CHAPTER 2

BACKGROUND

2.1 Introduction

The developmental process of multi-cellular growth and its interaction with natural evolution has always attracted much interest. With the rapid advancement of computational resources, investigation utilising simulation of such processes has become a improving prospect. It is a subject that has been widely recognised as being of fundamental importance by researchers in various strands of evolutionary computation and evolutionary biology. However, even with ever improving resources, progress has proven to be slow in coming and significant results rare and difficult to obtain.

As is often the case with modern research, this work exploits many existing research fields, overlapping them in numerous ways. This overlap in itself can cause difficulties, in particular with terms that do not carry over well between the fields. A good example is the commonly utilised term ‘development’. For some it would be perfectly valid to describe the work presented here as chiefly concerned with developmental processes and their relationship to evolution. However this could be taken to mean a number of very different things. For an evolutionary biologist, development can refer to two aspects, the evolutionary development of a species or the cellular development of an individual from single cell to maturity. These two aspects are highly interlinked and so
the term evolutionary developmental biology, shortened to evo-devo, is used [56]. For a psychologist, development has a very different meaning. It is used in the context of describing the change in an individual’s cognitive, emotional, intellectual, and social abilities through the course of its life span [27].

In reality both psychological development and evo-devo are just two sides of the same biological phenomena and are highly dependent upon each other [58]. In this work the term development will be used to refer to all of these concepts as a whole with specific individual concepts referred to directly throughout as necessary.

The remainder of this chapter will provide an overview of the history and current state of relevant research fields.

2.2 Evolutionary Computation

We are becoming increasingly constrained by the approach we take to engineering and developing new technologies as our technological requirements become more demanding [104, 11]. Nature can be of great inspiration to those scientists and engineers who strive to create a new generation of technologies that overcome the limitations of existing ones. Such approaches are often themselves termed ‘nature inspired’ and the benefits are widely displayed. Biological systems are capable of adaptation, self-organisation, self-repair and levels of complexity to which our current engineering endeavours pale in comparison.

The field of evolutionary computation has rapidly become one of the defining research fields in nature inspired techniques and of computer science in general. It is based on an extension of the principles of natural selection to form population based stochastic search algorithms, which are commonly known as Evolutionary Algorithms (EAs). Several fundamental types of EA have been defined such as Genetic Algorithms (GAs) [71], Genetic Programming (GP) [84], Evolutionary Strategies (ES’s) [115, 8] and Evolutionary Programming (EP) [49] to name but a few [17, 105]. In reality the field is much harder to
pigeonhole and generally varies in its application such that identified types tend to blend into, or are extensions of each other.

2.2.1 The Simple Genetic Algorithm

The simple genetic algorithm (sGA) was an early outline of simulated evolution in its simplest form as a population based stochastic search algorithm [71, 54]. It provides the fundamental basis from which most current evolutionary systems are extended. Figure 2.1 shows the various processes that make up the sGA.

Initialisation

Initialisation requires the creation of an initial population consisting of a set of individuals. Each individual is represented as a string of values that may be binary, integers, real or alphanumeric. For the initial population these values are usually generated randomly within the constraints of the representation used.

The choice of population size is very much problem dependent. It can be allowed to vary or it can be fixed large enough to avoid sensitivity to initial conditions and early convergence. For some problems in may be prudent to take the best, or average, of a number of evolutionary runs with independently randomised initial populations. Parallels can be drawn between the initial population of a GA and the random initialisation of weights in a neural network [83].

Evaluation

Each individual in the population is evaluated. The means of evaluation is often referred to as the fitness function and returns a numerical value representative of an individuals performance upon this function. The evaluation process is often the most computationally expensive of the different stages of the evolutionary process and includes the mapping of the genotypic representation (genotype) to the potential solution (phenotype).
Figure 2.1: Processes of the simple genetic algorithm.
Stopping Criteria

Since most sGA’s at some point converge to either a local or global optima and get stuck, there is no point in continuing the evolutionary process indefinitely. The stopping criteria of evolution often consist of a number of dependencies:

- Determining if the most optimal or a satisfactory solution has been discovered. Of course this requires the knowledge of what the optimal solution is or at least the fitness value to which it corresponds.

- A maximum cap on evolutionary generations, number of evaluations or computational time.

