow children \cup$ MUTATE($child_2$)
        end for
        EVALUATE($children$)
        $P' \leftarrow P' \cup children$ ▷ get best children population
    until $|P'| = popsize$
    $P \leftarrow P'$ ▷ replace with children population
end for
return P' member with highest fitness

5.3 Results

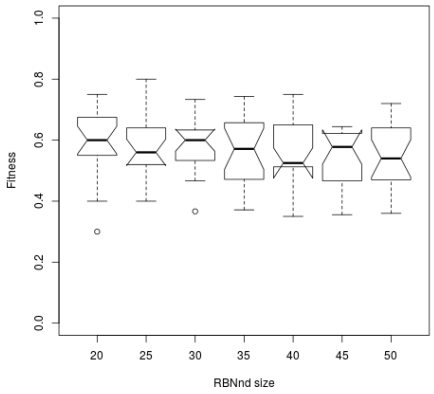
5.3.1 Controlling RBNs

To provide a benchmark for optimisation, the natural dynamics of randomly sampled non-deterministic RBNs (RBN_{nd}) (with probability $p = 0.7$ and $p = 0.3$) and deterministic RBNs were measured using the same fitness function used to evaluate controller networks. Figures 5.4a, 5.4c and 5.5a show the fitness distributions in this case. This indicates the level of fitness (i.e. distance to the target state) that can be achieved when no control is exerted over the target network, and gives a clear indication of how well the proposed control method works when applied to various target networks. It can be seen from these figures that without carrying out any control on them, randomly sampled networks tend towards a final state containing approximately equal numbers of 0s and 1s, indicated by fitness around 0.5 on average.

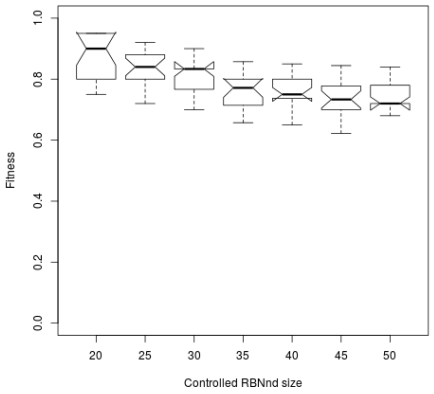
Figures 5.4b, 5.4d and 5.5b show the fitness distributions achieved when controller networks were evolved to perform state space targeting in randomly sampled deterministic and non-deterministic RBNs. From these plots, it is obvious that fitness values are much higher on average when a controller network is used, indicating that both controller RBN types can be evolved to guide other RBNs towards particular parts of their state space. Unsurprisingly and as expected, most runs do not find optimal controller networks for the randomly sampled target networks.

After observing and analysing Figures 5.4 and 5.5, it seems that, on average, there is little difference in the difficulty of the control problem regardless of whether the controlled networks are deterministic or non-deterministic RBNs. However, it is relevant to notice that the fitness distributions for deterministic RBN targets are generally wider, indicating that there may be more instances that are hard to control. It is also notable that for the deterministic RBN, the fitness distribution remains similar regardless of the size of the controlled network. This is not the case for non-deterministic RBNs, when smaller network appear significantly easier to control. This presumably reflects differences in the dynamical behaviours when

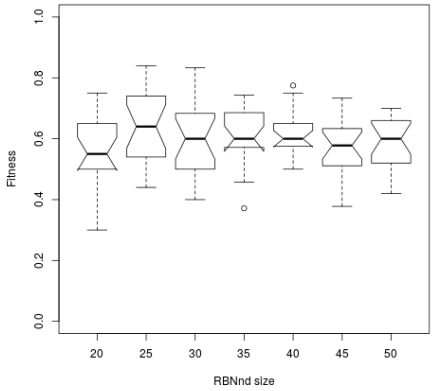
non-deterministic updates are used, for instance the loss of stable attractors.



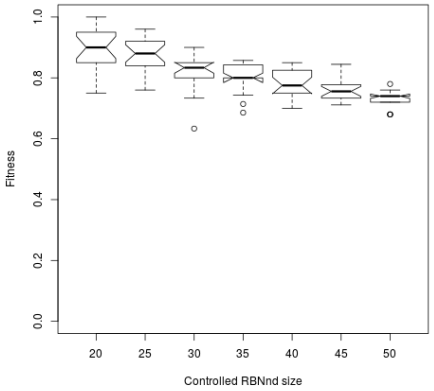
(a) RBN$_{nd}$ natural dynamics, $p = 0.7$



(b) RBN controlling RBN$_{nd}$, $p = 0.7$

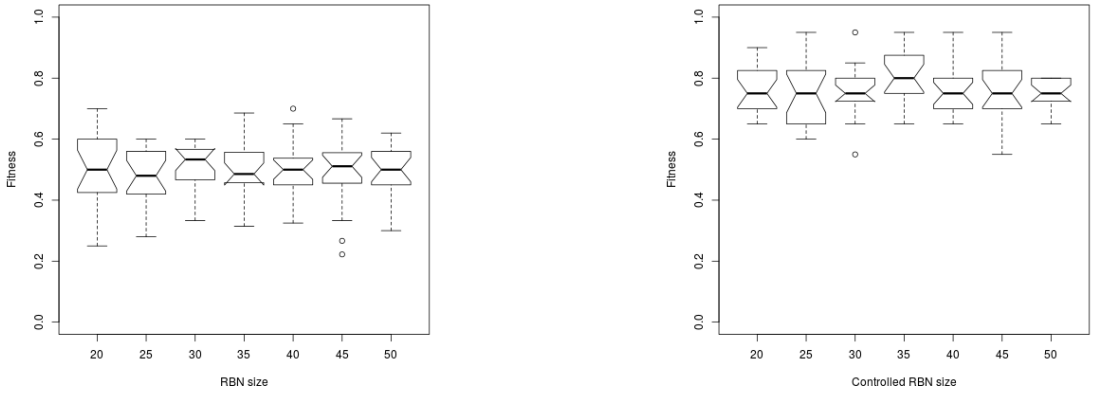


(c) RBN$_{nd}$ natural dynamics, $p = 0.3$



(d) RBN controlling RBN$_{nd}$, $p = 0.3$

Figure 5.4: Fitness distributions of non-deterministic (RBN$_{nd}$) following their natural dynamics (a,c) and under control (b,d), with probability $p = 0.3$ and $p = 0.7$. High fitness values are better. Notched box plots show summary statistics over 20 evolutionary runs. Overlapping notches indicate when median values (thick horizontal bars) are not significantly different at the 95% confidence level.



(a) RBN natural dynamics

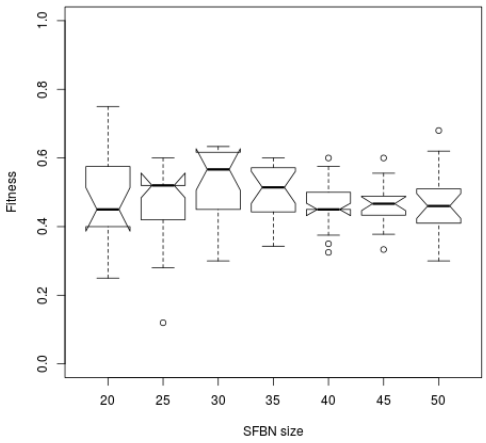
(b) RBN controlling RBN

Figure 5.5: Fitness distributions for RBNs following their natural dynamics and under control.

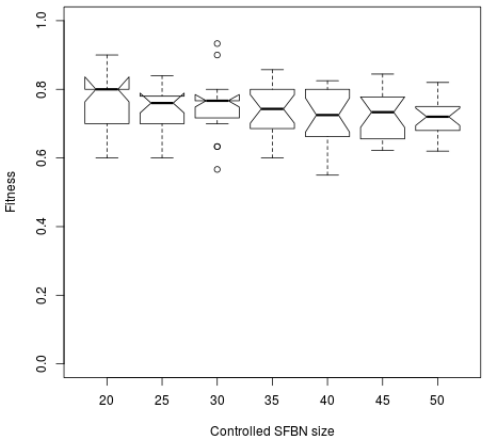
5.3.2 Controlling SFBNs

Figure 5.6a shows the fitness distributions for SFBNs in the absence of control, i.e. whilst following their natural dynamics. This shows that, without control, SFBNs also tend towards a final state containing approximately equal numbers of 0s and 1s, indicated by fitness around 0.5 on average. By comparison, Figures 5.6b–5.6d show the fitness distributions of controller networks evolved to carry out state space targeting in randomly sampled SFBNs. It is again clear that fitness values are much higher when a controller network is applied, showing that deterministic RBN controllers can also be evolved to push SFBNs towards a particular state in their state space.

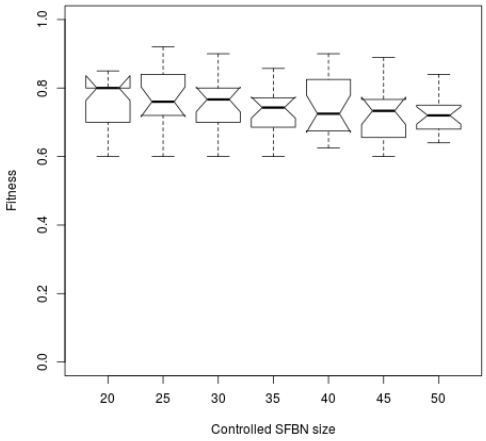
The distribution look very similar to the deterministic RBN, which suggests that the topology does not have a significant impact on controllability. Figure 5.6 also indicates that the choice of scale free exponent values had a relatively small impact on the difficulty of control, at least within the range used in these experiments.



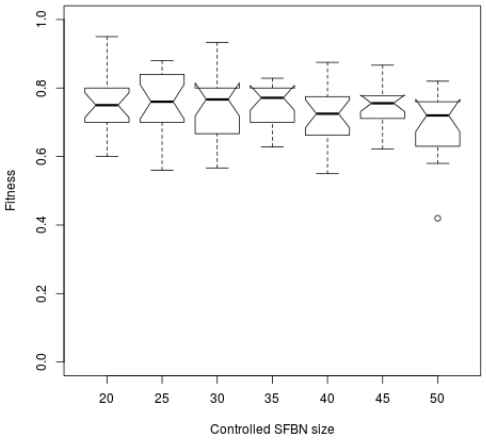
(a) No control, $\gamma_1 = 2.0$



(b) Control, $\gamma_1 = 2.0$



(c) Control, $\gamma_2 = 2.25$



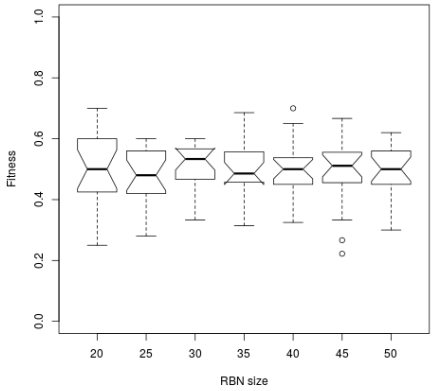
(d) Control, $\gamma_3 = 2.5$

Figure 5.6: Fitness distribution of SFBNs (a) following their natural dynamics, and (b–d) controlled with evolved RBNs, $\gamma_1 = 2.0$, $\gamma_2 = 2.25$ and $\gamma_3 = 2.5$.

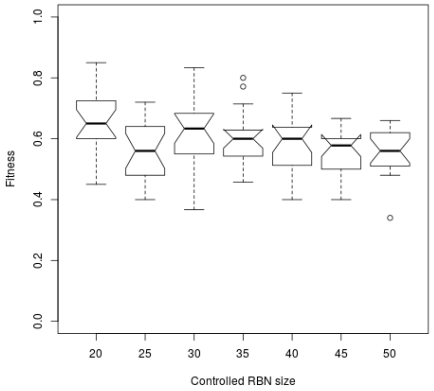
5.3.3 Varing the controller Type

For completeness and to assess the ability of other variants of BNs to carry out control tasks, experiments were also run using SFBNs and non-deterministic RBNs as the evolved controller network to control deterministic RBNs and SFBNs. Figure 5.7 and 5.8 summarises these experiments. The result show that non-deterministic RBNs can be optimised to control the dynamics of RBNs. Unlike deterministic and non-deterministic RBN controllers, SFBN controllers surprisingly appear to be significantly more difficult to evolve to carry out control, regardless of whether the

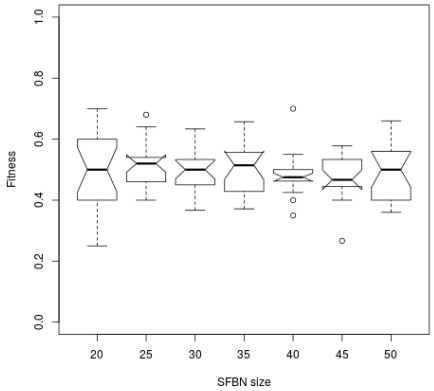
target networks are RBNs or SFBNs. This is in contradiction with the previous assumptions that SFBNs are more evolvable than networks with uniform connectivity (Aldana, 2003). Nevertheless, it does suggest that topology is an important consideration when optimising BNs to carry out control.



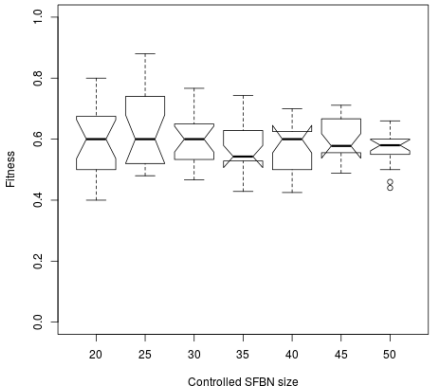
(a) RBN natural dynamics



(b) SFBN controlling RBN

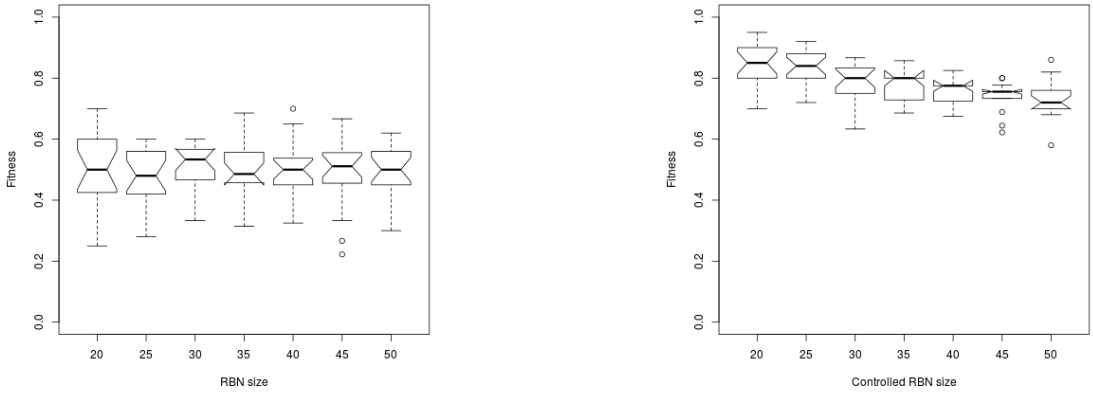


(c) SFBN natural dynamics, $\gamma_2 = 2.25$



(d) SFBN controlling SFBN, $\gamma_2 = 2.25$

Figure 5.7: Fitness distributions for SFBNs evolved to control RBNs and SFBNs.



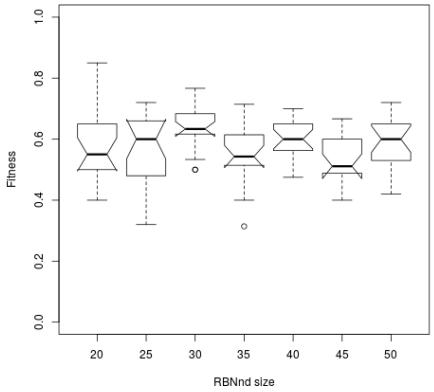
(a) RBN natural dynamics

(b) RBN_{nd} controlling RBN

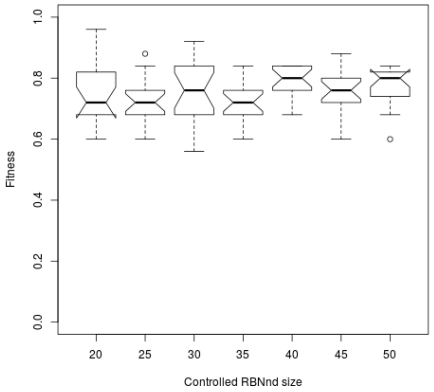
Figure 5.8: Fitness distributions for non-deterministic RBN_{nd} optimised to control deterministic RBNs. The probability used is 0.7.

5.3.4 Variable-Size Controllers

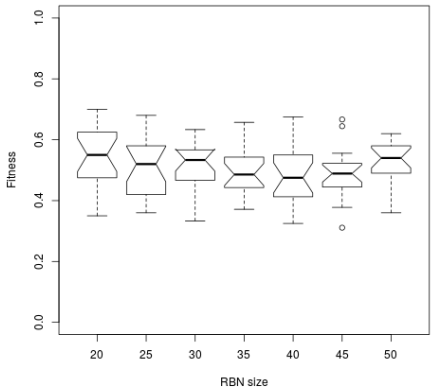
To analyse the effect of the controller network size on the control process, additional experiments were performed using a controller network whose size can change during the course of evolution. Figures 5.9 gives an indication of how the control process can be influenced when the controller network size is not fixed for each of the 20 evolutionary runs, i.e. nodes can be added and removed by the mutation operator. In this experiment the size of the controller network is in the range $[10, 25]$. At the end of several executions, it was found that the controller network size varied between 11 and 17. A comparison between Figure 5.8 and 5.9 shows that the fitness distributions are very similar, showing that controller network size variation within a certain range does not have a significant effect on the control process. It also, validates the choice of the controller network size of 15 in the earlier experiments.



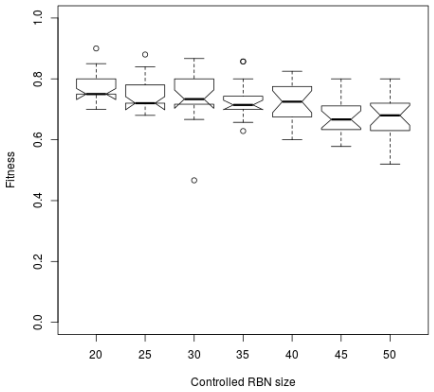
(a) RBN$_{nd}$ natural dynamics



(b) RBN controlling RBN$_{nd}$



(c) RBN natural dynamics



(d) RBN$_{nd}$ controlling RBN

Figure 5.9: Fitness distributions for deterministic RBNs and non-deterministic RBN$_{nd}$ following their natural dynamics and under control when the controller RBN and RBN$_{nd}$ size was able to vary during evolution.

5.3.5 Exploring Multiobjective Trade-Offs

This section presents the results of experiments done using MOEAs (in particular NSGA-II) to observe and analyse the trade-off between two objectives: the efficacy (or fitness value) and efficiency (or number of interventions) of the control and between four objectives: efficacy, efficiency, number of controller network time steps and how frequently it runs. It is important to note that this analysis is only applied to controlled SFBNs, because they appear to be more realistic modelsrbnNoControl of biological circuits. The controller type evolved to control the SFBN is the deterministic RBN. NSGA-II (Deb, 2001), a well known and widely used MOEA

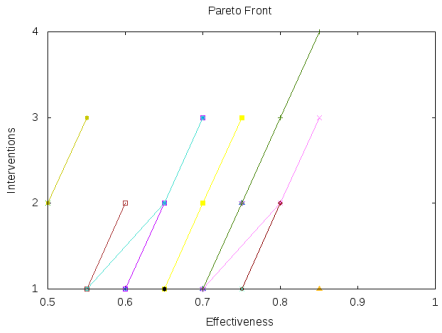
described in section 4.5.2 is used to explore trade-offs between control efficacy and efficiency.

This approach uses two objectives: the first is the distance from the target (or fitness); the second is the number of output couplings (or number of interventions) used to control the SFBN. This process is repeated using four objectives: efficacy (or fitness), efficiency (or number of interventions), controller time steps and controlled time steps in order to explore trade-offs between them. Effectiveness and the number of controlled time steps are maximized and interventions and the number of controller time steps are minimised. All the other parameters remain the same as for the standard evolutionary algorithm (see algorithm 5).

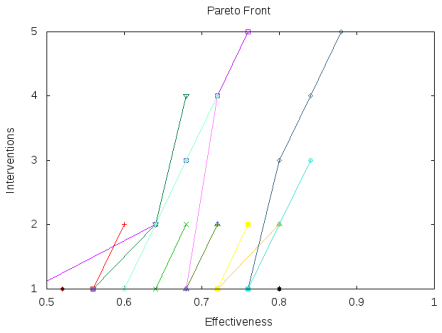
Figure 5.10 presents Pareto fronts for 20 evolutionary runs of NSGA-II for different SFBN sizes $[20 - 50]$, illustrating the trade-off between the effectiveness of control and the number of interventions (i.e. output couplings) used to implement control. It can be seen that there is a trade-off, with larger numbers of interventions generally leading to more effective control. These results suggest that, if these networks were to be implemented in real life (*in vivo*), there will likely be a trade-off between the difficulty of implementation, since more interventions are likely to be harder to implement, and the effectiveness of control. However, the extent of this trade-off will depend on the network being controlled. For the majority of the SFBN instances, there does not appear to be an advantage to having more than 2 or 3 interventions, and in many cases reasonable control can be enacted using only a single output coupling. This may reflect the topology of scale-free networks in particular, since interventions applied to hubs will have large effects on the dynamics.

5.3.6 Analysis of Evolved Controllers

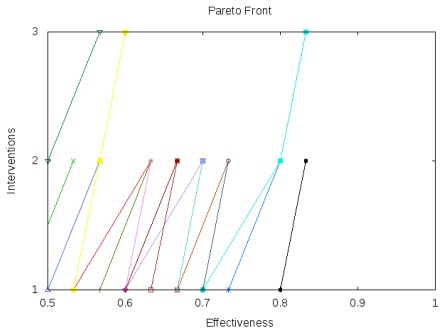
Figures 5.11–5.13 depict the control process carried out by a selection of evolved controllers. Figure 5.11 shows a deterministic RBN controlling a deterministic SFBN, showing that the controller intervenes 6 times in order to push the dynamics towards the target region of the state space. These interventions occur both whilst the controlled network is in a transient (e.g. between states 9 and 10) and when it has



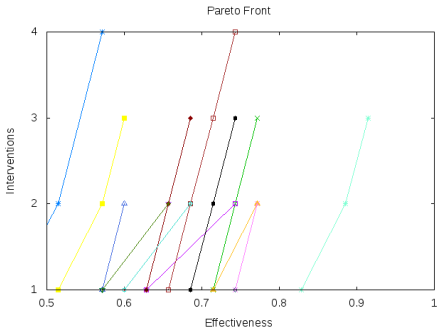
(a) SFBN size = 20



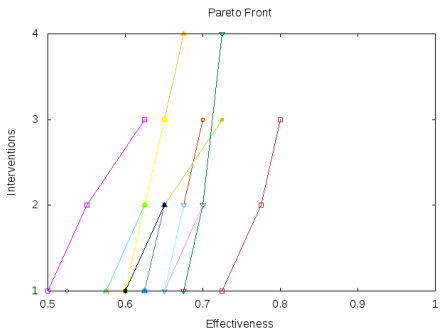
(b) SFBN size = 25



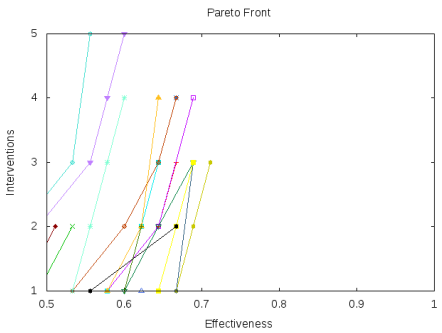
(c) SFBN size = 30



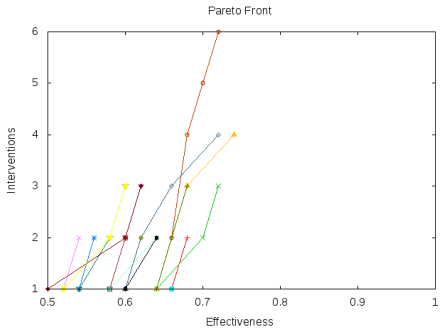
(d) SFBN size = 35



(e) SFBN size = 40



(f) SFBN size = 45



(g) SFBN size = 50

Figure 5.10: Pareto fronts, showing the trade-off between control efficacy and the number of interventions, $\gamma = 2.25$, $k = 2$. The different coloured lines (some coloured lines are duplicated) indicate the non-dominated solutions from 20 different runs, each with a different controlled SFBN.

reached an attractor (e.g. state 11 and 24). In the latter cases, the controller acts to push the controlled network out of a basin of attraction to a trajectory that takes it closer towards the target state. This is a fairly typical behaviour for many of the deterministic controllers we observed.

Figure 5.13 shows control traces for three non-deterministic controlled networks. It can be seen that the controlled system does not enter an attractor and that all interventions occur during its on-going transient behaviour. This is akin to chaos control problems, where the controllers must react to unpredictable behaviours.

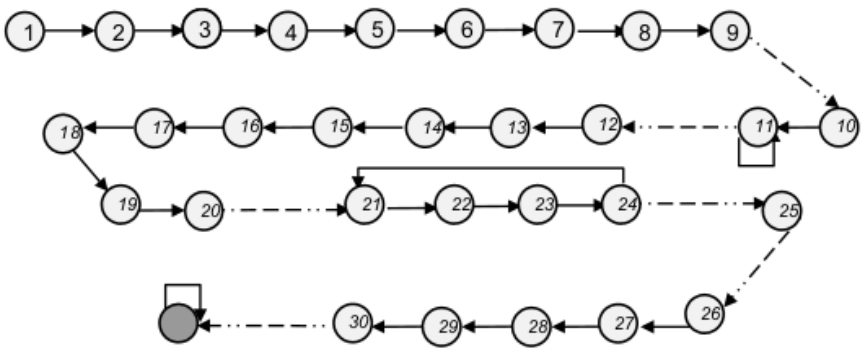
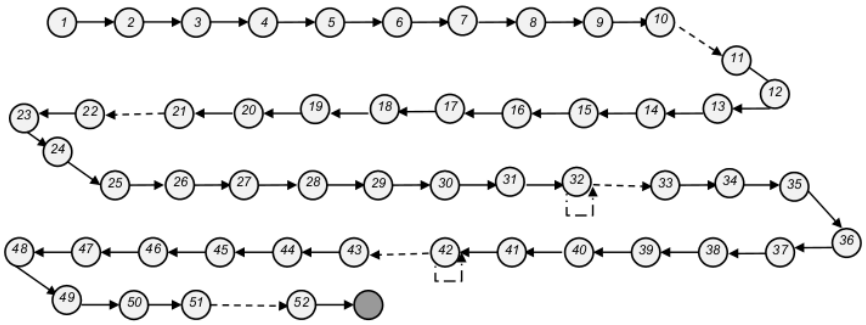


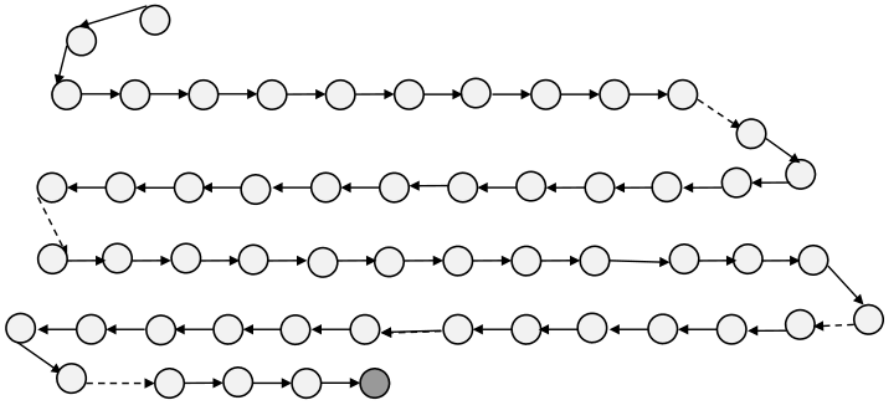
Figure 5.11: An illustration of the control process of a controlled SFBN using a controller RBN. Dashed arrows represent the controller network interventions and the dark grey node the final state of the controlled network. The initial state of the controlled RBN is all zero and the target state is all ones.

Figure 5.12: List of binary state shown in Figure 5.11

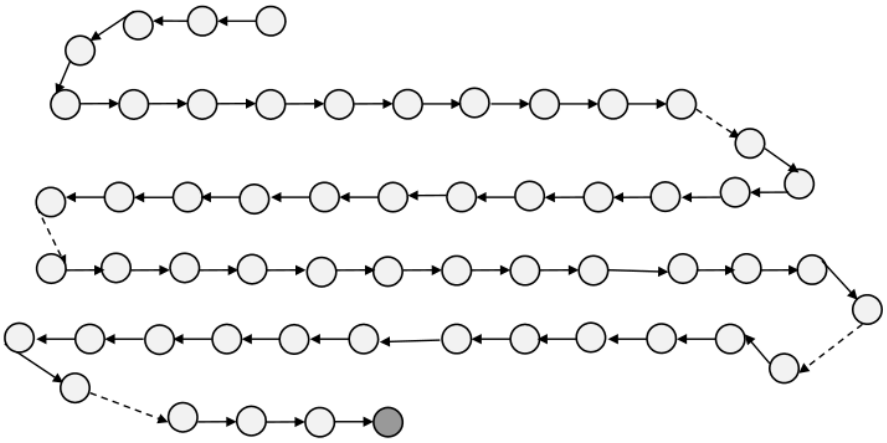
Network State	Binary representation
1	00000000000000000000
2	00110101110010101101
3	01110111100111001000
4	01110011000100011110
5	01110110000010101111
6	01110010110110111101
7	01110011010010011010
8	01110110100010101101
9	01110111110110011000
10	01110110000110011110
11	01110110010110111111
12	01110011010110111111
13	01110110000010101111
14	01110010110110111101
15	01110011010010011010
16	01110110100010101101
17	01110111110110011000
18	01110011000110011110
19	01110110000010101111
20	01110010110110111101
21	01110110010010011010
22	01110110110110111101
23	01110011010110011010
24	01110110000010111111
25	01110111110110011000
26	01110011000110011110
27	01110110000010101111
28	01110010110110111101
29	01110011010010011010
30	01110110000110011110
Final state	01110110010110111111



(c) Controlled network $N = 20$



(d) Controlled network $N = 35$



(e) Controlled network $N = 40$

Figure 5.13: An illustration of the control process for three deterministic controlled RBNs using deterministic RBN controllers. The controlled RBNs have sizes [20-40] nodes.

Figure 5.14: Binary representation of the controlled network ($N = 20$) state changes during the control process. (See Figure 5.13c)

Network state	Binary representation
1	00000000000000000000
2	10010001001000001110
3	11000010001001011010
4	10101110001000001010
5	11000010001001001110
6	10101110001010011010
7	01000010001001001110
8	01000010001011011110
9	00101110001110011110
10	00101010001110011110
11	00101010001110011110
12	01000010001000001110
13	11000010001010011110
14	01000010001010011010
15	01000010001000001110
16	11000010001010011110
17	01000010001010011010
18	01000010001000001110
19	11000010001010011110
20	01000010001010011010
21	01000110001010011000
22	01000011011001001110
23	10101110001111011110
24	00100110001011111110
25	00101110001111111110
26	00100110001111111010
27	00101110001101101110
28	10100110001111111010
29	00101110001001101010
30	10100110001101101010
31	10101110001001101010
32	10101010001001101010
33	10100110001000001110
34	11000010001011011010
35	00101110001000001010
36	11000010001001001010
37	10101110001000001010
38	11000010001001001110
39	10101110001010011010
40	01000010001001001110
41	10101110001110011110
42	01000010001011011110
43	01000010001011011110
44	00101110001110011110
45	01000010001011011010
46	00101110001100001110
47	11000010001011011010
48	00101110001000001010
49	11000010001001001010
50	10101110001000001010
51	11000010001001001110
52	01000110001001001100
Final state	10101111011111111110

5.4 Summary

Suitable and efficient control of GRNs is required in order to change the behaviour of biological cells. Nevertheless, previous work in this field indicates that this control problem is very hard, and can only be solved analytically when network topologies have a number of restrictions. In this chapter it has been demonstrated that BNs, a type of computational model of GRNs, can be optimised to control the trajectories of other BNs using evolutionary algorithms. Results shown here are obtained by using randomly sampled BNs with uniform (deterministic and non deterministic) and scale-free topologies.

Multiobjective evolutionary algorithms were used to explore the trade-off between maximising control efficacy and minimising the number of control interventions, observing that many SFBNs could be controlled with relatively few interventions, and often with only one intervention. The trade-off between maximising control efficacy and number of controlled time steps and minimising the number of control interventions and number of controller network time steps, also showed that controller timing parameters do not in general have a significant impact on the ability of the EA to fins effective controllers.

In the following chapter the methods developed in this chapter are applied to control executable Boolean models of real biological networks, in order to test whether the results obtained using the randomly sampled BNs and SFBNs apply to realistic biological systems.

Chapter 6

Controlling Boolean Models of Biological Networks

In the previous chapter Boolean networks (BNs) were applied to the problem of controlling trajectories in other variants of BNs: deterministic and non deterministic random Boolean networks (RBNs) and scale free Boolean networks (SFBNs). In this chapter, to give an indication of how well our control method works when applied to more realistic control problems, it is applied to specific Boolean models of biological networks such as the T cell receptor signalling pathway (Klamt et al., 2006), flower morphogenesis in *Arabidopsis thaliana* (Alvarez-Buylla et al., 2008) and budding yeast cell cycle regulation (Davidich and Bornholdt, 2008). As before, deterministic and non-deterministic BN controllers were evolved using evolutionary algorithms to carry out control. Controller networks' performances are observed and analysed to understand how these networks control the dynamics of these Boolean models of biological networks.

6.1 Boolean Models of Biological Networks

To investigate the ability of the proposed control method in real biological control problems, five Boolean models of biological networks were selected from the literature. These model well-known genetic regulatory systems (Mendoza and Xenarios,

2006),(Klamt et al., 2006), (Davidich and Bornholdt, 2008), (Alvarez-Buylla et al., 2008). A number of factors motivated this choice. The first objective was to look at the effect of the controlled network's size, which vary from 10 nodes to 40 nodes, on the ability to find controllers. The second is to show that the proposed control method works on networks with different state space structures. To address this, Boolean models with different numbers of stable states were chosen, since this gives some indication of the complexity of the dynamics: the selected Boolean models have between 3 and 13 stable states. Finally, the chosen Boolean models are biologically diverse, capturing a range of biological processes: morphogenesis, signalling and cell cycle regulation, that occur in a number of different species (single-celled organisms, plants, and animals). Each of these Boolean models is briefly described in this section.

6.1.1 T cell receptor signalling pathway

T cells are a subgroup of white blood cells that play a crucial role in the adaptive immune response, helping to protect the host against different pathogens such as virus and bacteria. The inappropriate activation of a T cell can lead to various autoimmune diseases. T cell receptor (TCR) is a membrane protein found on the surface of T cells which contributes to their activation by recognising antigen. A BN of the TCR signalling pathway is described in (Klamt et al., 2006) and is depicted in Fig. 6.1. It comprises 40 genes and has 8 point attractors, corresponding to different activation and proliferation cell states.

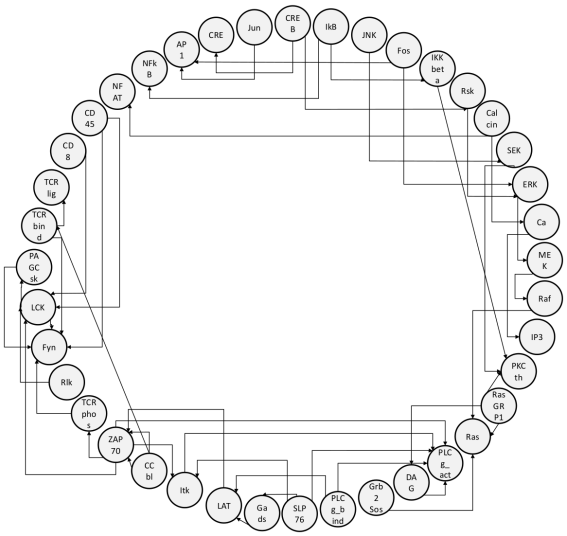


Figure 6.1: The T cell receptor signalling pathways regulatory network, showing the interactions between nodes. See (Klamt et al., 2006) for details of Boolean functions

6.1.2 T helper cell differentiation network

T helper cells, commonly called Th cells, are a type of T cell that plays a critical and key role in the adaptive immune system, where they help the immune activities of other immune cells such as B cell antibodies, plasma cells and cytotoxic T cells. T helper cells differentiate into one of the largest subcategories of cells, for example TFH, Th1, Th2, Th3, Th9 and Th17, which produce and release several types of T cell cytokines to regulate immune responses. A BN model of Th cell differentiation was developed in (Mendoza and Xenarios, 2006). This model, depicted in Fig. 6.2, captures the activities of 23 genes and has three point attractors, corresponding to different Th cell types.

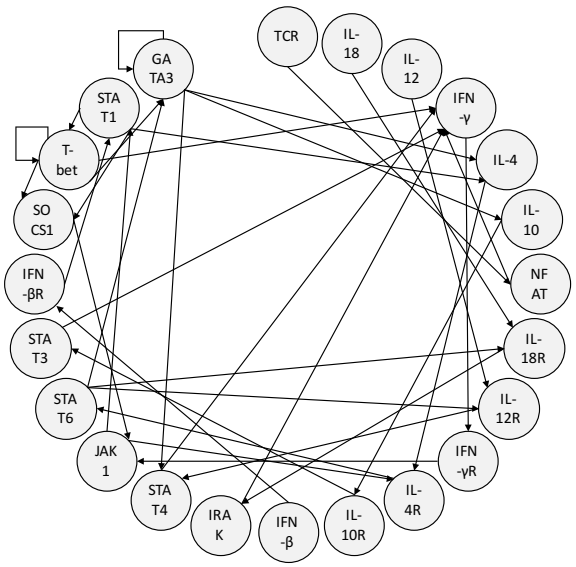


Figure 6.2: The T helper cell differentiation regulatory network, showing the interactions between nodes. See (Mendoza and Xenarios, 2006) for details of Boolean functions.

6.1.3 Flower morphogenesis in *Arabidopsis thaliana*

Morphogenesis, the development of an organism's form through the process of cell differentiation, is an important component of multicellular organisms, and often plays a role in disease development. The most widely studied models of morphogenesis concern flower development in plants, and particularly within the model species *Arabidopsis thaliana*, a small flowering plant. Flower morphogenesis occurs during the entire life cycle from groups of undifferentiated cells known as meristems. These develop into various different cell types in order to form the organs of a flower, for example sepals, petals, stamens and carpels. A BN model of flower morphogenesis in *Arabidobis thaliana* is described in (Alvarez-Buylla et al., 2008). It comprises 15 genes and has 10 point attractors, each corresponding to a different cell type. See Fig. 6.3.

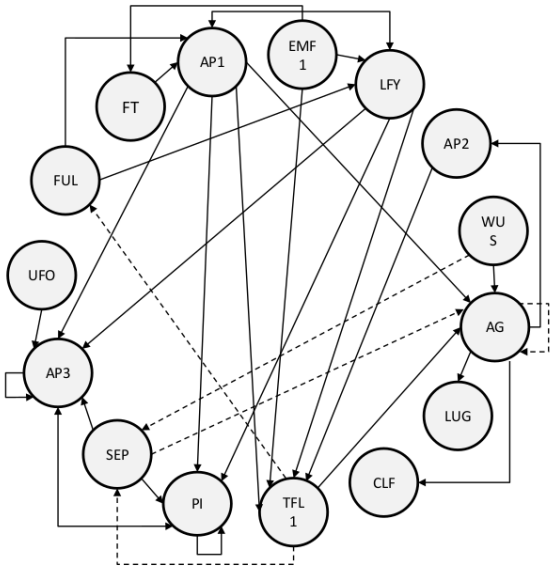


Figure 6.3: The flower morphogenesis in *Arabidopsis thaliana* regulatory network, showing the interactions between nodes. See (Mendoza et al., 1999), (Alvarez-Buylla et al., 2008) for details of Boolean functions.

6.1.4 Fission yeast cell cycle regulation

Fission yeast is the common name of *schizosaccharomyces pombe*, a unicellular eukaryote whose cells are rod-shaped and divide by medial fission. It is a well known system used to study cell growth and division, mainly because of their simple shape and their place within the eukaryotic lineage. The fission yeast cell cycle is the sequence of events that occur in a cell leading to duplication of all its components and its division into two almost identical daughter cells. A BN model of fission yeast cell cycle regulation is given in (Davidich and Bornholdt, 2008). It is formed by 10 genes and has 13 point attractors, corresponding to different stable cell states within the cell cycle. See Fig. 6.4.

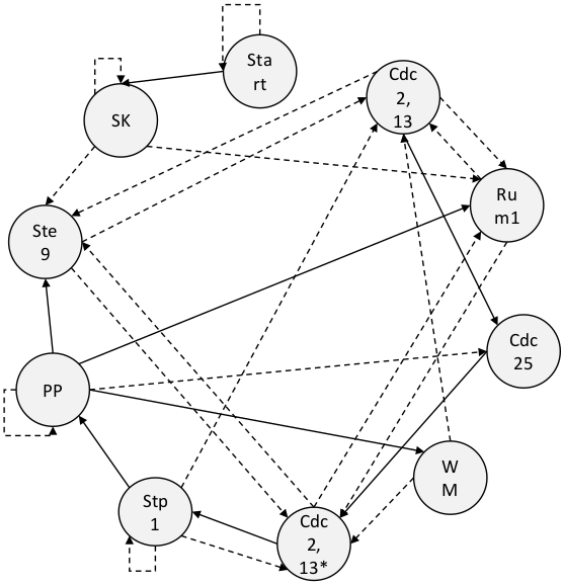


Figure 6.4: The fission yeast cell cycle regulation, showing the interactions between nodes. See (Davidich and Bornholdt, 2008) for details of Boolean functions.

6.1.5 Budding yeast cell cycle regulation

Budding yeast is another species of yeast that has been widely used to study the eukaryotic cell life cycle. As the name implies, new cells form as a bud that grows from an existing cell, rather than undergoing fission. A BN model of budding yeast cell cycle regulation is described in (Davidich and Bornholdt, 2008). It has 12 genes and 7 point attractors. See Fig. 6.5.

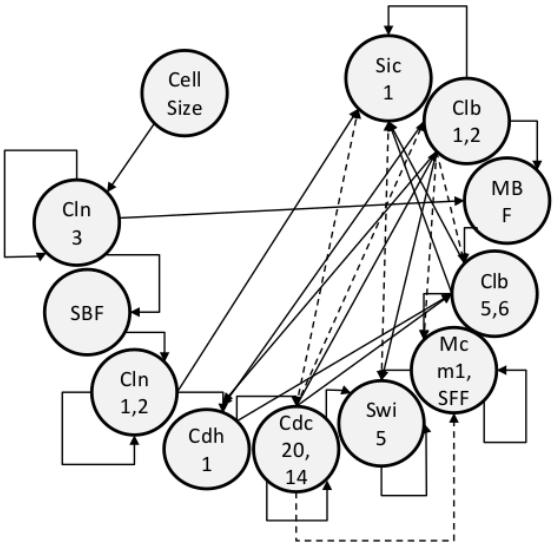


Figure 6.5: The budding yeast cell cycle regulation, showing the interactions between nodes. See (Li et al., 2004) for details of Boolean functions.

6.2 Experimental Methods and Evolutionary Parameters

The control method used in this chapter is similar to the methodology described in Chapter 5, Section 5.1.2. The evolutionary parameters, the number of generations per experiment and the controller genotype encoding (see Figure 5.3) and the formula to compute the fitness value remain the same as in Section 5.1.2, only the target state is different. Two different variants of BNs, deterministic and non-deterministic RBNs, are optimised and applied to the task of controlling Boolean models of real biological networks. The controller networks are evolved and optimised using genetic algorithms. The main idea is to apply control interventions (i.e. a series of perturbations) that guides a trajectory of a controlled network from an initial state to a particular stable state or attractor in its state space. As before, the efficacy (or effectiveness) of a controller's interventions are measured using a fitness function that returns the Euclidean distance between the target state and the actual state that is reached by the end of a control period of 50 time steps of the controlled networks.

6.3 Results

6.3.1 Controlling Trajectories form Random Initial States

In this sub-section, the control method is used to guide a trajectory beginning from a randomly sampled initial state to reach each stable state or attractor in all the Boolean models of biological networks described in section 6.1. The idea behind starting the control process from a random initial state to reach a target state is to give an indication of the general ability of the deterministic and non-deterministic controller BNs to control each network in this way. It also makes the problem more difficult to solve since the initial state is not predefined. All the experiments were run repeatedly with and without control for each target network with both controller networks.

The results from these experiments can be seen in Tables 6.1–6.5 and 6.6–6.10 (fitness distribution plots can be found in Appendix C). The Tables 6.1–6.5 and 6.6–6.10 show summary statistics for the fitness distributions of both the natural dynamics (which represent how close it gets to the target state in the absence of control) and controlled dynamics of each target biological network for each target stable state. Deterministic (Tables 6.1–6.5) and non-deterministic (Tables 6.6–6.10) controller BNs are respectively evolved to carry out the control task. All the P-values shown in the Tables were obtained using the Mann Whitney Wilcoxon test since the data did not fit a normal distribution. The results shown in Tables 6.1–6.5 and 6.6–6.10 are summarised in Tables 6.11 and 6.12, showing the mean fitness achieved and the number of target states reached in both cases with and without control, within each biological network.

For deterministic BN controllers, without control, only a small number of these attractors were reached (4/32) while, for non-deterministic BN controllers, without control, a larger number of attractors were reached (23/40). However, when attractors were reached, the standard deviations in fitness (the distance from the target) were generally large in comparison to the standard deviations in fitness with control. Examples of this are attractor 7 of the budding yeast cell cycle network, attractor 1 of the fission yeast cell cycle and attractor 1 of the flower morphogenesis in *Arabidopsis thaliana* network when standard deviations without control are respectively (0.479), (0.285) and (0.137) and with control (0) in each case. Given the difficulty of reaching the target without control, when deterministic dynamics are used this shows that, for a specific evolutionary run, the majority of the randomly sampled initial states will not be within the basin of attraction of the target attractor, making the control problems difficult to solve (i.e. they are non-trivial problems).

Tables 6.1–6.5 and 6.6–6.10 show that with control the target attractors are reached considerably more often than without control. Also, when the results from deterministic BN controllers are compared to non-deterministic BN controllers, it is apparent that the non-deterministic BN controllers perform better than deterministic BN controllers (see Tables 6.1–6.5 and 6.6–6.10). These performance differences

are statistically significant (see Tables 6.1–6.5 and 6.6–6.10). Non-deterministic BN controllers outperform the deterministic BN controllers in terms of the number of attractors reached, the median results and the standard deviation. This suggests that the stochastic property and the asynchronous updates increase performance. However, this is most likely because non-deterministic BNs are easier to control, since basins of attraction, are more permeable, not because non-deterministic BNs are better controllers.

In all the case study networks, the EA was able to find controllers that can target the majority of the steady states from a random initial state. Each time the target state was reached, the standard deviation between evolutionary runs tended to be very low ($Std.Dev < 0.04$). This means that most runs are able to find controller networks with optimal, or at least near-optimal, control strategies: the maximum likelihood estimation is 1.0 when BN controllers are successfully found and between $(0.946 - 0.997)$ otherwise.

The results shown in Tables 6.1–6.5 demonstrate that the deterministic BN controller search space has many local optima and also in the state space most of the random states probably fall far from the basin of attraction of a particular stable state, making the problem particularly hard when an arbitrary initial state is chosen. In this case, deterministic BN controllers can easily get stack in a local optima in the controller search space, where they will spend the last generations of the evolution run. In addition, it is possible that in the state space there is a presence of deceptive (or misleading) local optima. For example, in a number of cases there will not be valid transitions from states which differ by a single bit from the target. Where this is the case, there may be a possibility for using diversity preservation techniques, for instance crowding and fitness sharing, to navigate around local optima during optimisation.

Table 6.1: Fitness distributions for T cell receptor signaling pathway control, showing the normalised distances from the target for each of the system's stable states both with and without control. These results are obtained using deterministic BN controllers.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	0.996	0.009	1	0.851	0.060	0.950	1.278×10^{-08}
2	0.975	0.026	1	0.850	0.034	0.900	1.596×10^{-08}
3	0.996	0.009	1	0.843	0.075	0.975	2.745×10^{-08}
4	0.975	0	0.975	0.869	0.066	0.950	3.664×10^{-09}
5	0.996	0.009	1	0.861	0.066	0.950	9.115×10^{-09}
6	0.969	0.010	0.975	0.917	0.055	0.975	1.49×10^{-05}
7	0.975	0	0.975	0.868	0.048	0.950	5.66×10^{-09}
8	1	0	1	0.844	0.051	0.95	3.073×10^{-09}
General Mean	0.985	0.008	0.990	0.863	0.057	0.950	1.872×10^{-06}

Table 6.2: Fitness distributions for T-helper cell differentiation control.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	0.972	0.065	1	0.553	0.067	0.652	1.094×10^{-08}
2	1	0	1	0.601	0.179	0.826	3.823×10^{-09}
3	0.867	0.045	0.913	0.510	0.161	0.826	5.285×10^{-08}
General Mean	0.946	0.036	0.971	0.554	0.135	0.768	2.253×10^{-08}

Table 6.3: Fitness distributions for flower morphogenesis in *Arabidopsis thaliana* control.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	1	0	1	0.863	0.137	1	1.094×10^{-08}
2	0.926	0.030	0.933	0.561	0.124	0.800	2.75×10^{-09}
3	0.989	0.033	1	0.635	0.100	0.733	5.693×10^{-09}
4	0.933	0	0.933	0.800	0.049	0.866	2.726×10^{-09}
5	1	0	1	0.835	0.144	0.933	2.549×10^{-09}
6	0.933	0	0.933	0.217	0.150	0.800	3.027×10^{-09}
7	1	0	1	0.919	0.042	1	3.3×10^{-08}
8	1	0	1	0.624	0.074	0.733	3.062×10^{-09}
9	1	0	1	0.382	0.184	0.933	4.479×10^{-09}
10	0.996	0.015	1	0.256	0.059	0.333	4.45×10^{-09}
General Mean	0.977	0.008	0.979	0.609	0.110	0.831	7.263×10^{-09}

Table 6.4: Fitness distributions for fission yeast cell cycle control.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	1	0	1	0.442	0.285	1	3.916×10^{-08}
2	1	0	1	0.321	0.171	0.900	2.25×10^{-09}
3	0.921	0.042	1	0.594	0.102	0.700	6.823×10^{-09}
4	0.994	0.022	1	0.447	0.219	0.900	4.107×10^{-09}
5	0.994	0.023	1	0.505	0.246	0.900	5.482×10^{-09}
6	1	0	1	0.573	0.133	0.900	1.921×10^{-09}
7	1	0	1	0.484	0.121	0.800	2.377×10^{-09}
8	0.900	0	0.900	0.763	0.095	0.900	2.088×10^{-09}
9	1	0	1	0.600	0.124	0.800	12.483×10^{-06}
10	0.984	0.037	1	0.405	0.154	0.900	2.457×10^{-09}
11	0.921	0.041	1	0.552	0.134	0.800	1.274×10^{-08}
12	1	0	1	0.382	0.184	0.933	8.583×10^{-09}
13	0.994	0.022	1	0.536	0.134	0.800	3.873×10^{-09}
General Mean	0.997	0.014	0.992	0.508	0.161	0.864	1.980×10^{-07}

Table 6.5: Fitness distributions for budding yeast cell cycle control problem.

	Control			No Control			
At	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	1	0	1	0.543	0.165	0.666	2.788×10^{-09}
2	1	0	1	0.627	0.321	0.916	2.088×10^{-09}
3	1	0	1	0.442	0.416	0.916	2.25×10^{-09}
4	1	0	1	0.500	0.328	0.833	2.544×10^{-09}
5	1	0	1	0.605	0.393	0.916	2.859×10^{-09}
6	0.916	0	0.916	0.521	0.249	0.750	2.335×10^{-09}
7	1	0	1	0.434	0.479	1	1.036×10^{-05}
General Mean	0.988	0	0.988	0.524	0.335	0.855	1.482×10^{-06}

Table 6.6: Fitness distributions for T cell receptor signaling pathway control, showing the normalised distances from the target for each of the system's stable states both with and without control. These results are obtained using non-deterministic BN controllers.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	1	0	1	0.872	0.058	0.975	7.438×10^{-09}
2	0.995	0.013	1	0.885	0.057	0.975	4.387×10^{-08}
3	1	0	1	0.892	0.062	1	2.855×10^{-08}
4	0.993	0.011	1	0.900	0.030	0.950	2.411×10^{-08}
5	1	0	1	0.928	0.034	0.970	6.828×10^{-09}
6	0.997	0.007	1	0.911	0.030	0.950	1.287×10^{-08}
7	0.995	0.010	1	0.905	0.036	0.950	2.146×10^{-08}
8	1	0	1	0.910	0.040	0.975	7.452×10^{-08}
General Mean	0.997	0.005	1	0.900	0.043	0.968	2.745×10^{-08}

Table 6.7: Fitness distributions for T-helper cell differentiation control.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	0.980	0.041	1	0.569	0.111	0.782	2.268×10^{-08}
2	0.989	0.034	1	0.560	0.261	0.956	4.274×10^{-08}
3	0.873	0.027	0.913	0.528	0.152	0.782	4.18×10^{-08}
General Mean	0.947	0.034	0.697	0.552	0.174	0.840	2.893×10^{-08}

Table 6.8: Fitness distributions for flower morphogenesis in *Arabidopsis thaliana* control.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	1	0	1	0.870	0.150	1	7.845×10^{-08}
2	0.996	0.014	1	0.676	0.099	0.933	8.914×10^{-09}
3	1	0	1	0.733	0.086	1	2.079×10^{-08}
4	1	0	1	0.823	0.169	0.933	4.59×10^{-09}
5	1	0	1	0.873	0.175	1	8.939×10^{-06}
6	0.983	0.036	1	0.62	0.308	0.933	1.043×10^{-07}
7	1	0	1	0.896	0.170	1	2.71×10^{-06}
8	1	0	1	0.706	0.130	1	2.17×10^{-07}
9	1	0	1	0.752	0.265	1	2.549×10^{-08}
10	1	0	1	0.693	0.328	1	2.971×10^{-07}
General Mean	0.977	0.005	1	0.764	0.188	0.979	1.240×10^{-06}

Table 6.9: Fitness distributions for fission yeast cell cycle control.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	1	0	1	0.570	0.384	1	2.194×10^{-04}
2	1	0	1	0.494	0.379	1	1.048×10^{-05}
3	1	0	1	0.655	0.170	0.900	5.686×10^{-09}
4	1	0	1	0.620	0.345	1	5.458×10^{-05}
5	1	0	1	0.435	0.256	1	1.055×10^{-08}
6	1	0	1	0.655	0.243	1	5.589×10^{-07}
7	1	0	1	0.695	0.264	1	8.585×10^{-06}
8	1	0	1	0.515	0.318	1	9.589×10^{-07}
9	0.900	0	0.900	0.790	0.096	0.900	5.451×10^{-05}
10	1	0	1	0.635	0.181	1	2.329×10^{-07}
11	1	0	1	0.620	0.270	0.900	5.773×10^{-09}
12	1	0	1	0.595	0.203	0.900	6.123×10^{-09}
13	1	0	1	0.570	0.243	1	2.194×10^{-07}
General Mean	0.992	0	0.992	0.603	0.257	0.969	2.688×10^{-05}

Table 6.10: Fitness distributions for budding yeast cell cycle control problem.

	Control			No Control			
Attractors	Mean	Std. Dev	Max	Mean	Std. Dev	Max	p-value
1	1	0	1	0.650	0.276	1	2.436×10^{-05}
2	1	0	1	0.604	0.330	1	3.033×10^{-07}
3	1	0	1	0.570	0.397	1	4.783×10^{-06}
4	1	0	1	0.650	0.353	1	0.0001515
5	1	0	1	0.612	0.375	1	0.0001377
6	0.954	0.042	1	0.558	0.277	0.916	3.698×10^{-06}
7	1	0	1	0.587	0.442	1	0.0003859
General Mean	0.993	0.006	1	0.604	0.362	0.988	0.000101

Table 6.11: Summary of the results, showing the mean fitness and number of attractors reached for each target network both when under the control of evolved deterministic BN controllers and when following its natural dynamics (no control) from a random initial state.

		Mean Fitness		Attractors Reached		
Network name	Size	Control	No Control	Total	Control	No control
Fission yeast cell cycle	10	0.997	0.508	13	12	1
Budding yeast cell cycle	12	0.988	0.524	7	6	1
Arabidopsis thaliana	15	0.977	0.609	10	7	2
T helper cell differentiation	23	0.946	0.554	3	2	0
T cell receptor signalling	40	0.985	0.863	8	5	0

Table 6.12: Summary of the results, showing the mean fitness and number of attractors reached for each target network both when under the control of evolved non-deterministic BN controllers and when following its natural dynamics (no control) from a random initial state.

		Mean Fitness		Attractors Reached		
Network name	Size	Control	No Control	Total	Control	No control
Fission yeast cell cycle	10	0.992	0.603	13	12	9
Budding yeast cell cycle	12	0.993	0.604	7	7	6
Arabidopsis thaliana	15	0.997	0.764	10	10	7
T helper cell differentiation	23	0.947	0.552	3	2	0
T cell receptor signalling	40	0.997	0.900	8	7	1

6.3.2 Controlling Trajectories Between Attractors

In the previous section, the initial states for the control problem were randomly selected. This gives a good indication of the difficulty of carrying out control in general within these systems. However, in practice we can expect a cell's dynamics to remain close to an attractor for most of the time (Huang et al., 2009), (Huang and Kauffman, 2013), so a more realistic control problem (i.e. the kind the might

expect an *in vivo* controller to solve) is to control a trajectory from an attractor to another attractor.

Tables 6.13–6.17 and 6.18–6.22 show the results of using the proposed control method to move from one attractor to other attractors in each Boolean model of the case study biological networks with respectively deterministic and non-deterministic BN controllers. Each table shows the combinations between each of the first three attractors with the remaining attractors.

In general most of the target attractors were reached; only a few target attractors: in fission yeast cell network (3), T cell receptor signalling pathway (4) and budding yeast cell cycle (3) using deterministic BN controllers and in fission yeast cell network (3), T cell receptor signalling pathway (1) and budding yeast cell cycle (3) using non-deterministic BN controllers were not able to be reached in each set of 20 evolutionary runs. However, in the cases where the evolutionary algorithm was not able to successively find controller networks, they were at least near-optimal ($0.895 - 0.974$). Optimal controller networks for all attractor combinations across all runs were successfully found in flower morphogenesis *Arabidopsis thaliana* and T helper cell differentiation for both controller networks.

Each gray cell in Tables 6.13, 6.15, 6.16 and 6.18 indicates when respectively the deterministic and non-deterministic BN controllers were not able to reach the target attractor in all of the 20 evolutionary runs, suggesting that they are particularly hard problems. A value below 1.0 in the Tables shows that not all 20 evolutionary runs led to an optimal controller. For example in Table 6.13, when the control method is used to move from attractor 1 (initial state) to reach attractor 12 (target state) in the fission yeast cell cycle network, out of the 20 evolutionary runs 11 reached the target state ($fit = 0.953$). In general, the control problem appears to be significantly easier to solve when starting at an attractor rather then a random initial state. Also, results using non-deterministic BN controllers are better than deterministic BN controllers results.

Table 6.13: Fitness distributions for fission yeast cell cycle control, showing the normalised distances from one stable state (or attractors = At) to another using deterministic BN controllers. A fitness of 1 is optimal.

At	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1.0	1.0	0.923	1.0	1.0	1.0	1.0	1.0	0.895	1.0	1.0	0.953	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 6.14: Fitness distributions for flower morphogenesis in *Arabidopsis thaliana* control.

At	1	2	3	4	5	6	7	8	9	10
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 6.15: Fitness distributions for T cell receptor signalling pathway control.

At	1	2	3	4	5	6	7	8
1	1.0	0.975	1.0	0.970	1.0	0.974	0.971	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	0.917	1.0	1.0

Table 6.16: Fitness distributions for budding yeast cell cycle control.

At	1	2	3	4	5	6	7
1	1.0	1.0	1.0	1.0	1.0	0.946	1.0
2	1.0	1.0	1.0	1.0	1.0	0.950	1.0
3	1.0	1.0	1.0	1.0	1.0	0.917	1.0

Table 6.17: Fitness distributions for T helper cell differentiation control.

At	1	2	3
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	1.0	1.0	1.0

Table 6.18: Fitness distributions for fission yeast cell cycle control, showing the normalised distances from one stable state (or attractors = At) to another using non-deterministic BN controllers. A fitness of 1 is optimal.

At	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1.0	1.0	1	1.0	1.0	1.0	1.0	1.0	0.895	1.0	1.0	1	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.895	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.895	1.0	1.0	1.0	1.0

Table 6.19: Fitness distributions for flower morphogenesis in *Arabidopsis thaliana* control.

At	1	2	3	4	5	6	7	8	9	10
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 6.20: Fitness distributions for T cell receptor signalling pathway control.

At	1	2	3	4	5	6	7	8
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	0.982	1.0	1.0	1.0	1.0	1.0	1.0

Table 6.21: Fitness distributions for budding yeast cell cycle control.

At	1	2	3	4	5	6	7
1	1.0	1.0	1.0	1.0	1.0	0.933	1.0
2	1.0	1.0	1.0	1.0	1.0	0.946	1.0
3	1.0	1.0	1.0	1.0	1.0	0.937	1.0

Table 6.22: Fitness distributions for T helper cell differentiom control.

At	1	2	3
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	1.0	1.0	1.0

6.3.3 Analysis

The evolved deterministic and non-deterministic controller networks were of 15 genes in size and were not able to reach all target states of the studied controlled networks. Some target networks appear to be more difficult to control than others. For example, when deterministic BN controllers are used, the flower morphogenesis in *Arabidopsis thaliana* and the T cell receptor signalling both have three steady states which were not reachable; however, in both cases, the systems could be controlled to states not far from the target state (respectively $Max = 0.975$ and $Max = 0.933$). Also, the EA successfully has found non-deterministic controller RBNs that can control dynamics of all the attractors in the following networks: T cell receptor signalling pathway, flower morphogenesis and budding yeast cell cycle.

With the two type of controller networks, T-helper cell differentiation and fission yeast cell cycle both have one steady state which was not reachable ($Max = 0.913$ and $Max = 0.9$). With these results it is not evident to find a simple relationship between the difficulty to the control task and the number of attractors: For instance, using the deterministic BN controllers, the fission yeast cell cycle network, which has the largest number of attractors (13), was the easiest to control. Also, with the T helper cell differentiation which has the smallest number of attractors (3), all the

attractors were not reachable (2/3) with both deterministic and non-deterministic BN controllers.

Nevertheless, for the results obtained using deterministic BN controllers there is a mild negative correlation (-0.23) between network size and control fitness, and indeed the largest network (T cell receptor signalling) was one of the hardest to control (see Tables 6.6–6.10). In contrast, the results from using non-deterministic BN controllers show that the network size does not have a significant effect on the control task because of the very small negative correlation (-0.064) between network size and control fitness (see Tables 6.6–6.10). This may suggest that the structure of the state space has more of an impact on the control process.

Figures 6.7 and 6.6 illustrate examples of evolved control processes for T-helper cell differentiation. These show the interactions between the controller network and the controlled network. The dashed arrows show the controller network interventions. They also give an indication about the number of time the controller network intervenes in the control process. For example from Figure 6.6 it can be seen that to reach the attractor 1 (a_1) the controller network intervenes 6 times.

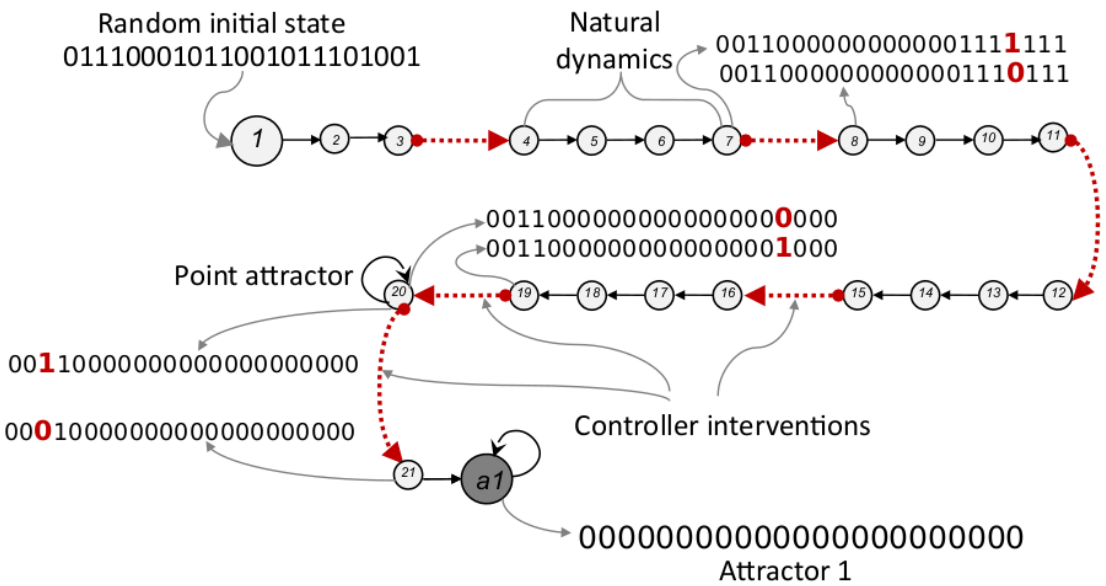


Figure 6.6: An evolved controller controlling a trajectory from a random initial state to an attractor in the T-helper cell differentiation network.

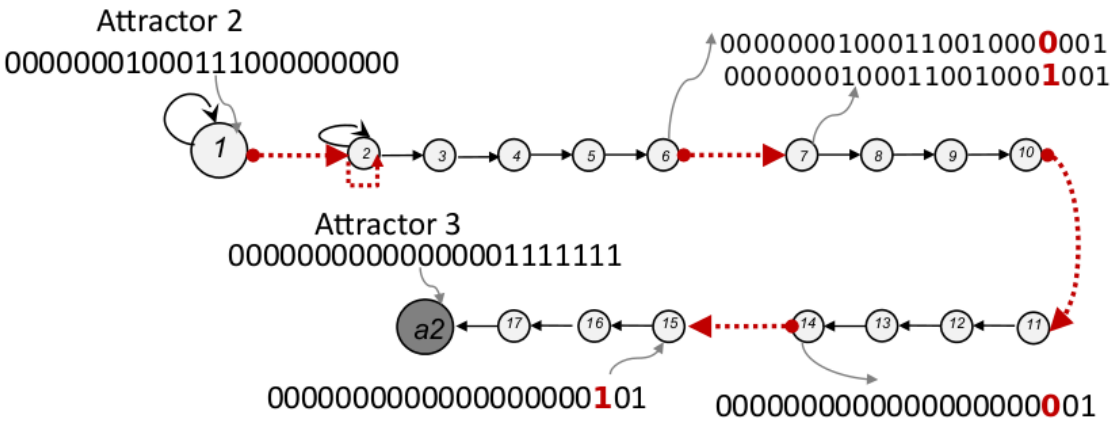


Figure 6.7: An evolved controller of controlling a trajectory from a attractor 2 as the initial state to control attractor 3 in the T-helper cell differentiation network.

Table 6.23: Binary representation of the controlled network (attractor 1 (a_1)) state changes during the control process shown in Figure 6.6.

Network State	Binary representation
1	011100010110010111101001
2	001100010000000010011100
3	001100010000000000001110
4	001100010000000000001110
5	001100000000000001001111
6	001100000000000001101111
7	001100000000000001111111
8	001100000000000001110111
9	001100000000000000111011
10	001100000000000001011101
11	001100000000000001101110
12	000100000000000001100110
13	000000000000000000110011
14	000000000000000000011001
15	000000000000000001001100
16	001000000000000001000100
17	001100000000000000100010
18	001100000000000000010001
19	001100000000000000001000
20	001100000000000000000000
21	000100000000000000000000
a_1	000000000000000000000000

6.4 Summary

In this chapter, both deterministic and non-deterministic controller BNs were applied to the task of controlling Boolean models of different biological networks. The results have shown that most of the control tasks were successfully solved in all the runs and in the few cases where optimal controllers were not found, the fitness value was close to optimal. For instance, BN controllers were not able to control all attractors in the T helper differentiation and fission yeast cell cycle networks. This demonstrates that the BN controllers have limits in terms of their behaviours during the control process. However, it is important to notice that BN controllers are evolved using evolutionary algorithms, with a set population, mutation and crossover parameters and changing these can modify current results and generate different results. Also, the use of diversity preservation techniques such as fitness sharing could help to move around local optima during the control process, and therefore obtain better results.

A comparison of the two BN controller types show that the non-deterministic BN controllers perform better than the deterministic BN controllers when applied to T cell receptor signalling pathway, T helper cell differentiation, flower morphogenesis, budding yeast cell cycle and fission yeast cell cycle control tasks, with the non-deterministic BN controllers being able to control all attractor trajectories in 3 of the 5 controlled networks by guiding each attractor from a randomly sampled initial state to a biologically-meaningful state, and the deterministic BN controllers were able to control most of the attractors in these networks. The overall results are promising and demonstrate that even standard evolutionary algorithms can solve state space targeting problems, and can do so in a way that does not require *a priori* knowledge and understanding of the target networks' dynamics and does not require a restricted topology.

Chapter 7

Summary and Conclusions

This chapter summarises the work reported in this thesis and presents the conclusions. It also validates the initial hypothesis, discusses the limitations of the experimental method, and finally provides suggestions for future work.

7.1 Summary

The work reported in this thesis started with the hypothesis that artificial gene regulatory networks can be used as bio-inspired control architectures to perform state space targeting in other artificial gene regulatory networks and models of actual biological circuits. Biological systems complex behaviours have constantly been a source of inspiration to many science fields such as bioinformatics, mathematics and computer science. This is more notable in computer science with the development of modern areas of research such as artificial intelligence (for example artificial neural networks, artificial immune systems and robotics) in which both methods and objectives are to imitate and reproduce biological system behaviours. These fields are constantly evolving.

Artificial gene regulatory network are a type of artificial biochemical network which take inspiration from the regulatory interactions between genes and with other substances in biological cells. They capture how genes regulate each other's protein expression level constantly by producing transcription factors. Genes are one of

the most omnipresent biological components, a fundamental physical and functional unit of hereditary information which usually indicates the elementary structure of a protein. Although genes play an important role in gene regulation, it has been shown that gene regulatory networks are not exclusively made of genes, but a mix of genetic and epigenetic (such as chromatin) structures.

In this thesis Boolean networks, the simplest and perhaps best known, computational model of gene regulatory networks are used to carry out state space targeting in artificial gene regulatory networks. Three variants of Boolean networks have been used as controller and controlled networks: deterministic and non-deterministic Boolean networks and scale free Boolean networks. This thesis has also explored the ability of deterministic and non-deterministic Boolean network controllers to control the dynamics of Boolean models of actual biological circuits: the T cell receptor signalling pathway, flower morphogenesis in *Arabidopsis thaliana*, T-helper cell differentiation, fission yeast cell cycle regulation and budding yeast cell cycle regulation, were used as controlled networks.

Deterministic Boolean network controllers have been optimised using evolutionary algorithms to perform state space targeting in other deterministic Boolean networks, in non-deterministic Boolean networks and scale free Boolean networks. In all these three control tasks, the objective was to guide each of these controlled networks from an initial state (set to be all zeroes) to reach the target state which is all ones. As expected, most runs did not find optimal solutions (or controller networks) for each of the randomly sampled target networks. However the fitness values are much higher on average when the control is applied than without control. It appears that there is no difference in the difficulty of the control problem regardless of the type of the controlled network, but the fitness values from deterministic Boolean networks are slightly higher than the fitness values from scale free Boolean networks.

Non-deterministic and scale free Boolean networks have been evolved to carry out state space targeting in randomly sampled deterministic Boolean networks. The results have shown that unlike non-deterministic Boolean network controllers, scale free Boolean network controllers are harder to evolve to carry out control either

in deterministic Boolean networks or in scale free Boolean networks. This is in contradiction with previous assumptions that scale free Boolean networks are more evolvable than deterministic and non-deterministic Boolean networks (i.e. networks with uniform connectivity). Also, within the range considered in this study, the size of the controlled and controller networks and the scale free exponent value had a relatively small effect on the difficulty of control. Furthermore, it has been observed that the proposed control method is relatively scalable as the size of the controlled network increases.

Multi-objective evolutionary algorithms (NSGA-II) were used to observe the trade-off between two objectives (number of interventions and effectiveness of the control) and later between four objectives (number of control interventions, efficacy, number of controller and controlled networks time steps). Scale free Boolean networks were chosen to do these experiments, because they are a more realistic model of gene regulatory networks. The analysis of the Pareto fronts have shown that control can often be performed using only a single coupling term. In addition, Pareto fronts for four objectives have shown that the number of controller network time steps and the frequency at which its runs do not have an impact on the control process.

Finally, deterministic and non-deterministic Boolean network controllers have been used to influence the dynamics of Boolean models of real biological circuits, by controlling their trajectory towards a particular target state. This has been done in two ways: controlling trajectories from a random initial state to reach each attractor (or steady state) and controlling trajectories between attractors. While for the former the control problems were randomly chosen, the latter is a more realistic control problem (i.e. it can be observed in real life). The results show that the EA is able to find controller BNs that can solve most of the control problems looked at. It has to be noted that the size of the controlled networks and the number of attractors seem to have a relatively small effect on controllability. Again the size of the controller network does not have a notable impact, as in previous experiments in this thesis.

7.2 Conclusions

The objective of this research was to find efficient control strategies that can be used to control the dynamics and genetic states of artificial gene regulatory networks. Throughout this thesis BNs were evolved using EAs to control a number of BNs and a range of Boolean models of biological networks. The analysis of the performance and behaviours of the EA and the controller and controlled networks have led to the following conclusions.

Boolean networks can control other Boolean networks

Throughout experimentation Boolean networks have been optimised to carry out state space targeting, targeting both attractors and arbitrary states in different networks. In all cases, it was shown that BNs can be optimised to govern the dynamics of the controlled networks. When using models of actual biological networks as the controlled system, most control task could be solved optimally.

Boolean network controller can be designed and optimised using evolutionary algorithms

In all the Boolean models of real biological regulatory networks studied, the evolutionary algorithm was able to find optimal controller networks that can target the majority of the attractors from random initial state and moving from attractor to attractor. In the case where evolutionary algorithms did not find optimal controller networks, they get close with a high fitness value. Evolutionary algorithms do not require knowledge of the controlled system's state space or dynamical regimes in order to solve these control problems. This is an advantage over other approaches such as semi-tensor product of matrices and algebraic approaches.

This approach also works on networks with both uniform and scale free topolo-

gies: deterministic and non-deterministic Boolean networks and scale free Boolean networks, the latter being more typical of real biological systems. The results also show that non-deterministic Boolean networks are easier to control, probably because they do not get trapped in basins of attraction like deterministic Boolean networks.

The scalability of the approach

Another interesting finding is the scalability of this approach. The fitness distributions with the controlled networks size suggest that the scalability of the approach is relatively good. In all the runs meaningful control still occurs for the largest controlled networks regardless of whether the target network is a deterministic or non-deterministic Boolean network, or a scale free Boolean network or a Boolean model of a real regulatory network. This is an advantage over control processes that use theoretic approaches.

Reducing the number of interventions

The use of multi-objective evolutionary algorithms has shown that there is a trade-off between the number of coupling terms and the control efficacy (or fitness). However, from the Pareto fronts it can be seen that a single intervention can be used to carry out control tasks in many cases. It is important to be able to use only a few interventions, especially if the controllers will ultimately be used within cells. This is because each coupling term represents a physical connection to the native GRNs. The greater the number of coupling terms, the harder the implementation becomes.

Controller Architectures

Different controller networks: deterministic and non-deterministic Boolean networks and scale free Boolean networks were evolved in this work to perform control in different target networks. Results from the experiments have shown that deterministic and non-deterministic Boolean network controllers perform much better than the

scale free Boolean network controllers. This is a slightly surprising result since scale free Boolean network controllers are generally believed to be more evolvable than Boolean networks. However, this might indicate that scale free networks are less able to implement computation, at least of the kind required for these control tasks.

7.3 Limitations of This Research

Although significant work has been done to check and validate the conclusions reported throughout this work, using theoretical understanding and experiments, there are a number of limitations of the experimental techniques which might restrict their generalisation. These limitations will constitute the main focus of future work.

The performance of the control method has not been rigorously observed and analysed upon large size models of actual biological networks. This is mostly due to the lack of availability of such models. Also, there have been very few comparisons between the performance of the control strategy used in this work and other existing control methods such as algebraic approaches (Murrugarra et al., 2016), (Hou et al., 2016). This can be explained by the absence of existing results applying control approaches to perform state space targeting in an entire Boolean network, rather these approaches target specific genes in the network which will be controlled (Kim et al., 2013).

However, comparison between network natural dynamics and dynamics after control give a clear indication about how well the proposed control method works when applied to certain problems. Also, more work is required to understand why SFBNs are hard to evolve to perform control in any Boolean networks. Looking in more detail into this might give interesting insights the nature and implementation of computation within biological networks.

There is also a need for more consideration of the practicality of evolving controllers in simulation and then refining them into synthetic biology implementation. Refining evolved BNs into actual synthetic biology realisations would involve a number of extra challenges. For instance, in this work timing parameters were evolved to allow the controller and controlled systems to operate over different timescales. This

may also be possible to do within synthetic biology implementations, e.g. using RNA interference rather than transcription factors to speed up the controller's logic, but it would not be trivial. Another issue might be limitations placed on the controller's size or topology due to the difficulty of avoiding cross-talk within synthetic biology circuits. This is still an early work, and there remains significant work to be done to show that this is a viable approach to designing synthetic GRNs.

7.4 Future Work

There are a number of issues raised by this research which need to be addressed in future work and experimentation as follow:

- Investigate the use of different type of networks for both controller and controlled networks, for instance dealing with stochasticity, different time scales.
- Apply the optimised controller networks to a wider range of real biological networks models, and use this knowledge to improve understanding of how to control complex dynamical systems.
- Explore the possibility of using other models for the controller and controlled systems, for example continuous-state models for the controller network, and agent-based models for the controlled network. For instance, there has recently been a lot of work on designing robust executable models of biological systems (Greaves et al., 2013), (Albergante et al., 2013). By using these kind of models as controlled systems, there is a potential to generate useful new biological knowledge.
- Analyse controller networks dynamics.

Appendix A

Acronyms

ABN – Artificial Biochemical Network

AGRN – Artificial Gene Regulatory Network

BN – Boolean Network

DNA – Deoxyribonucleic Acid

EA – Evolutionary Algorithm

GA – Genetic Algorithm

GP – Genetic Programming

GRN – Gene Regulatory Network

MOEA – Multi-objective Evolutionary Algorithm

MOOP – Multi-objective Optimisation Problem

mRNA – Messenger Ribonucleic Acid

NSGA II – Non-dominated Sorting Genetic Algorithm II

OGY – Ott, Grebogi and Yorke

PBN – Probabilistic Boolean Network

RBN – Random Boolean Network (deterministic)

RBN$_{nd}$ – Random Boolean Network (non-deterministic)

RNA – Ribonucleic Acid

SFBN – Scale Free Boolean Network

tRNA – Transfer Ribonucleic Acid

Appendix B

Mathematical Symbols

BN_d – controlled Boolean network

BN_r – controller Boolean network

C_F – feedback connections

C_I – control interventions

$child_1, child_2$ – Children

d_t – the distance from the last step of an evolutionary run to the target state

F, f – Boolean functions

fit – fitness value

in – input

k – connectivity

$maxgen$ – maximum number of generations

N – number of nodes

N_{Ed} – controlled network size

N_{ones} – number ones in controlled network state at the end of an evolutionary run

Net_{Size} – target network size

out – output

P – Initial population

P' – New population

P_{init} – Initial population

$P_{children}$ – Population of children (new population)

p_i – Individual form populations (P, P')

p_1, p_2 – Parents

$popsize$ – Population size

p – probability

p_{inp} – probability distribution

s – network states

S_d – controlled state

S_r – controller state

t – time

t_d – controlled time step

t_r – controlled time step

α, γ – scale free exponent

c, d, i, n – variables

χ – represents the entire network during update

Appendix C

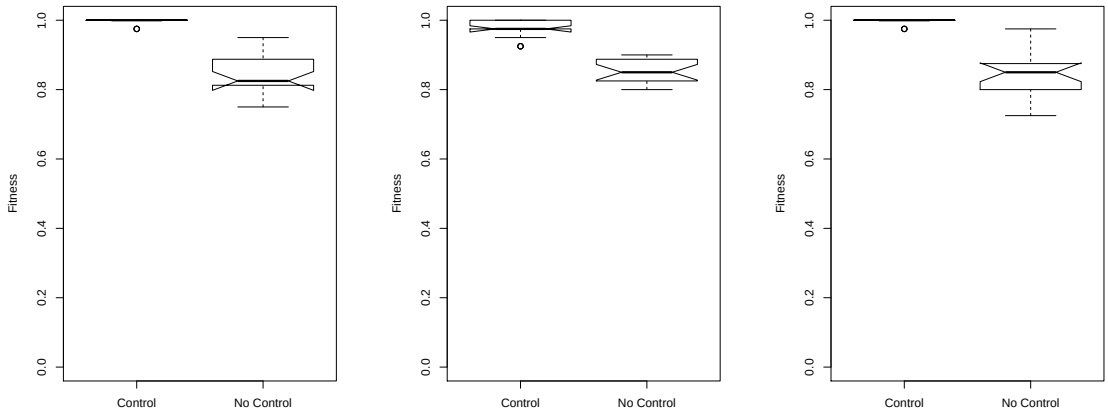
Controlling Boolean Models of Biological Networks Plots

The following plots show the fitness distributions for all the Boolean models of biological networks (T cell receptor signalling pathway, T helper cell differentiation, flower morphogenesis in arabidopsis thaliana, fission yeast cell cycle and budding yeast cell cycle) controlled in this thesis. These plots are showing the normalised distances from the target for each of the system's stable states both with and without control when starting at randomly sampled initial states. These results are obtained using deterministic and non-deterministic controller RBNs. Non-deterministic controller RBNs perform better than deterministic controller RBNs, see for example Figures C.1b, C.1d, C.1f and C.6b, C.1d, C.6f.

In addition to the control of the stable states of the Boolean models of biological networks from a random initial state, the proposed control method is used to move between attractors. The results of these experiments are shown in the following plots.

Figure C.1: Fitness distributions for the T cell receptor signalling pathway control problem.

(a) Stable state 1 (b) Stable state 2 (c) Stable state 3



(d) Stable state 4 (e) Stable state 5 (f) Stable state 7

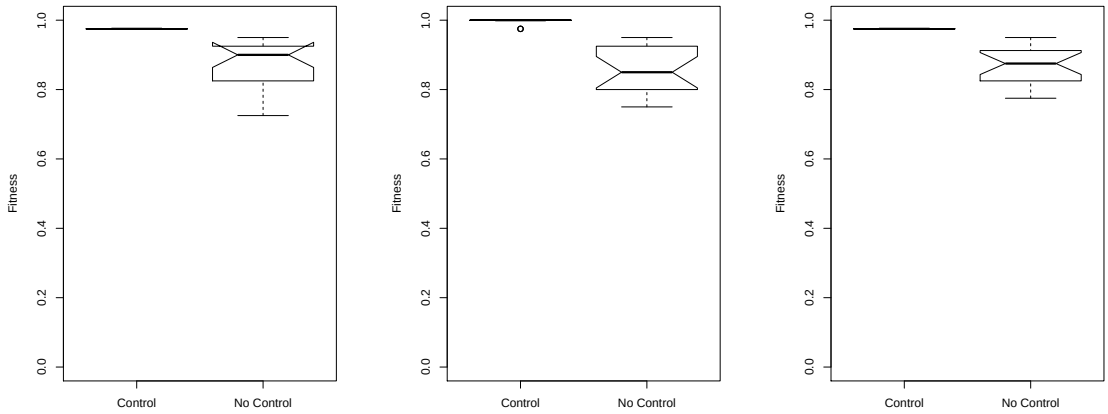


Figure C.2: Fitness distributions for the T helper cell differentiation control problem.

(a) Stable state 1 (b) Stable state 2 (c) Stable state 3

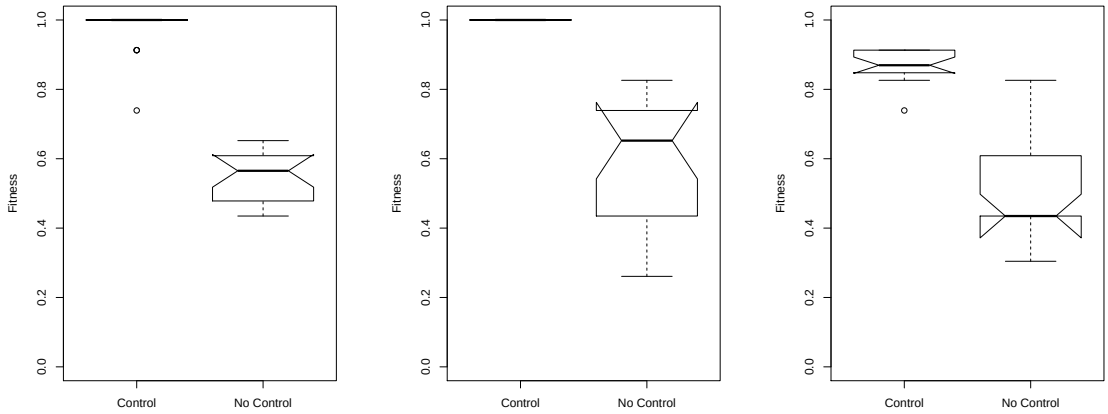
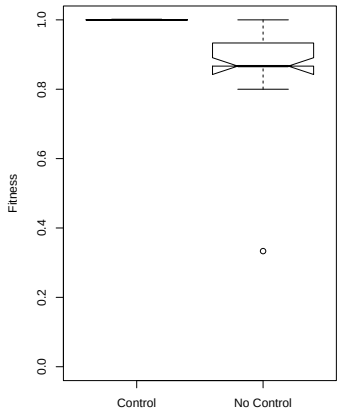
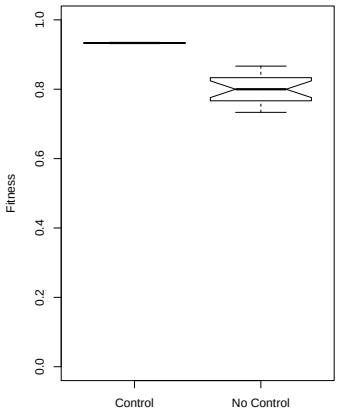


Figure C.3: Fitness distributions for the flower morphogenesis control problem.

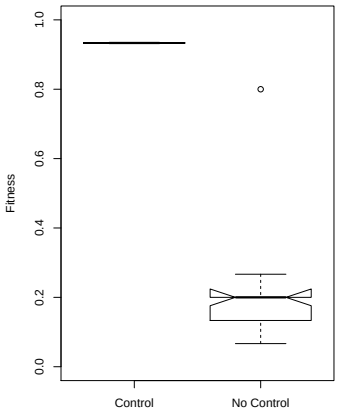
(a) Stable state 2



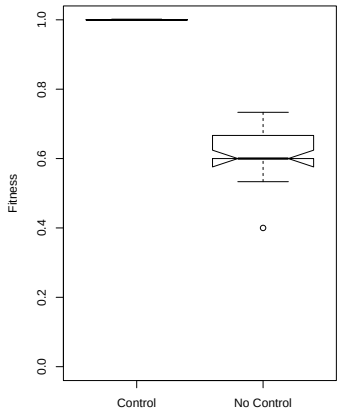
(b) Stable state 4



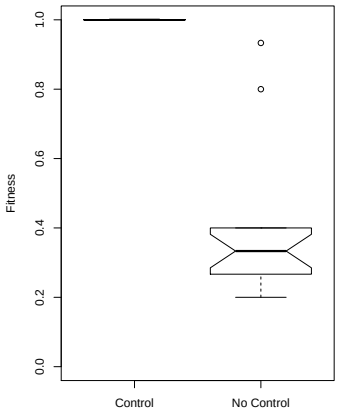
(c) Stable state 6



(d) Stable state 8



(e) Stable state 9



(f) Stable state 10

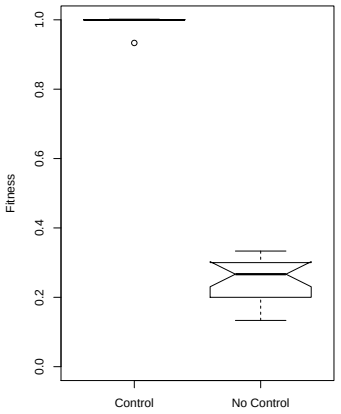
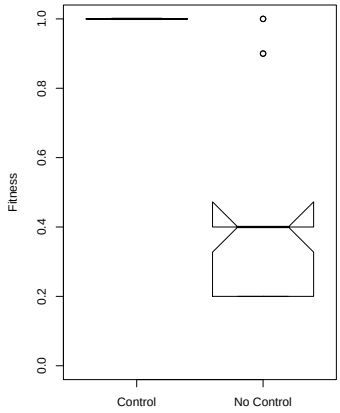
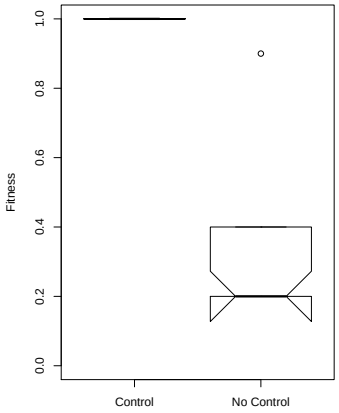


Figure C.4: Fitness distributions for the fission yeast cell yeast control.

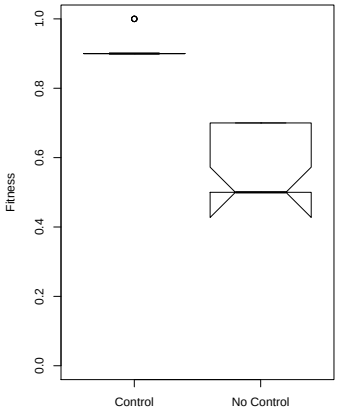
(a) Stable state 1



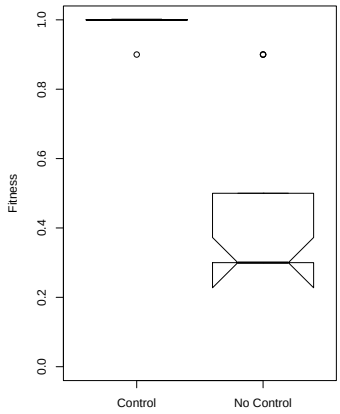
(b) Stable state 2



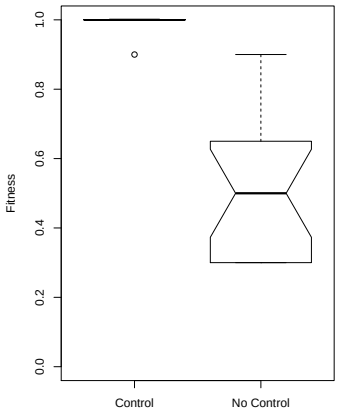
(c) Stable state 3



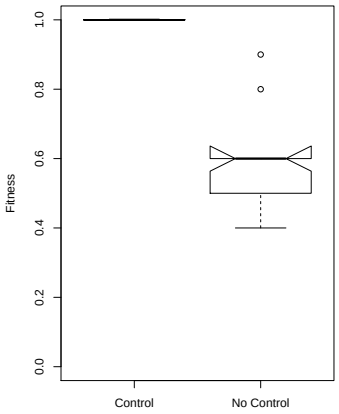
(d) Stable state 4



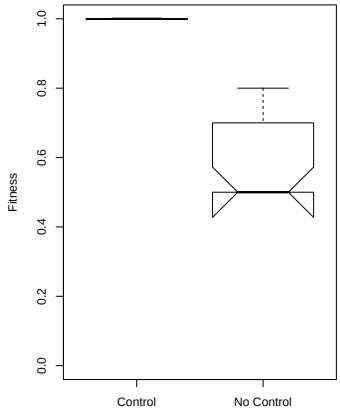
(e) Stable state 5



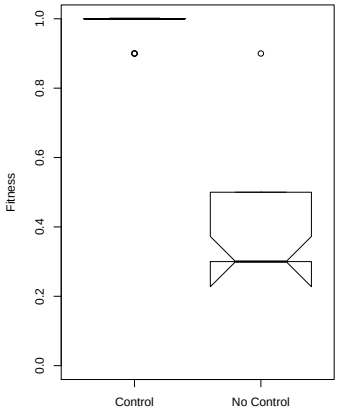
(f) Stable state 6



(g) Stable state 10



(h) Stable state 11



(i) Stable state 12

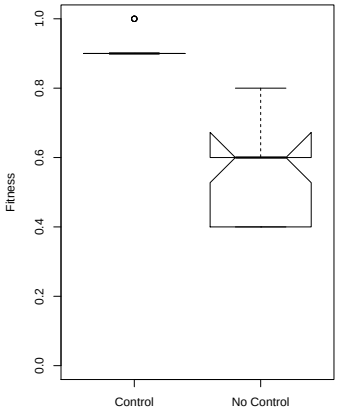
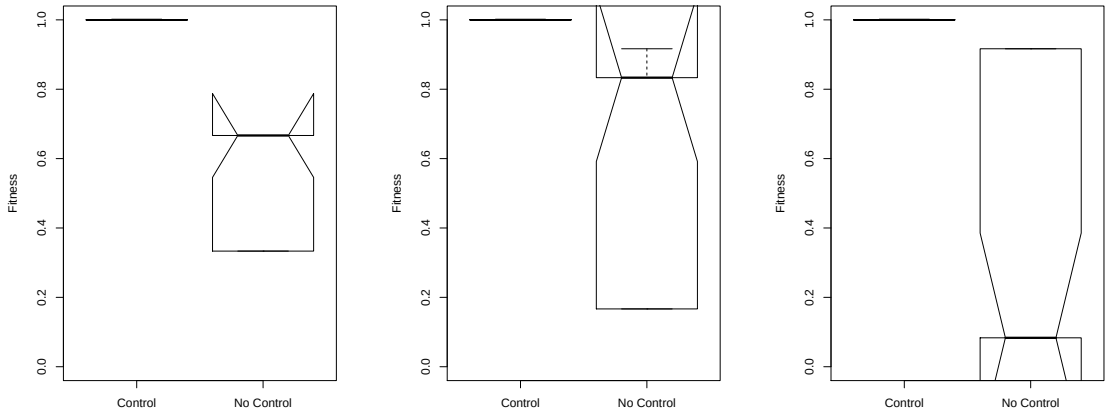


Figure C.5: Fitness distributions for the budding yeast cell cycle control.

(a) Stable state 1 (b) Stable state 2 (c) Stable state 3



(d) Stable state 4 (e) Stable state 5 (f) Stable state 6

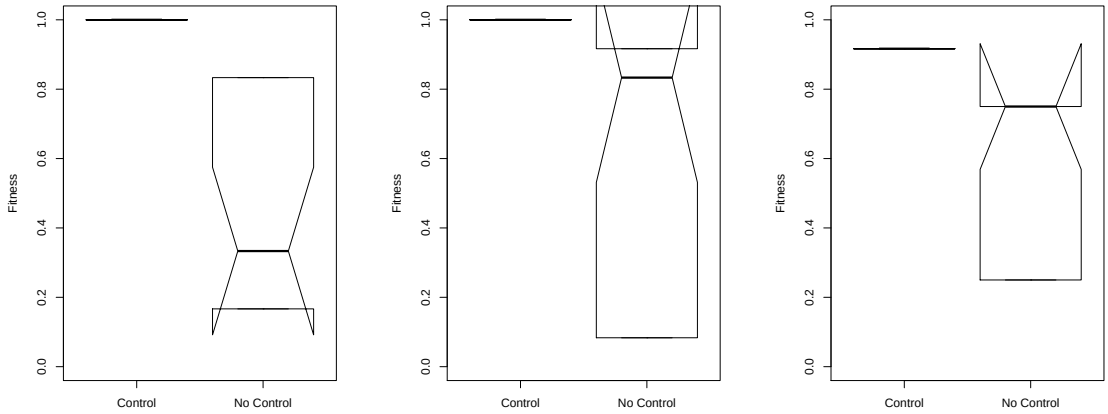
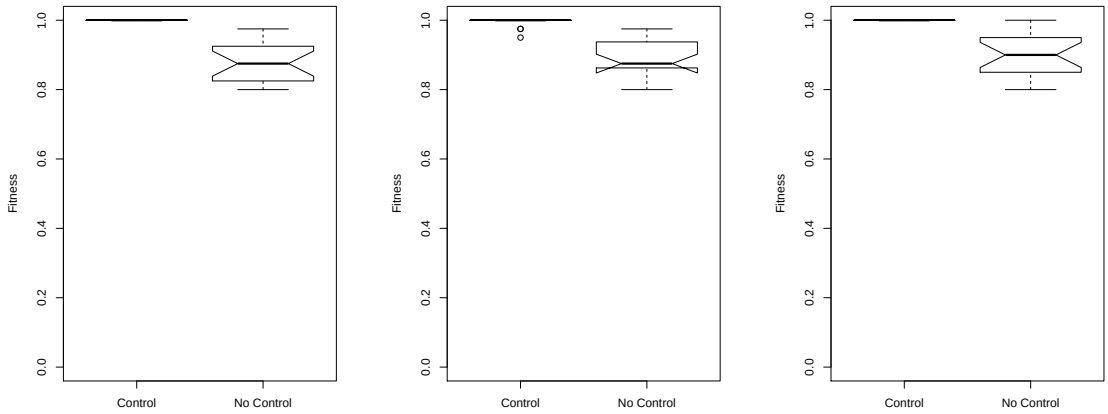


Figure C.6: Fitness distributions for the T cell receptor signalling pathway control problem.

(a) Stable state 1 (b) Stable state 2 (c) Stable state 3



(d) Stable state 4 (e) Stable state 5 (f) Stable state 7

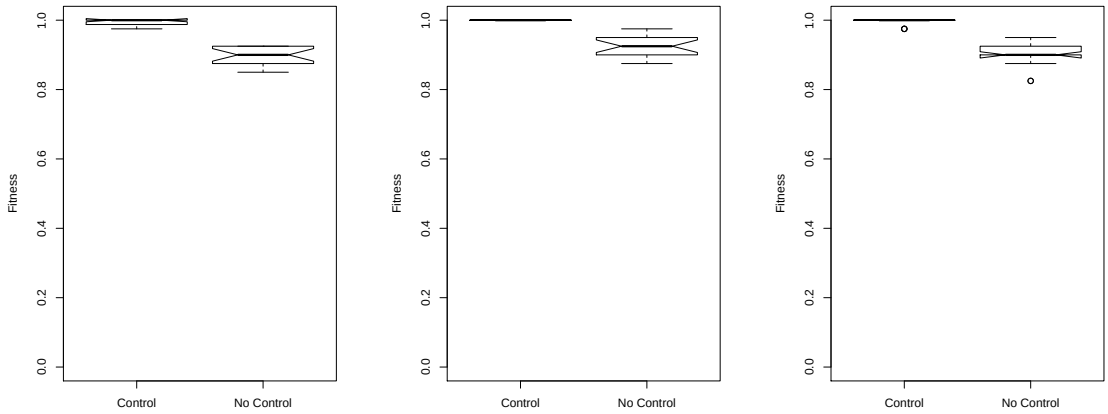


Figure C.7: Fitness distributions for the T helper cell differentiation control problem.

(a) Stable state 1 (b) Stable state 2 (c) Stable state 3

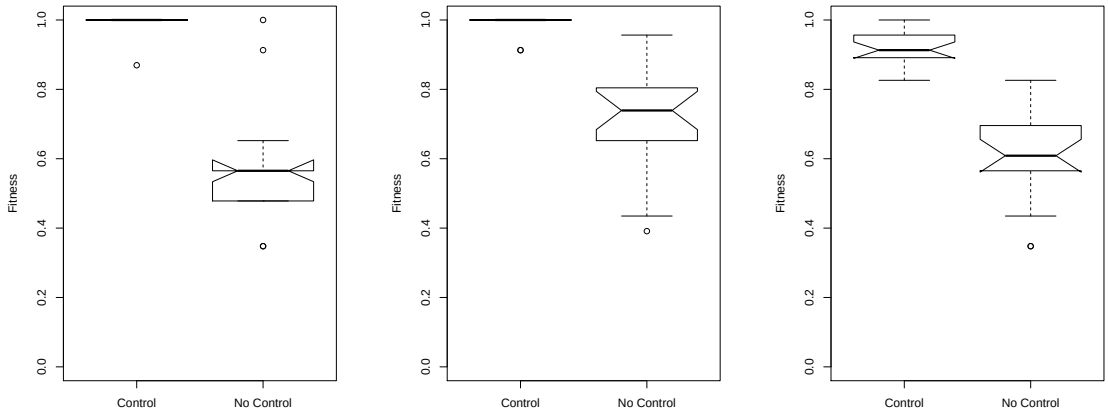
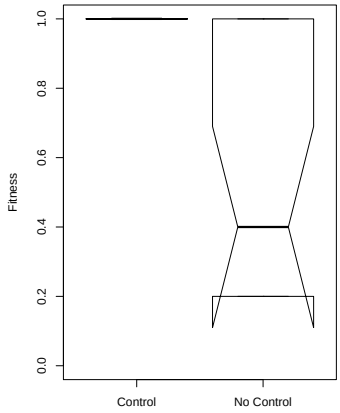
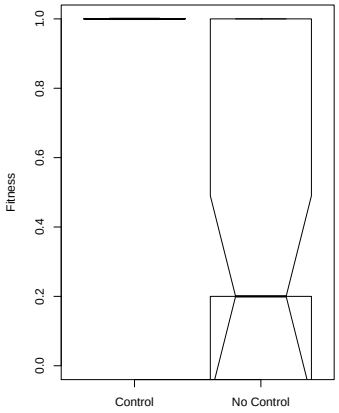


Figure C.9: Fitness distributions for the fission yeast cell yeast control.

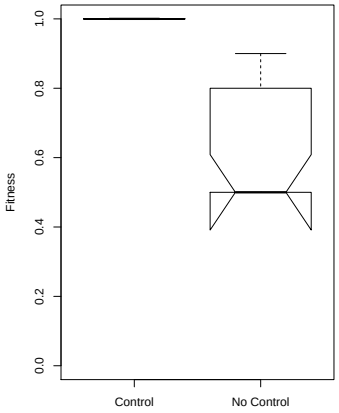
(a) Stable state 1



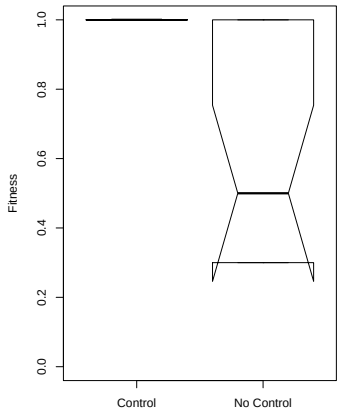
(b) Stable state 2



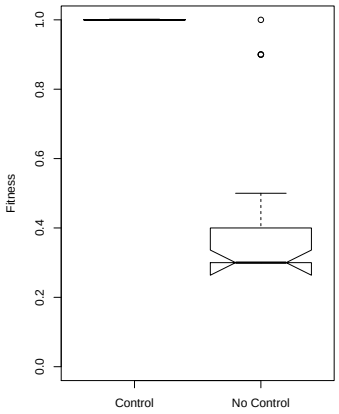
(c) Stable state 3



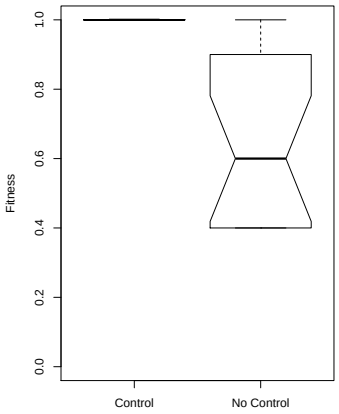
(d) Stable state 4



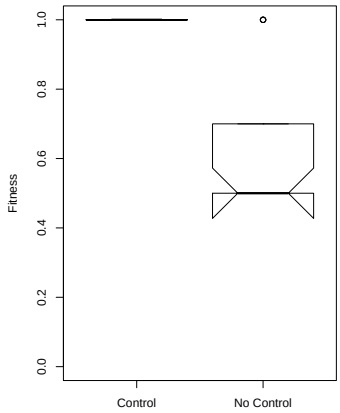
(e) Stable state 5



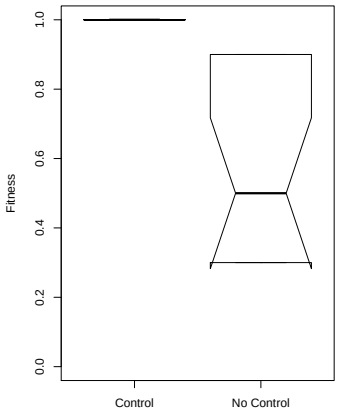
(f) Stable state 6



(g) Stable state 10



(h) Stable state 11



(i) Stable state 12

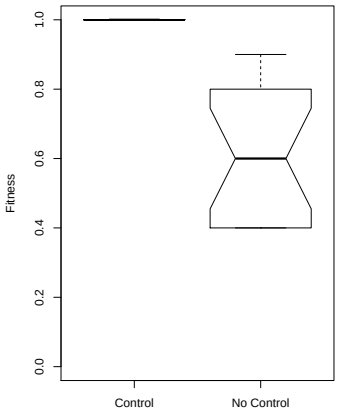
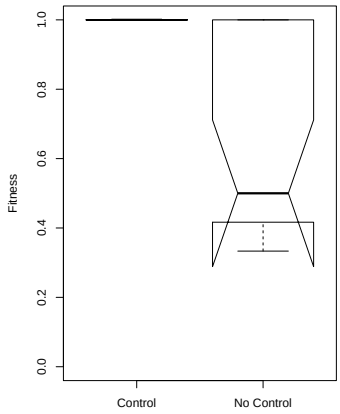
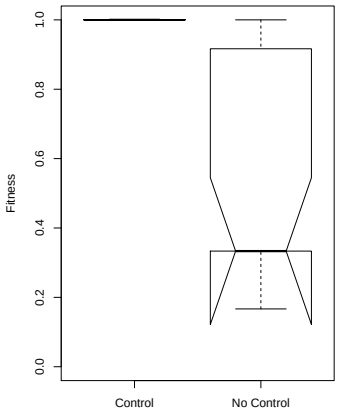


Figure C.10: Fitness distributions for the budding yeast cell cycle control.

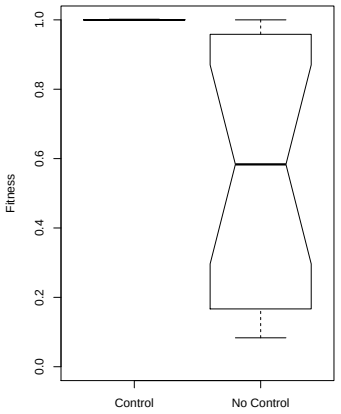
(a) Stable state 1



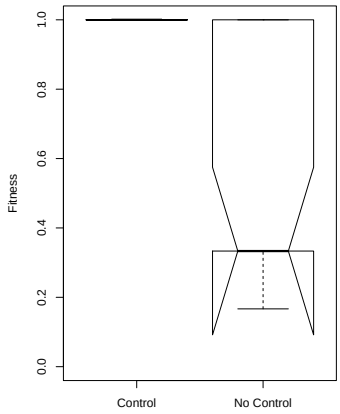
(b) Stable state 2



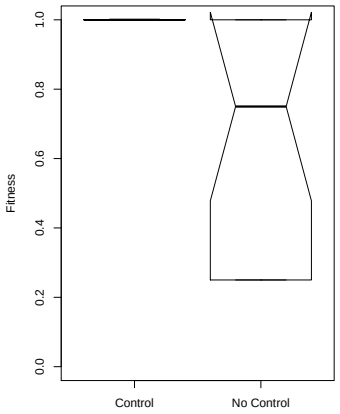
(c) Stable state 3



(d) Stable state 4



(e) Stable state 5



(f) Stable state 6

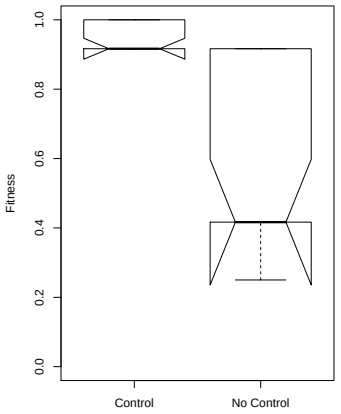
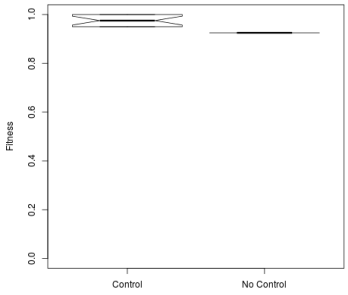
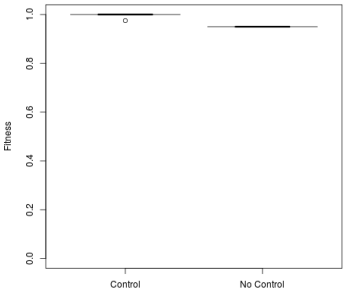


Figure C.11: Fitness distributions for the T cell receptor signalling pathway control problem. Moving for attractor to other attractors

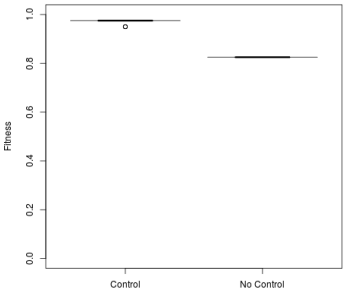
(a) Stable state 2



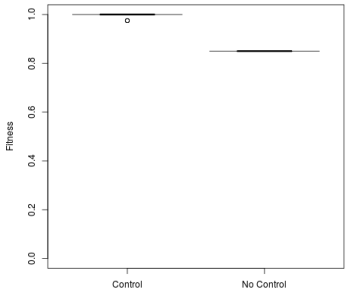
(b) Stable state 3



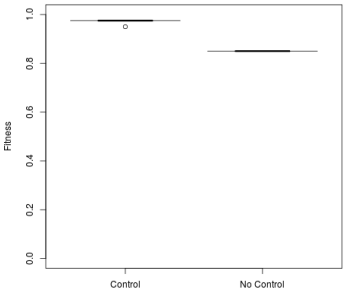
(c) Stable state 4



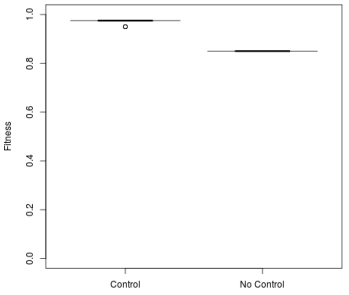
(d) Stable state 5



(e) Stable state 6



(f) Stable state 7



(g) Stable state 8

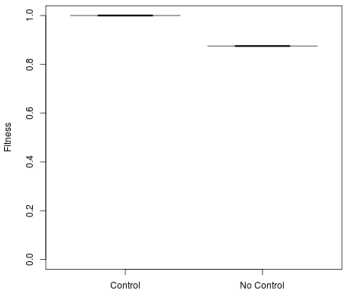
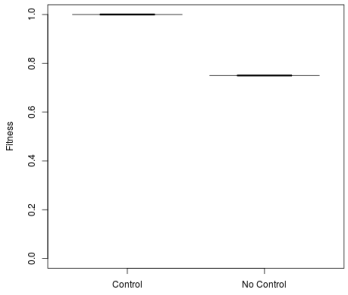
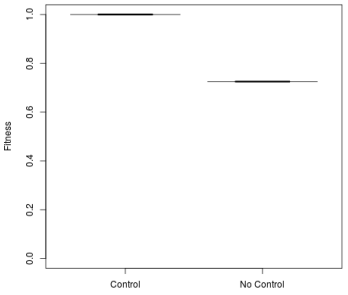


Figure C.12: Fitness distributions for the T cell receptor signalling pathway control problem. Moving for attractor 2 to other attractors

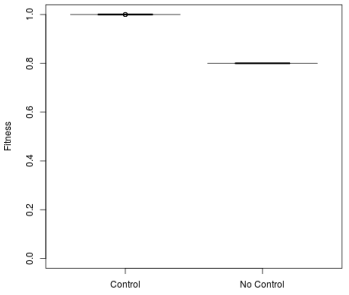
(a) Stable state 1



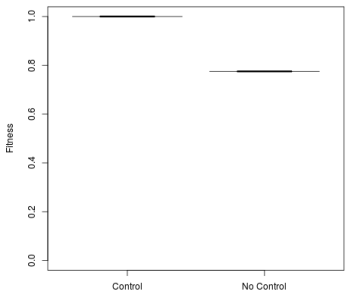
(b) Stable state 3



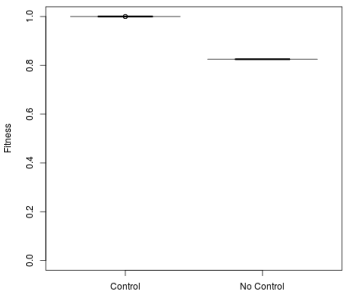
(c) Stable state 4



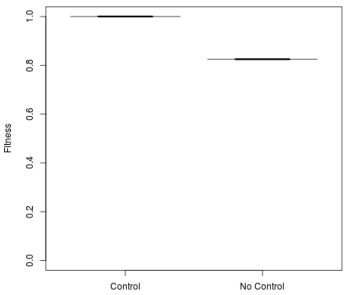
(d) Stable state 5



(e) Stable state 6



(f) Stable state 7



(g) Stable state 8

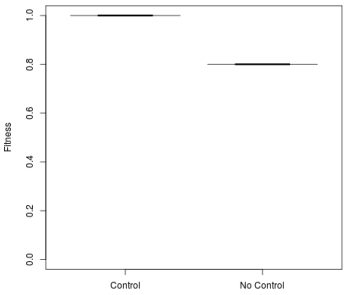
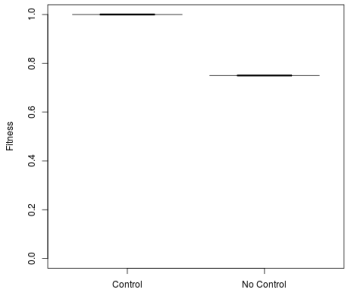
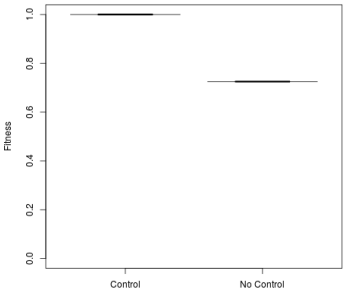


Figure C.13: Fitness distributions for the T cell receptor signalling pathway control problem. Moving for attractor 2 to other attractors

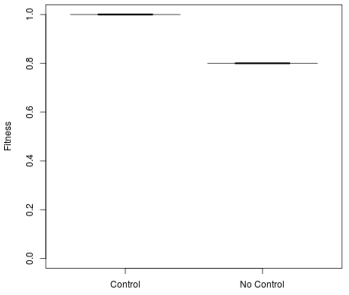
(a) Stable state 1



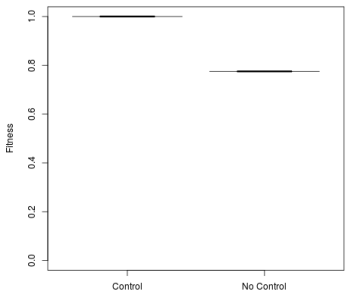
(b) Stable state 2



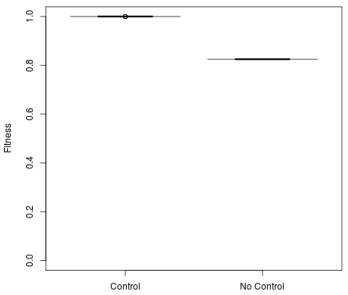
(c) Stable state 4



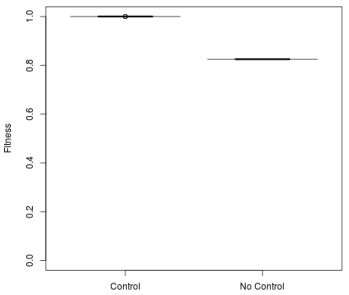
(d) Stable state 5



(e) Stable state 6



(f) Stable state 7



(g) Stable state 8

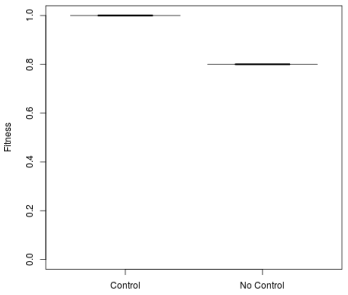
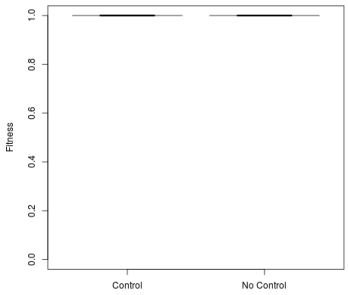
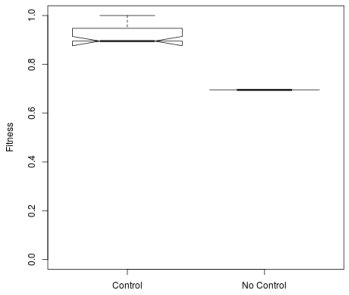


Figure C.14: Fitness distributions for the fission yeast cell yeast control. Moving for attractor 1 to other attractors

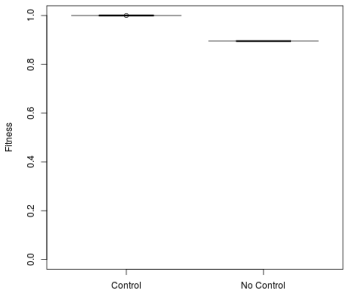
(a) Stable state 2



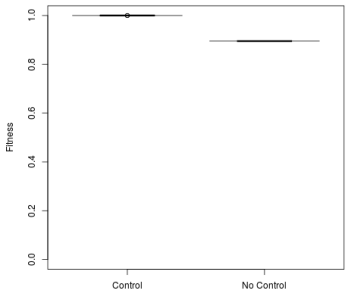
(b) Stable state 3



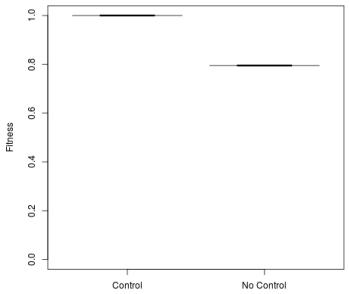
(c) Stable state 4



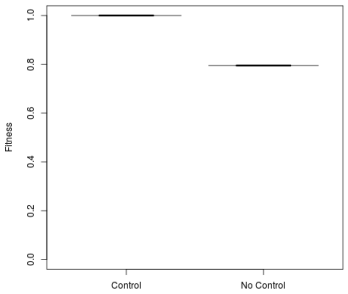
(d) Stable state 5



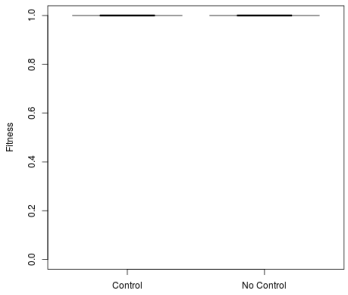
(e) Stable state 6



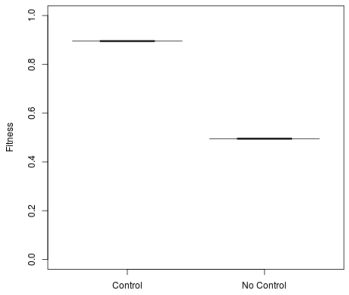
(f) Stable state 7



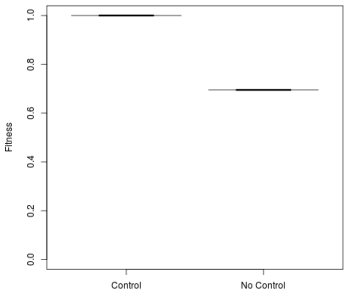
(g) Stable state 8



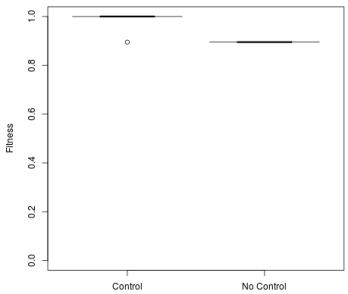
(h) Stable state 9



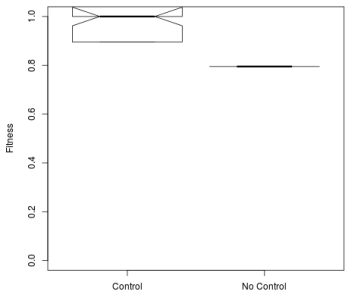
(i) Stable state 10



(j) Stable state 11



(k) Stable state 12



(l) Stable state 13

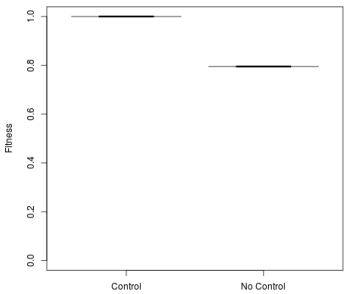
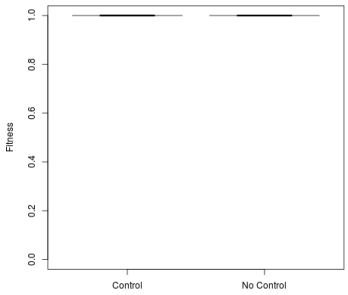
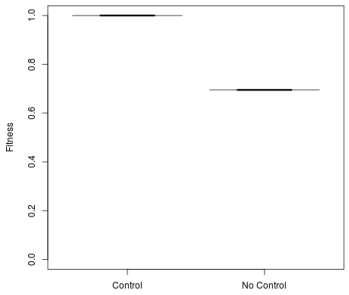


Figure C.15: Fitness distributions for the fission yeast cell yeast control. Moving for attractor 2 to other attractors

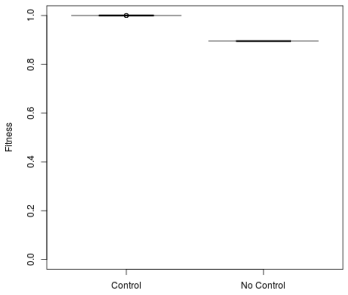
(a) Stable state 1



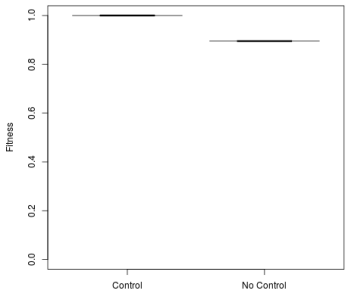
(b) Stable state 3



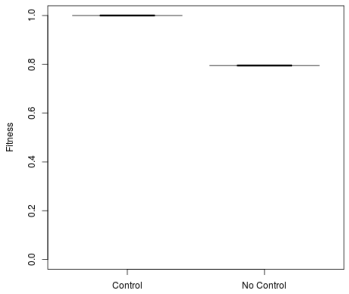
(c) Stable state 4



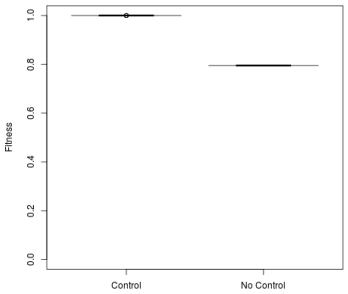
(d) Stable state 5



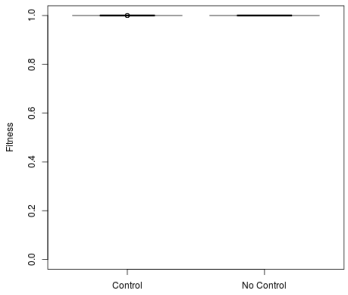
(e) Stable state 6



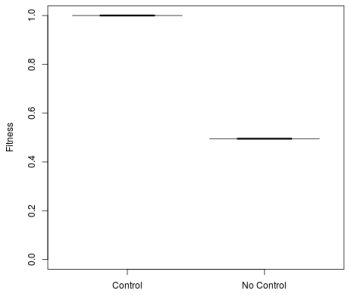
(f) Stable state 7



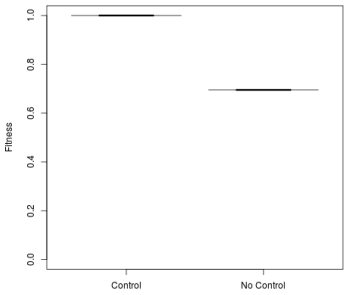
(g) Stable state 8



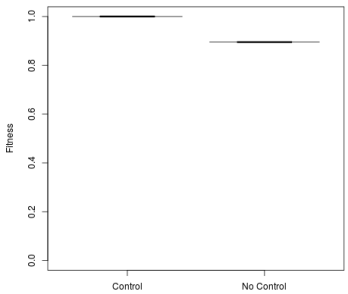
(h) Stable state 9



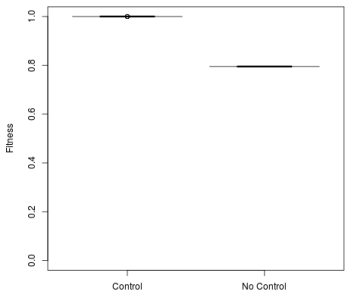
(i) Stable state 10



(j) Stable state 11



(k) Stable state 12



(l) Stable state 13

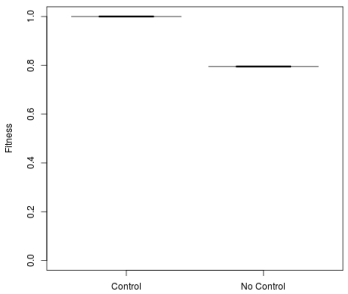
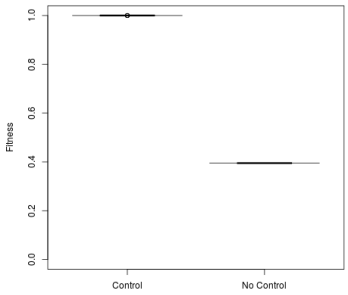
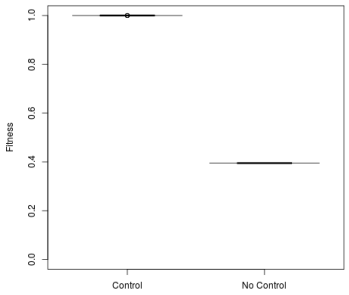


Figure C.16: Fitness distributions for the fission yeast cell yeast control. Moving for attractor 3 to other attractors

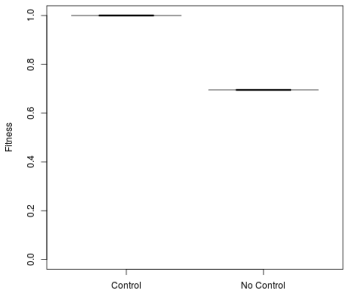
(a) Stable state 1



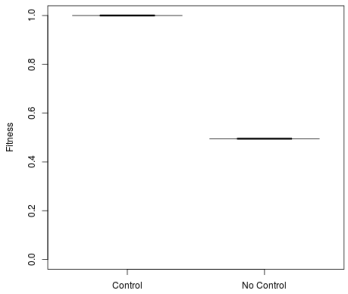
(b) Stable state 2



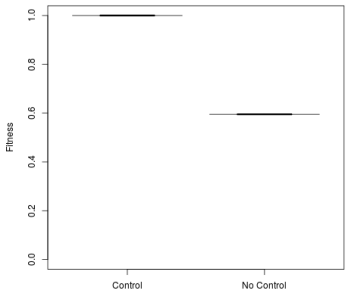
(c) Stable state 4



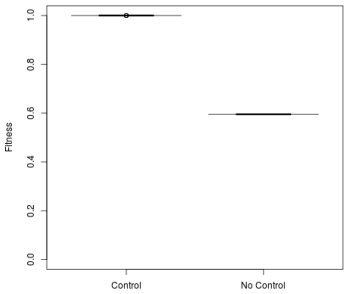
(d) Stable state 5



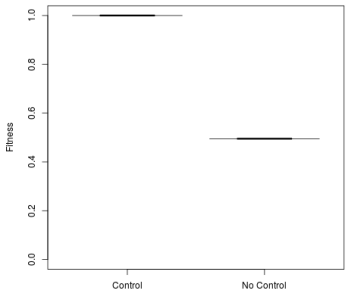
(e) Stable state 6



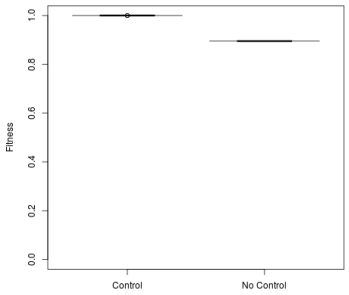
(f) Stable state 7



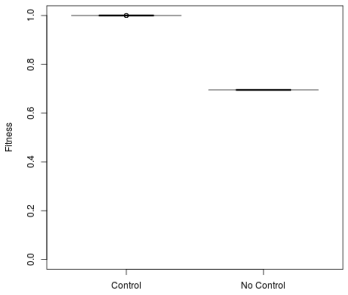
(g) Stable state 8



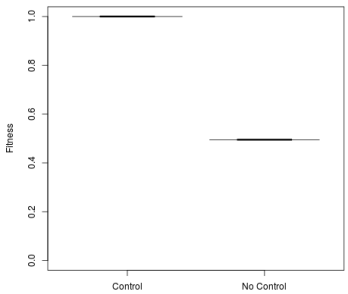
(h) Stable state 9



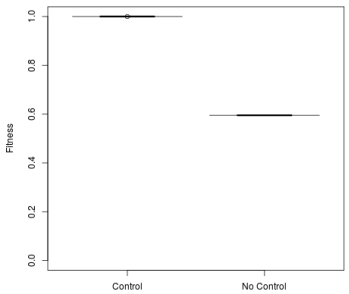
(i) Stable state 10



(j) Stable state 11



(k) Stable state 12



(l) Stable state 13

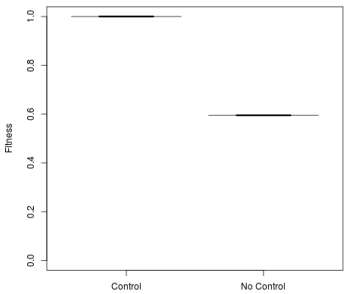
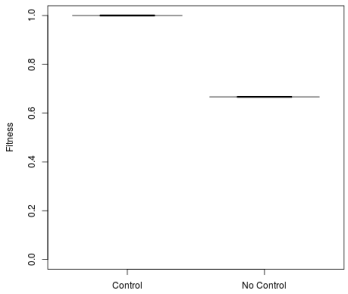
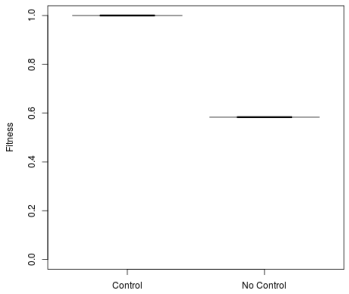


Figure C.17: Fitness distributions for the budding yeast cell yeast control. Moving for attractor 1 to other attractors

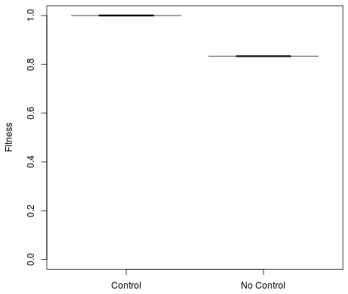
(a) Stable state 2



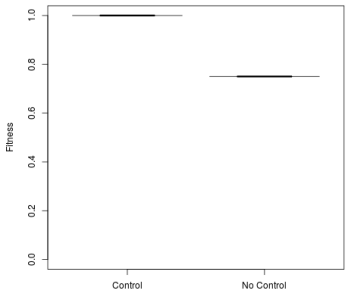
(b) Stable state 3



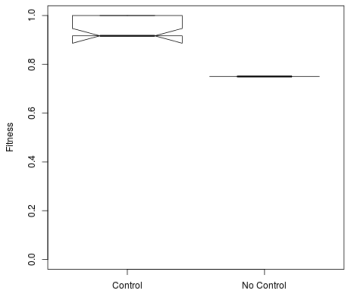
(c) Stable state 4



(d) Stable state 5



(e) Stable state 6



(f) Stable state 7

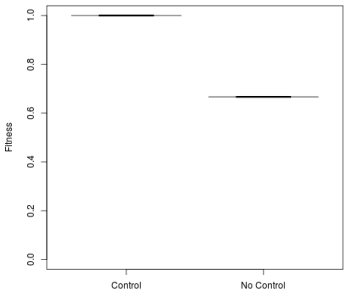
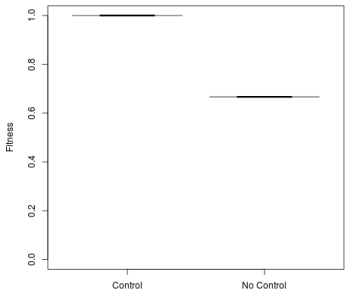
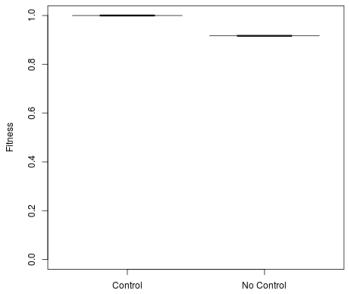


Figure C.18: Fitness distributions for the budding yeast cell yeast control. Moving for attractor 2 to other attractors

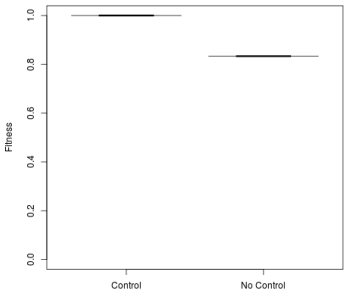
(a) Stable state 1



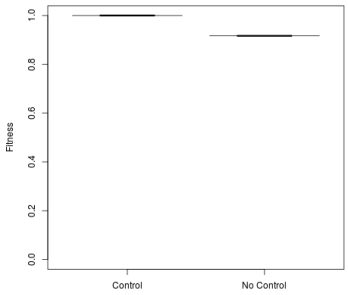
(b) Stable state 3



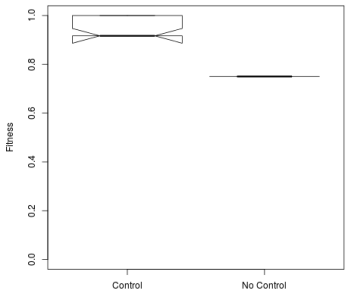
(c) Stable state 4



(d) Stable state 5



(e) Stable state 6



(f) Stable state 7

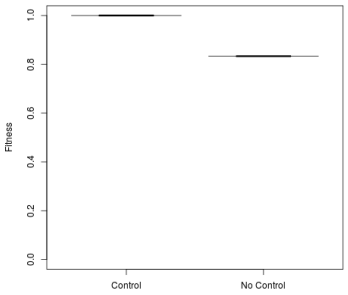
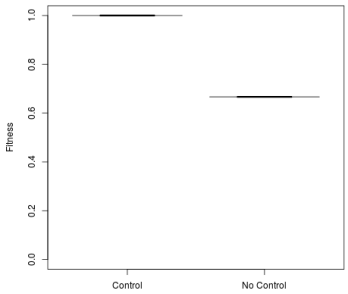
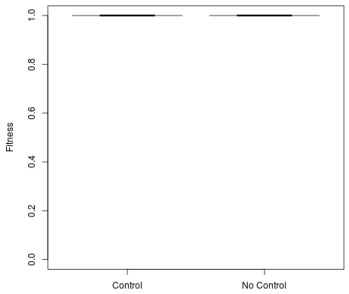


Figure C.19: Fitness distributions for the budding yeast cell yeast control. Moving for attractor 3 to other attractors

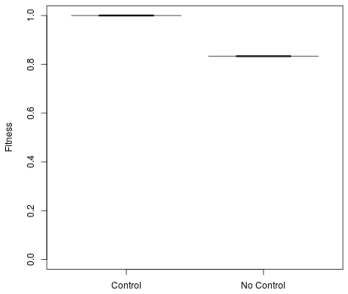
(a) Stable state 1



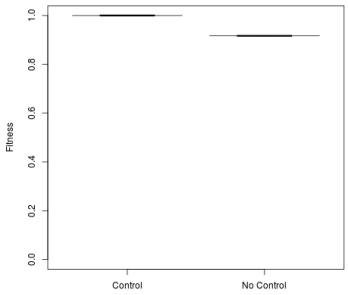
(b) Stable state 2



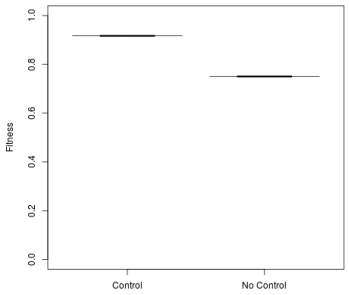
(c) Stable state 4



(d) Stable state 5



(e) Stable state 6



(f) Stable state 7

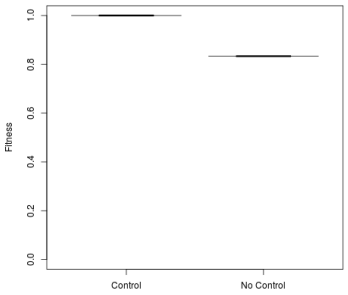
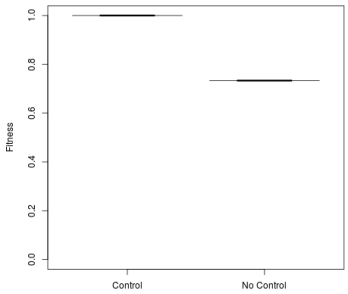
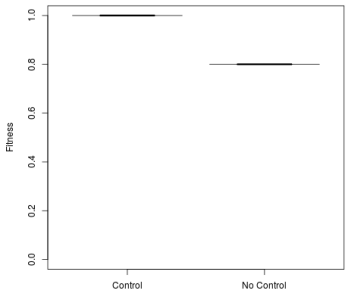


Figure C.20: Fitness distributions for the flower morphogenesis in Arabidobis thaliana control. Moving for attractor 1 to other attractors

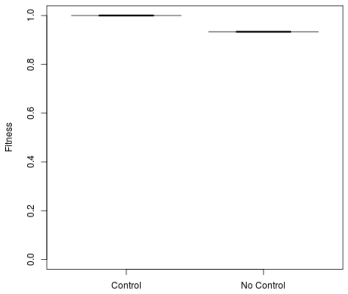
(a) Stable state 2



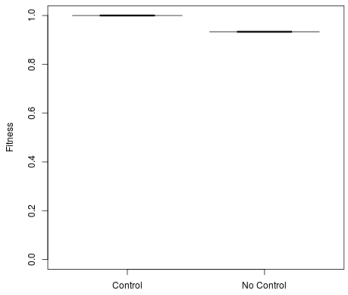
(b) Stable state 3



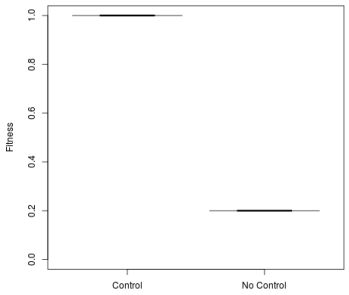
(c) Stable state 4



(d) Stable state 5



(e) Stable state 6



(f) Stable state 7

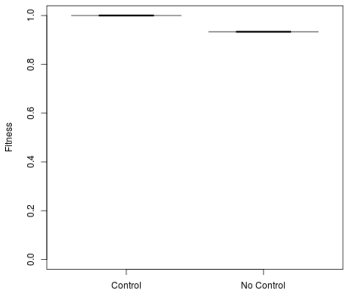
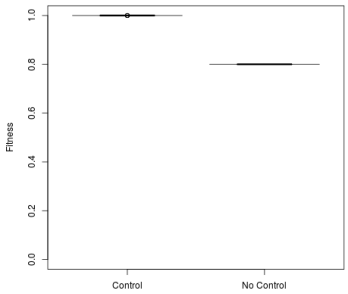
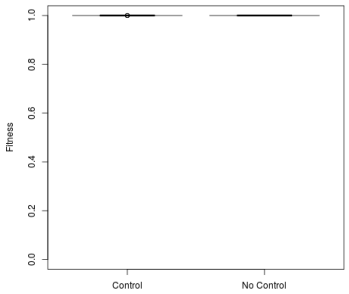


Figure C.21: Fitness distributions for the flower morphogenesis in Arabidobis thaliana control. Moving for attractor 2 to other attractors

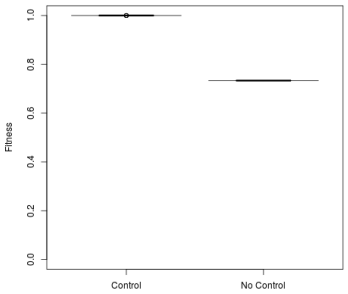
(a) Stable state 1



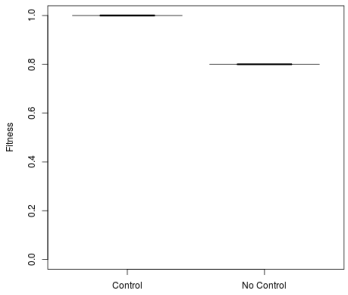
(b) Stable state 3



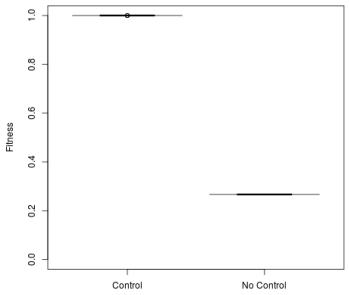
(c) Stable state 4



(d) Stable state 5



(e) Stable state 6



(f) Stable state 7

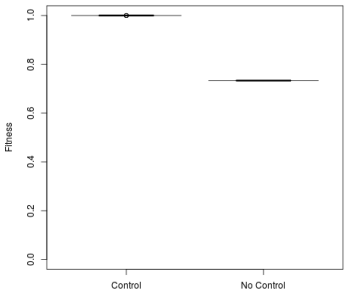
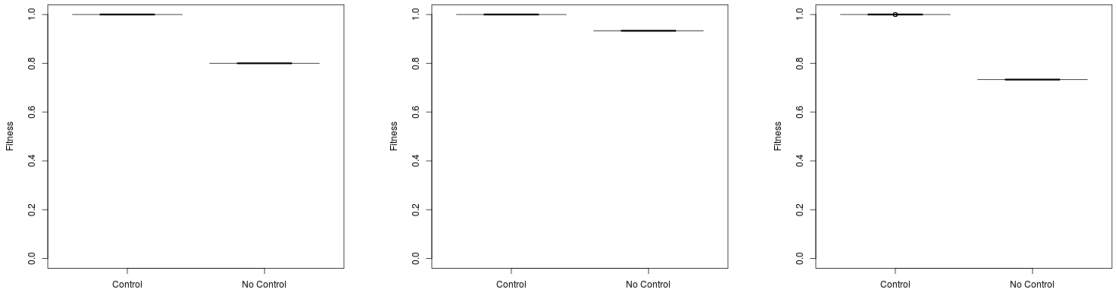


Figure C.22: Fitness distributions for the flower morphogenesis in Arabidobis thaliana control. Moving for attractor 1 to other attractors

(a) Stable state 2 (b) Stable state 3 (c) Stable state 4



(d) Stable state 5 (e) Stable state 6 (f) Stable state 7

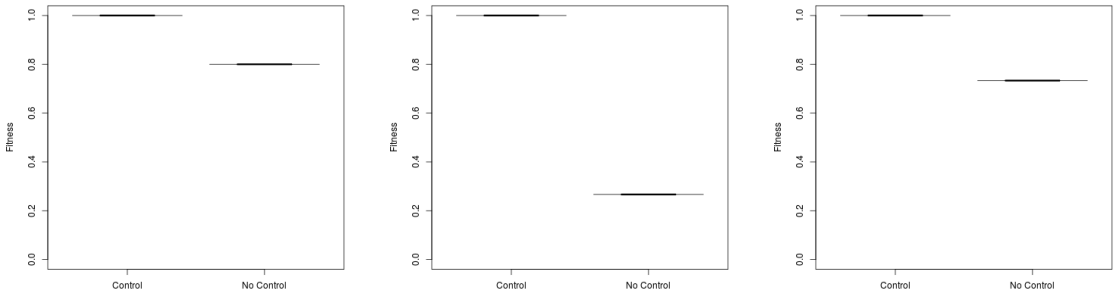


Figure C.23: Fitness distributions for the T-helper cell differentiation control. Moving for attractor 1 to other attractors

(a) Stable state 2 (b) Stable state 3

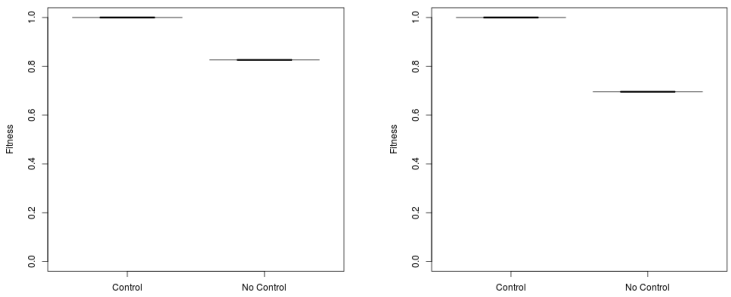


Figure C.24: Fitness distributions for the T-helper cell differentiation control. Moving for attractor 2 to other attractors

(a) Stable state 1 (b) Stable state 3

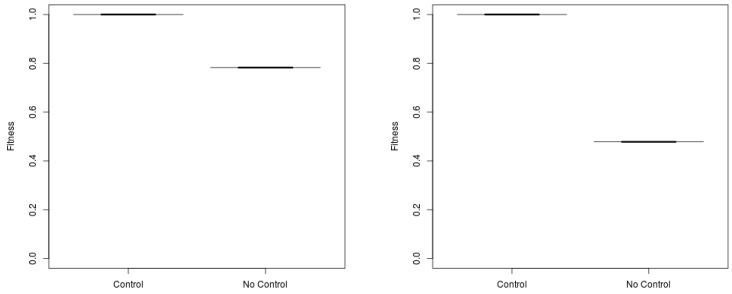
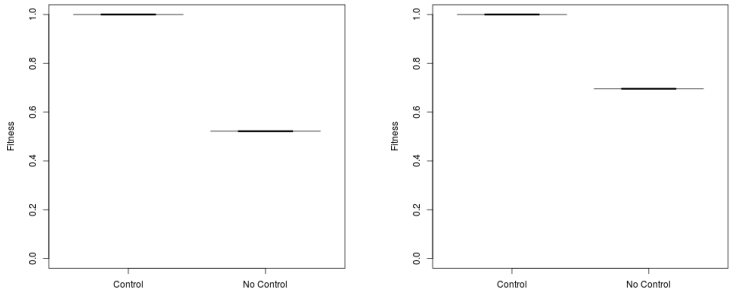


Figure C.25: Fitness distributions for the T-helper cell differentiation control. Moving for attractor 3 to other attractors

(a) Stable state 1 (b) Stable state 2



References

Abul, Osman; Alhaj, R, and Polat, Faruk. Markov decision processes based optimal control policies for probabilistic Boolean networks. In *Bioinformatics and Bioengineering, 2004. BIBE 2004. Proceedings. Fourth IEEE Symposium on*, pages 337–344. IEEE, 2004.

Ahnert, Sebastian E; Fink, Thomas MA, and Zinovyev, Andrei. How much non-coding DNA do eukaryotes require? *J. Theor. Biol.*, 252(4):587–592, 2008.

Akutsu, Tatsuya. Mathematical models and computational methods for inference of genetic networks. *Evolutionary Computation in Gene Regulatory Network Research*, pages 30–48.

Akutsu, Tatsuya; Hayashida, Morihiro; Ching, Wai-Ki, and Ng, Michael K. Control of Boolean networks: hardness results and algorithms for tree structured networks. *J. Theor. Biol.*, 244(4):670–679, 2007.

Albergante, Luca; Timmis, Jon; Beattie, Lynette, and Kaye, Paul M. A Petri net model of granulomatous inflammation: implications for IL-10 mediated control of Leishmania donovani infection. *PLoS Computational Biology*, 9(11):e1003334, 2013.

Albert, Réka and Barabási, Albert-László. Dynamics of complex systems: Scaling laws for the period of Boolean networks. *Phys. Rev. Lett.*, 84(24):5660, 2000.

Albert, Réka and Othmer, Hans G. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *J. Theor. Biol.*, 223(1):1–18, 2003.

Albert, Reka and Thakar, Juilee. Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 6(5): 353–369, 2014.

Alberts, Bruce; Johnson, Alexander; Lewis, Julian; Raff, Martin; Roberts, Keith, and Walter, Peter. *Molecular Biology of the Cell*. Garland Science, 2002.

Aldana, Maximino. Boolean dynamics of networks with scale-free topology. *Physica D: Nonlinear Phenomena*, 185(1):45–66, 2003.

Aldana, Maximino; Balleza, Enrique; Kauffman, Stuart, and Resendiz, Osbaldo. Robustness and evolvability in genetic regulatory networks. *J. Theor. Biol.*, 245 (3):433–448, 2007.

Alvarez-Buylla, Elena R; Chaos, Álvaro; Aldana, Maximino; Benítez, Mariana; Cortes-Poza, Yuriria; Espinosa-Soto, Carlos; Hartasánchez, Diego A; Lotto, R Beau; Malkin, David; Santos, Gerardo J Escalera, and others, . Floral morphogenesis: stochastic explorations of a gene network epigenetic landscape. *Plos One*, 3(11): e3626, 2008.

Amaral, Luís AN; Díaz-Guilera, Albert; Moreira, Andre A; Goldberger, Ary L, and Lipsitz, Lewis A. Emergence of complex dynamics in a simple model of signaling networks. *PNAS of the United States of America*, 101(44):15551–15555, 2004.

Ashbrook, James B. *Brain, Culture & the Human Spirit: Essays from an Emergent Evolutionary Perspective*. University Press of America, 1993.

Azuma, Shun-ichi and Imura, Jun-ichi. Synthesis of optimal controllers for piecewise affine systems with sampled-data switching. *Automatica*, 42(5):697–710, 2006.

Bäck, Thomas. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996.

Bäck, Thomas; Fogel, David B, and Michalewicz, Zbigniew. *Handbook of evolutionary computation*. CRC Press, 1997.

Bäck, Thomas; Fogel, David B, and Michalewicz, Zbigniew. *Evolutionary Computation 1: Basic Algorithms and Operators*. CRC Press, 2000.

Banzhaf, Wolfgang. On the dynamics of an artificial regulatory network. In *European Conference on Artificial Life*, pages 217–227. Springer, 2003.

Banzhaf, Wolfgang. On evolutionary design, embodiment, and artificial regulatory networks. In *Embodied Artificial Intelligence*, pages 284–292. 2004.

Barabási, Albert-László and Bonabeau, Eric. Scale-free networks. *Scientific American*, 288(5):50–59, 2003.

Bentley, Peter J. Evolving fractal gene regulatory networks for robot control. In *European Conference on Artificial Life*, pages 753–762. Springer, 2003.

Bentley, Peter J. Controlling robots with fractal gene regulatory networks. In *Recent Developments in Biologically Inspired Computing*, pages 320–339. IGI Global, 2004.

Bentley, Peter J. Evolving fractal gene regulatory networks for graceful degradation of software. In *Self-star Properties in Complex Information Systems*, pages 21–35. Springer, 2005.

Beyer, Hans-Georg and Schwefel, Hans-Paul. Evolution strategies–a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

Bhalla, Upinder S and Iyengar, Ravi. Emergent properties of networks of biological signaling pathways. *Science*, 283(5400):381–387, 1999.

Bilke, Sven and Sjunnesson, Fredrik. Stability of the Kauffman model. *Physical Review E*, 65(1):016129, 2001.

Blount, Zachary D; Borland, Christina Z, and Lenski, Richard E. Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. *PNAS*, 105(23):7899–7906, 2008.

Bornholdt, Stefan and Rohlf, Thimo. Topological evolution of dynamical networks: global criticality from local dynamics. *Phys. Rev. Lett.*, 84(26):6114, 2000.

Bornholdt, Stefan and Sneppen, Kim. Neutral mutations and punctuated equilibrium in evolving genetic networks. *Phys. Rev. Lett.*, 81(1):236, 1998.

Brameier, Markus F and Banzhaf, Wolfgang. *Linear Genetic Programming*. Springer Science & Business Media, 2007.

Bray, Dennis. Protein molecules as computational elements in living cells. *Nature*, 376(6538):307, 1995.

Bull, Larry. A simple computational cell: coupling Boolean gene and protein networks. *Artificial Life*, 18(2):223–236, 2012.

Bull, Larry and Preen, Richard. On dynamical genetic programming: Random boolean networks in learning classifier systems. In *Genetic Programming, Proceedings of the 12th European Conference on Genetic Programming, EuroGP 2009, Leonardo Vanneschi, Steven Gustafson, Alberto Moraglio, Ivanoe De Falco and Marc Ebner, 5481, LNCS*, pages 37–48. 2009.

Burks, Arthur W. Von Neumann's self-reproducing automata (No. 08226-11-T). Technical report, DTIC Document, 1969.

Chaos, Alvaro; Aldana, Max; Espinosa-Soto, Carlos; de León, Berenice García Ponce; Arroyo, Adriana Garay, and Alvarez-Buylla, Elena R. From genes to flower patterns and evolution: dynamic models of gene regulatory networks. *Journal of Plant Growth Regulation*, 25(4):278–289, 2006.

Chen, Hao; Li, Xingde, and Sun, Jitao. Stabilization, controllability and optimal control of Boolean networks with impulsive effects and state constraints. *IEEE Transactions on Automatic Control*, 60(3):806–811, 2015.

Cheng, Daizhan and Qi, Hongsheng. Controllability and observability of Boolean control networks. *Automatica*, 45(7):1659–1667, 2009.

Cheng, Daizhan and Zhao, Yin. Identification of Boolean control networks. *Automatica*, 47(4):702–710, 2011.

Cheng, Daizhan; Qi, Hongsheng, and Li, Zhiqiang. *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*. Springer, 2010.

Clark, Andrew; Lee, Phillip; Alomair, Basel; Bushnell, Linda, and Poovendran, Radha. Combinatorial algorithms for control of biological regulatory networks. *arXiv preprint arXiv:1701.05273*, 2017.

Clauset, Aaron; Shalizi, Cosma Rohilla, and Newman, Mark EJ. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

Coello, Carlos A Coello; Lamont, Gary B; Van Veldhuizen, David A, and others, . *Evolutionary Algorithms for Solving Multi-Objective Problems*, volume 5. Springer, 2007.

Cohen, Reuven; Havlin, Shlomo, and Ben-Avraham, Daniel. Structural properties of scale free networks. *Handbook of Graphs and Networks*, 2003.

Colegrave, N and Collins, Sinéd. Experimental evolution: experimental evolution and evolvability. *Heredity*, 100(5):464–470, 2008.

Conrad, Michael. Bootstrapping on the adaptive landscape. *BioSystems*, 11(2-3): 167–182, 1979.

Conrad, Michael. The geometry of evolution. *BioSystems*, 24(1):61–81, 1990.

Cooper, Tim F; Rozen, Daniel E, and Lenski, Richard E. Parallel changes in gene expression after 20,000 generations of evolution in Escherichia coli. *PNAS*, 100(3): 1072–1077, 2003.

Cury, José ER and Baldissera, Fabio L. Systems biology, synthetic biology and control theory: a promising golden braid. *Annual Reviews in Control*, 37(1):57–67, 2013.

Cussat-Blanc, Sylvain and Pollack, Jordan. A cell-based developmental model to generate robot morphologies. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 537–544. ACM, 2012.

Cussat-Blanc, Sylvain; Sanchez, Stéphane, and Duthen, Yves. Controlling cooperative and conflicting continuous actions with a gene regulatory network. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 187–194. IEEE, 2012.

Dallidis, Stylianos E and Karafyllidis, Ioannis G. Boolean network model of the *Pseudomonas aeruginosa* quorum sensing circuits. *IEEE Transactions on Nanobioscience*, 13(3):343–349, 2014.

Darabos, Christian; Giacobini, Mario, and Tomassini, Marco. Semi-synchronous activation in scale-free boolean networks. In *Advances in Artificial Life*, pages 976–985. 2007.

Darwin, Charles. On the origin of species by means of natural selection. *Murray, London*, 1859.

Davidich, Maria I and Bornholdt, Stefan. Boolean network model predicts cell cycle sequence of fission yeast. *PloS One*, 3(2):e1672, 2008.

Dawkins, Richard. The selfish gene. *New York, New York, USA*, 1976.

Deb, Kalyanmoy. *Multi-Objective Optimization using Evolutionary Algorithms*, volume 16. John Wiley & Sons, 2001.

Deb, Kalyanmoy; Agrawal, Samir; Pratap, Amrit, and Meyarivan, Tanaka. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel problem solving from nature PPSN VI*, pages 849–858. Springer, 2000.

Deb, Kalyanmoy; Pratap, Amrit; Agarwal, Sameer, and Meyarivan, TAMT. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput*, 6(2):182–197, 2002.

Drossel, Barbara. Random boolean networks. *Reviews of Nonlinear Dynamics and Complexity*, 1:69–110, 2008.

Drossel, Barbara; Mihaljev, Tamara, and Greil, Florian. Number and length of attractors in a critical Kauffman model with connectivity one. *Phys. Rev. Lett.*, 94(8):088701, 2005.

Dubrova, Elena; Teslenko, Maxim, and Tenhunen, Hannu. A computational scheme based on random Boolean networks. In *Transactions on Computational Systems Biology X*, pages 41–58. Springer, Berlin Heidelberg, 2008.

Dworkin, Martin. *The Prokaryotes: vol. 4: Bacteria: Firmicutes, Cyanobacteria*. Springer Science & Business Media, 2006.

Eiben, A. E. and Smith, J. E. *Introduction to Evolutionary Computing*. Springer, 2008.

Fauré, Adrien; Naldi, Aurélien; Chaouiya, Claudine, and Thieffry, Denis. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–e131, 2006.

Félix, Marie-Anne and Wagner, Andreas. Robustness and evolution: concepts, insights and challenges from a developmental model system. *Heredity*, 100(2): 132–140, 2008.

Fletcher, Hugh and Hickey, Ivor. *BIOS Instant Notes in Genetics*. Garland Science, 2012.

Fogel, Lawrence J; Owens, AJ, and Walsh, MJ. On the evolution of artificial intelligence(Artificial intelligence generated by natural evolution process). In *National Symposium on Human Factors in Electronics,5th, San Diego, Calif*, pages 63–76, 1964.

Fornasini, Ettore and Valcher, Maria Elena. Optimal control of Boolean control networks. *IEEE Transactions on Automatic Control*, 59(5):1258–1270, 2014.

Forst, Christian V and Schulten, Klaus. Evolution of metabolisms: a new method for the comparison of metabolic pathways using genomics information. *J. Comput. Biol.*, 6(3-4):343–360, 1999.

Fuente, Luis A; Lones, Michael A; Turner, Alexander P; Caves, Leo S; Stepney, Susan, and Tyrrell, Andy M. Adaptive robotic gait control using coupled artificial signalling networks, hopf oscillators and inverse kinematics. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1435–1442, 2013.

Fumiã, Herman F and Martins, Marcelo L. Boolean network model for cancer pathways: predicting carcinogenesis and targeted therapy outcomes. *PloS One*, 8 (7):e69008, 2013.

Gao, Bo; Peng, Haipeng; Zhao, Dawei; Zhang, Wenguang, and Yang, Yixian. Attractor transformation by impulsive control in boolean control network. *Mathematical Problems in Engineering*, 2013, 2013.

Garg, Abhishek; Xenarios, Ioannis; Mendoza, Luis, and DeMicheli, Giovanni. An efficient method for dynamic analysis of gene regulatory networks and in silico gene perturbation experiments. *Lecture Notes in Computer Science*, 4453:62–76, 2007.

Gates, Alexander J and Rocha, Luis M. Control of complex networks requires both structure and dynamics. *Scientific Reports*, 6:24456, 2016.

Gershenson, Carlos. Introduction to random Boolean networks. *In Bedau, M., P. Husbands, T. Hutton, S. Kumar, and H. Suzuki (eds.) Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, pages 160–173, 2004.

Gershenson, Carlos; Kauffman, Stuart A, and Shmulevich, Ilya. The role of redundancy in the robustness of random Boolean networks. *arXiv preprint nlin/0511018*, 2005.

Giacomantonio, Clare E and Goodhill, Geoffrey J. A Boolean model of the gene regulatory network underlying Mammalian cortical area development. *PLoS Computational Biology*, 6(9):e1000936, 2010.

Goldberg, David E and Deb, Kalyanmoy. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, 1:69–93, 1991.

Goudarzi, Alireza; Teuscher, Christof; Gulbahce, Natali, and Rohlf, Thimo. Emergent criticality through adaptive information processing in Boolean networks. *Phys. Rev. Lett.*, 108(12):128702, 2012.

Greaves, Richard B; Read, Mark; Timmis, Jon; Andrews, Paul S; Butler, James A; Gerckens, Bjorn-Ole, and Kumar, Vipin. In silico investigation of novel biological pathways: the role of CD200 in regulation of T cell priming in experimental autoimmune encephalomyelitis. *Biosystems*, 112(2):107–121, 2013.

Harvey, Inman. The microbial genetic algorithm. In *European Conference on Artificial Life*, pages 126–133. Springer, 2009.

Harvey, Inman and Bossomaier, Terry. Time out of joint: attractors in asynchronous random boolean networks. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 67–75. MIT Press, Cambridge, 1997.

Hintze, Arend and Adami, Christoph. Evolution of complex modular biological networks. *PLoS Comput Biol*, 4(2):e23, 2008.

Hoffmann, Alexander; Levchenko, Andre; Scott, Martin L, and Baltimore, David. The IκB-NF-κB signaling module: temporal control and selective gene activation. *Science*, 298(5596):1241–1245, 2002.

Holland, John H. *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* U Michigan Press, 1975.

Holme, Petter and Saramäki, Jari. Temporal networks. *Physics Reports*, 519(3): 97–125, 2012.

Hou, Wenpin; Tamura, Takeyuki; Ching, Wai-Ki, and Akutsu, Tatsuya. Finding and analyzing the minimum set of driver nodes in control of Boolean networks. *Advances in Complex Systems*, 19(03):1650006, 2016.

Huang, Sui and Kauffman, Stuart. How to escape the cancer attractor: rationale

and limitations of multi-target drugs. In *Seminars in cancer biology*, volume 23, pages 270–278. Academic Press, 2013.

Huang, Sui; Eichler, Gabriel; Bar-Yam, Yaneer, and Ingber, Donald E. Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Phys. Rev. Lett.*, 94(12):128701, 2005.

Huang, Sui; Ernberg, Ingemar, and Kauffman, Stuart. Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. In *Seminars in cell & developmental biology*, volume 20, pages 869–876. Elsevier, 2009.

Igamberdiev, Abir U; Bykova, Natalia V, and Kleczkowski, Leszek A. Origins and metabolism of formate in higher plants. *Plant Physiology and Biochemistry*, 37 (7-8):503–513, 1999.

Jacob, François and Monod, Jacques. Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol*, 3(3):318–356, 1961.

Joachimczak, Michal and Wróbel, Borys. Evolving gene regulatory networks for real time control of foraging behaviours. In *ALIFE*, pages 348–358. Springer, 2010.

Karlebach, Guy and Shamir, Ron. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780, 2008.

Karp, Peter D and Mavrovouniotis, Michael L. Representing, analyzing, and synthesizing biochemical pathways. *IEEE Expert*, 9(2):11–21, 1994.

Kauffman, Stuart; Peterson, Carsten; Samuelsson, Björn, and Troein, Carl. Random Boolean network models and the yeast transcriptional network. *PNAS*, 100(25): 14796–14799, 2003.

Kauffman, Stuart A. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22(3):437–467, 1969.

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

Kaushik, Aman Chandra and Sahi, Shakti. Boolean network model for GPR142 against Type 2 diabetes and relative dynamic change ratio analysis using systems and biological circuits approach. *Systems and Synthetic Biology*, 9(1-2):45–54, 2015.

Kholodenko, Boris N. Cell-signalling dynamics in time and space. *Nature reviews Molecular Cell Biology*, 7(3):165–176, 2006.

Kim, Junil; Park, Sang-Min, and Cho, Kwang-Hyun. Discovery of a kernel for controlling biomolecular regulatory networks. *Scientific Reports*, 3:2223, 2013.

Kirschner, Marc and Gerhart, John. Evolvability. *PNAS*, 95(15):8420–8427, 1998.

Kitano, Hiroaki. Computational systems biology. *Nature*, 420(6912):206–210, 2002.

Kitano, Hiroaki. Cancer as a robust system: implications for anticancer therapy. *Nature Reviews Cancer*, 4(3):227–235, 2004.

Kitano, Hiroaki. Towards a theory of biological robustness. *Molecular Systems Biology*, 3(1):137, 2007.

Klamt, Steffen; Saez-Rodriguez, Julio; Lindquist, Jonathan A; Simeoni, Luca, and Gilles, Ernst D. A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, 7(1):56, 2006.

Knowles, Joshua and Corne, David. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, pages 98–105. IEEE, 1999.

Knowles, Joshua D and Corne, David W. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2): 149–172, 2000.

Kobayashi, Kaoru and Hiraishi, Kunihiko. A Petri net-based approach to control of Boolean networks. In *Networking and Computing (ICNC), 2012 Third International Conference on*, pages 399–403. IEEE, 2012a.

Kobayashi, Koichi and Hiraishi, Kunihiko. An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks. *Automatica*, 47(6):1260–1264, 2011.

Kobayashi, Koichi and Hiraishi, Kunihiko. Optimal control of probabilistic Boolean networks using polynomial optimization. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 95(9):1512–1517, 2012b.

Krohn, Jean and Gorse, Denise. Fractal gene regulatory networks for control of nonlinear systems. In *International Conference on Parallel Problem Solving from Nature*, pages 209–218. Springer, 2010.

Kumar, Sanjeev. A developmental genetics-inspired approach to robot control. In *Proceedings of the 7th annual workshop on Genetic and evolutionary computation*, pages 304–309. ACM, 2005.

Kurganov, Boris I; Lyubarev, Arkady E; Sanchez-Ruiz, Jose M, and Shnyrov, Valery L. Analysis of differential scanning calorimetry data for proteins criteria of validity of one-step mechanism of irreversible protein denaturation. *Biophysical Chemistry*, 69(2):125–135, 1997.

Kuyucu, Tuze. *Evolution of Circuits in Hardware and the Evolvability of Artificial Development, PhD.* PhD thesis, University of York, 2010.

Kyozuka, Junko; Harcourt, R; Peacock, WJ, and Dennis, ES. Eucalyptus has functional equivalents of the Arabidopsis AP1 gene. *Plant Molecular Biology*, 35 (5):573–584, 1997.

Lai, Liangxue; Park, Kwang-Wook; Cheong, Hee-Tae; Kühholzer, Birgit; Samuel, Melissa; Bonk, Aaron; Im, Gi-Sun; Rieke, August; Day, Billy N; Murphy, Clifton N, and others, . Transgenic pig expressing the enhanced green fluorescent protein produced by nuclear transfer using colchicine-treated fibroblasts as donor cells. *Mol. Reprod. Dev.*, 62(3):300–306, 2002.

Laschov, Dmitriy and Margaliot, Michael. A maximum principle for single-input

Boolean control networks. *IEEE Transactions on Automatic Control*, 56(4):913–917, 2011.

Lazarus, Christopher and Hu, Huosheng. Using genetic programming to evolve robot behaviours. In *Proceedings of the 3rd British Conference on Autonomous Mobile Robotics & Autonomous Systems*, volume 5, 2001.

Le Novère, Nicolas. Quantitative and logic modelling of molecular and gene networks. *Nature Reviews Genetics*, 16(3):146–158, 2015.

Lenski, Richard E. Bacterial evolution and the cost of antibiotic resistance. *International Microbiology*, 1(4):265–270, 2010.

Li, Fangfei. Pinning control design for the stabilization of Boolean networks. *IEEE Transactions on Neural Networks and Learning Systems*, 27(7):1585–1590, 2016.

Li, Fangfei and Sun, Jitao. Controllability of Boolean control networks with time delays in states. *Automatica*, 47(3):603–607, 2011.

Li, Fangfei and Sun, Jitao. Controllability and optimal control of a temporal Boolean network. *Neural Networks*, 34:10–17, 2012.

Li, Fangting; Long, Tao; Lu, Ying; Ouyang, Qi, and Tang, Chao. The yeast cell-cycle network is robustly designed. *PNAS of the United States of America*, 101(14): 4781–4786, 2004.

Liu, Yang-Yu and Barabási, Albert-László. Control principles of complex systems. *Reviews of Modern Physics*, 88(3):035006, 2016.

Liu, Yang-Yu; Slotine, Jean-Jacques, and Barabási, Albert-László. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

Lodish, Harvey; Berk, Arnold; Darnell, James E; Kaiser, Chris A; Krieger, Monty; Scott, Matthew P; Bretscher, Anthony; Ploegh, Hidde; Matsudaira, Paul, and others, . *Molecular Cell Biology*. Macmillan, 2008.

Lones, Michael; Turner, Alexander P; Caves, Leo S; Stepney, Susan, and Tyrrell, Andy M. Adaptive robotic gait control using coupled artificial signalling networks, hopf oscillators and inverse kinematics. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1435–1442, 2013a.

Lones, Michael A. *Enzyme Genetic Programming: Modelling Biological Evolvability in Genetic Programming.* PhD thesis, The University of York, Heslington, York, YO10 5DD, UK, 2004.

Lones, Michael A. Metaheuristics in nature-inspired algorithms. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2014), Workshop on Metaheuristic Design Patterns (MetaDeeP)*, pages 1419–1422. ACM, 2014. ISBN 9781450328814.

Lones, Michael A. Computing with artificial gene regulatory networks. *H. Iba and N. Noman (eds), Evolutionary Algorithms in Gene Regulatory Network Research, Wiley*, 2016.

Lones, Michael A; Tyrrell, Andy M; Stepney, Susan, and Caves, Leo S. Controlling complex dynamics with artificial biochemical networks. In *Genetic Programming, 13th European Conference, EuroGP2010, Istanbul, Turkey, April 7-9, 2010. Proceedings, Anna Isabel Esparcia-Alcázar and Anikó Ekárt andSara Silva and Stephen Dignum and A. Sima Etaner-Uyar, Lecture Notes in Computer Science, 6021*, pages 159–170. 2010.

Lones, Michael A; Tyrrell, Andy M; Stepney, Susan, and Caves, LSD. Controlling legged robots with coupled artificial biochemical networks. *Advances in Artificial Life, ECAL*, pages 465–472, 2011.

Lones, Michael A; Smith, Stephen L; Tyrrell, Andy M; Alty, Jane E, and Jamieson, DR Stuart. Characterising neurological time series data using biologically motivated networks of coupled discrete maps. *BioSystems*, 112(2):94–101, 2013b.

Lones, Michael A; Turner, Alexander P; Fuente, Luis A; Stepney, Susan; Caves,

Leo SD, and Tyrrell, Andy M. Biochemical connectionism. *Natural Computing*, 12(4):453–472, 2013c.

Lones, Michael A; Fuente, Luis A; Turner, Alexander P; Caves, Leo SD; Stepney, Susan; Smith, Stephen L, and Tyrrell, Andy M. Artificial biochemical networks: evolving dynamical systems to control dynamical systems. *IEEE Trans. Evol. Comput*, 18(2):145–166, 2014.

Marijuán, Pedro C; del Moral, Raquel, and Navarro, Jorge. On eukaryotic intelligence: signaling system's guidance in the evolution of multicellular organization. *Biosystems*, 114(1):8–24, 2013.

Maslov, Sergei; Sneppen, Kim, and Alon, Uri. Correlation profiles and motifs in complex networks. *Handbook of Graphs and Networks*, pages 168–198, 2003.

Mattick, John S. Non-coding RNAs: the architects of eukaryotic complexity. *EMBO Reports*, 2(11):986–991, 2001.

Mendel, Gregor. Versuche uber pflanzenhybriden. *Verh. natf. ver. Brunn. abh.*, 3: S–3, 1865.

Mendoza, Luis and Xenarios, Ioannis. A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theoretical Biology and Medical Modelling*, 3(1):13, 2006.

Mendoza, Luis; Thieffry, Denis, and Alvarez-Buylla, Elena R. Genetic control of flower morphogenesis in *Arabidopsis thaliana*: a logical analysis. *Bioinformatics (Oxford, England)*, 15(7):593–606, 1999.

Mesot, Bertrand and Teuscher, Christof. Deducing local rules for solving global tasks with random Boolean networks. *Physica D: Nonlinear Phenomena*, 211(1):88–106, 2005.

Miller, Jeffrey H. *The Operon*, volume 7. Cold Spring Harbor Laboratory Pr, 1980.

Miller, Julian F and Thomson, Peter. Cartesian genetic programming. In *European Conference on Genetic Programming*, pages 121–132. Springer, 2000.

Mirsky, Henry P; Liu, Andrew C; Welsh, David K; Kay, Steve A, and Doyle, Francis J. A model of the cell-autonomous mammalian circadian clock. *PNAS*, 106(27):11107–11112, 2009.

Mitchell, Melanie. *An Introduction to Genetic Algorithms*. MIT Press, 1998.

Motter, Adilson E. Networkcontrology. *Chaos*, 25(9):097621, 2015.

Murata, Tadahiko and Ishibuchi, Hisao. MOGA: Multi-objective genetic algorithms. In *Evolutionary Computation, 1995., IEEE International Conference on*, volume 1, page 289. IEEE, 1995.

Murrugarra, David; Veliz-Cuba, Alan; Aguilar, Boris, and Laubenbacher, Reinhard. Identification of control targets in Boolean molecular network models via computational algebra. *BMC Systems Biology*, 10(1):94, 2016.

Newman, Mark EJ. Modularity and community structure in networks. *PNAS*, 103 (23):8577–8582, 2006.

Nicolau, Miguel; Schoenauer, Marc, and Banzhaf, Wolfgang. Evolving genes to balance a pole. In *European Conference on Genetic Programming*, pages 196–207, 2010.

Poli, Riccardo; Kennedy, James, and Blackwell, Tim. Particle swarm optimization. *Swarm Intelligence*, 1(1):33–57, 2007.

Poret, Arnaud and Boissel, Jean-Pierre. An in silico target identification using boolean network attractors: avoiding pathological phenotypes. *Comptes Rendus Biologies*, 337(12):661–678, 2014.

Purcell, Oliver and Lu, Timothy K. Synthetic analog and digital circuits for cellular computation and memory. *Current opinion in biotechnology*, 29:146–155, 2014.

Purdom, G and Anderson, K. Analysis of Barry Hall's research of the *E. coli* ebg operon: Understanding the implications for bacterial adaptation to adverse environments. In *Proceedings of the Sixth International Conference on Creationism*, pages 149–163. Springer, 2008.

Quick, Tom; Nehaniv, Chrystopher L; Dautenhahn, Kerstin, and Roberts, Graham. Evolving embodied genetic regulatory network-driven control systems. In *European Conference on Artificial Life*, pages 266–277. Springer, 2003.

Reil, Torsten. Dynamics of gene expression in an artificial genome—implications for biological and artificial ontogeny. In *European Conference on Artificial Life*, pages 457–466. Springer, 1999.

Ribeiro, Andre; Zhu, Rui, and Kauffman, Stuart A. A general modeling strategy for gene regulatory networks with stochastic dynamics. *J. Comput. Biol.*, 13(9): 1630–1639, 2006.

Roli, Andrea; Manfroni, Mattia; Pinciroli, Carlo, and Birattari, Mauro. On the design of Boolean network robots.

Ros, Vera ID and Hurst, Gregory DD. Lateral gene transfer between prokaryotes and multicellular eukaryotes: ongoing and significant? *BMC biology*, 7(1):20, 2009.

Saadatpour, Assieh and Albert, Réka. Boolean modeling of biological regulatory networks: a methodology tutorial. *Methods*, 62(1):3–12, 2013.

Saadatpour, Assieh; Wang, Rui-Sheng; Liao, Aijun; Liu, Xin; Loughran, Thomas P; Albert, István, and Albert, Réka. Dynamical and structural analysis of a T cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia. *PLoS Computational Biology*, 7(11):e1002267, 2011.

Saez-Rodriguez, Julio; Simeoni, Luca; Lindquist, Jonathan A; Hemenway, Rebecca; Bommhardt, Ursula; Arndt, Boerge; Haus, Utz-Uwe; Weismantel, Robert; Gilles, Ernst D; Klamt, Steffen, and others, . A logical model provides insights into T cell receptor signaling. *PLoS Computational Biology*, 3(8):e163, 2007.

Samuelsson, Björn and Troein, Carl. Superpolynomial growth in the number of attractors in Kauffman networks. *Phys. Rev. Lett.*, 90(9):098701, 2003.

Sarkar, Sahotra and Plutynski, Anya. *A Companion to the Philosophy of Biology*, volume 90. John Wiley & Sons, 2010.

Schaffer, J David. Multiple objective optimization with vector evaluated genetic algorithm. In *Proceeding of the First International Conference of Genetic Algorithms and Their Application*, pages 93–100. Lawrence Erlbaum Associates, 1985.

Schilling, Gabriele; Becher, Mark W; Sharp, Alan H; Jinnah, Hyder A; Duan, Kui; Kotzuk, Joyce A; Slunt, Hilda H; Ratovitski, Tamara; Cooper, Jillian K; Jenkins, Nancy A, and others, . Intranuclear inclusions and neuritic aggregates in transgenic mice expressing a mutant N-terminal fragment of huntingtin. *Human Molecular Genetics*, 8(3):397–407, 1999.

Serra, Roberto; Villani, Marco, and Agostini, Luca. On the dynamics of scale-free Boolean networks. In *Italian Workshop on Neural Nets*, pages 43–49. Springer, 2003.

Shi, Xiaolong; Wang, Zhiyu; Deng, Chenyan; Song, Tao; Pan, Linqiang, and Chen, Zhihua. A novel bio-sensor based on DNA strand displacement. *PloS One*, 9(10): e108856, 2014.

Shi-Jian, Chen and Yi-Guang, Hong. Control of random Boolean networks via average sensitivity of Boolean functions. *Chinese Physics B*, 20(3):036401, 2011.

Shimeld, Sebastian M. The evolution of the hedgehog gene family in chordates: insights from amphioxus hedgehog. *Development Genes and Evolution*, 209(1): 40–47, 1999.

Shmulevich, Ilya; Dougherty, Edward R; Kim, Seungchan, and Zhang, Wei. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

Singh, Vijai. Recent advances and opportunities in synthetic logic gates engineering in living cells. *Systems and Synthetic Biology*, 8(4):271–282, 2014.

Sivanandam, SN and Deepa, SN. *Introduction to Genetic Algorithms*. Springer Science & Business Media, 2007.

Stelling, Jörg; Sauer, Uwe; Szallasi, Zoltan; Doyle, Francis J, and Doyle, John. Robustness of cellular functions. *Cell*, 118(6):675–685, 2004.

Stepney, Susan. Visualising random boolean network dynamics: effects of perturbations and canalisation. In *European Conference on Artificial Life*, pages 369–376. Springer, 2009.

Taou, Nadia S and Lones, Michael A. Towards in Vivo Genetic Programming: Evolving Boolean Networks to Determine Cell States. In *European Conference on Genetic Programming*, pages 151–165. Springer, 2018.

Taou, Nadia S; Corne, David W, and Lones, Michael A. Towards Intelligent Biological Control: Controlling Boolean Networks with Boolean Networks. In *European Conference on the Applications of Evolutionary Computation*, pages 351–362. Springer, 2016a.

Taou, Nadia S; Corne, David W, and Lones, Michael A. Evolving Boolean networks for biological control: State space targeting in scale free Boolean networks. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2016 IEEE Conference on*, pages 1–6. IEEE, 2016b.

Taou, Nadia S; Corne, David W, and Lones, Michael A. Investigating the use of Boolean networks for the control of gene regulatory networks. *Journal of Computational Science*, 26:147–156, 2018.

Taylor, Tim. A genetic regulatory network-inspired real-time controller for a group of underwater robots. In *Intelligent Autonomous Systems*, volume 8, pages 403–412. Citeseer, 2004.

Trefzer, Martin A; Kuyucu, Tüze; Miller, Julian F, and Tyrrell, Andy M. Evolution and analysis of a robot controller based on a gene regulatory network. In *International Conference on Evolvable Systems*, pages 61–72. Springer, 2010.

Turner, Alexander P; Lones, Michael A; Fuente, Luis A; Stepney, Susan; Caves, Leo SD, and Tyrrell, Andy. The artificial epigenetic network. In *Evolvable Systems (ICES), 2013 IEEE International Conference on*, pages 66–72. Springer, 2013.

Veliz-Cuba, Alan and Stigler, Brandilyn. Boolean models can explain bistability in the lac operon. *J. Comput. Biol.*, 18(6):783–794, 2011.

Veliz-Cuba, Alan; Arthur, Joseph; Hochstetler, Laura; Klomps, Victoria, and Korpi, Erikka. On the relationship of steady states of continuous and discrete models arising from biology. *Bulletin of Mathematical Biology*, 74(12):2779–2792, 2012.

Veliz-Cuba, Alan; Aguilar, Boris; Hinkelmann, Franziska, and Laubenbacher, Reinhard. Steady state analysis of Boolean molecular network models via model reduction and computational algebra. *BMC Bioinformatics*, 15(1):221, 2014.

Von Neumann, John; Burks, Arthur W, and others, . Theory of self-reproducing automata. *IEEE Transactions on Neural Networks*, 5(1):3–14, 1966.

Wilson, Brenda A; Salyers, Abigail A; Whitt, Dixie D; Winkler, Malcolm E, and others, . *Bacterial Pathogenesis: a Molecular Approach.* Number Ed. 3. American Society for Microbiology (ASM), 2011.

Yao, Xin and Liu, Yong. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, 8(3):694–713, 1997.

Yao, Yao; Marchal, Kathleen, and Van de Peer, Yves. Improving the adaptability of simulated evolutionary swarm robots in dynamically changing environments. *PloS One*, 9(3):e90695, 2014.

Zahadat, Payam; Christensen, David Johan; Schultz, Ulrik Pagh; Katebi, Serajeddin, and Stoy, Kasper. Fractal gene regulatory networks for robust locomotion control of modular robots. In *International Conference on Simulation of Adaptive Behavior*, pages 544–554. Springer, 2010.

Zanin, Massimiliano and Pisarchik, Alexander N. Boolean networks for cryptography and secure communication. *Nonlinear Science Letters B: Chaos, Fractal and Synchronization. Vol*, 1(1):27–34, 2011.

Zhan, Song; Miller, Julian F, and Tyrrell, Andy M. An evolutionary system using

development and artificial Genetic Regulatory Networks for electronic circuit design. *Biosystems*, 98(3):176–192, 2009.

Zhang, Shu-Qin; Hayashida, Morihiro; Akutsu, Tatsuya; Ching, Wai-Ki, and Ng, Michael K. Algorithms for finding small attractors in Boolean networks. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007(1):20180, 2007.

Zhao, Yin; Li, Zhiqiang, and Cheng, Daizhan. Optimal control of logical control networks. *IEEE Transactions on Automatic Control*, 56(8):1766–1776, 2011.

Zitzler, Eckart and Thiele, Lothar. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput*, 3(4):257–271, 1999.

Zitzler, Eckart; Laumanns, Marco, and Thiele, Lothar. SPEA2: Improving the strength Pareto evolutionary algorithm. 2001.