- Fitness has not improved for some predetermined amount of time, evaluations or generations.

Selection

Potential parents are selected from the population with a bias toward fitter individuals. There are a number of approaches to selection including:

- Proportional selection, used in the original GA developed by Holland [71], ensures that the probability of selection for an individual is proportional to its fitness relative to the rest of the population. The roulette wheel analogy makes this more explicit, where each individual is allocated an area on the wheel, in relative proportion to its fitness.

- Rank Based selection is simply a matter of ordering the population in terms of fitness. A probability distribution, typically linear, is then applied across the ranking. This causes a normalising of the range of the raw fitness values of the
population and enables a consistent selection pressure across the population at each
generation [9].

- Tournament selection involves selecting a subset of the population of individuals at
random and then choosing the fitter of the subset to act as a parent. The selection
pressure is dependent upon the size of the subset which is selected at random [55].

These parents are then exposed to a set of search operators in order to produce a
number of offspring. Search operators can be categorised into two forms, those that
exchange information between individuals (crossover) and those that apply independent
alterations to an individual (mutation).

Crossover

Crossover, also known as recombination, implies an exchange or merge of individual
elements or values (alleles) between individuals. There are many different forms of
crossover such as one-point, two-point and uniform, but essentially the only real difference
is in how the elements, which are to be exchanged or merged, are defined. Its success is
highly dependent upon the genetic representation.

Researchers disagree on the importance and role of crossover in both natural and
simulated evolutionary systems. Early views, originating from Holland’s original work
[71] is that crossover is the primary force for adaptation in a GA whilst mutation prevents
stagnation. More focus has now been given to the role of mutation in the GA.

It is also useful to note, however, that crossover is not ubiquitously throughout natural
evolution [121]. Sexual reproduction utilises crossover whereas asexual reproduction does
not, relying instead on more elaborate forms of hereditary information exchange between
individuals.
Mutation

Mutation operators alter individual, or sets of, alleles. These alterations commonly consist of either random replacement, irrespective of the allele value, or a random shift in value. Mutations can be equated to local search operators and as such define a neighbourhood structure across the search space such that the neighbours of any given individual are potential mutants.

Replacement

The offspring, resulting from search operators acting on selected parents, are then integrated into the population by replacing a number of existing individuals following a set of predefined rules to form the next generation of individuals.

The number of offspring that are placed into the population at each generation can have important effects upon the evolutionary dynamics. One extreme is the steady state GA in which only one individual is changed in the population at each generation. The other extreme is the generational GA that ensures that at each generation the entire population is replaced by new offspring. Since these differing forms of recombination are essentially part of the selection process they have important effects upon the selection pressure of a simulated evolutionary system. Generational GA’s tend to converge very quickly to optima but can prove extremely noisy with rapid changes in average population fitness and diversity. Steady state GA’s are generally slower to converge but ensure smooth changes in the average population fitness and diversity over time. They also exhibit much more biologically plausible behaviour than the generational approach [31].

Elitism is a common tactic to ensure that the optimal fitness of the population does not decrease during evolution by always retaining the fittest individual in the population during replacement [76]. It is especially useful for algorithms where there is a high chance of the fittest individual being replaced, as is common in generational approaches.
2.2.2 Theorems of Simulated Evolution

An important point to note about evolutionary algorithms is that they are stochastic population based processes. This means that they are not deterministic and so the next state of the system cannot be exactly predicted based only upon the previous state. This is due to the application of random search operators which ensure the next state is only partially derived from the previous state.

There have been numerous approaches to try to explain, within a theoretical or analogous background, the fundamental processes at work within a simulated evolutionary system. Although these theorems have advanced our understanding of how simulated evolution works, producing a successful algorithm is still something of a black art.

The Search Landscape

Search operators are at the core of any search algorithm. We can consider these operators, particularly mutation, as defining a neighbourhood structure determining which individuals can be reached from any given individual. We can imagine this neighbourhood structure as a landscape upon which the search process moves around. This is a very useful tool since it allows us to visualise how a search algorithm may perform on a given task by relating it to the position and movement of individuals upon this landscape.

There is, however, a problem with this approach. Neighbourhood structures are easy to visualise for a mutation operator since mutation can only imply a fixed range of individuals. However, for crossover in its various forms, where a pair of individuals are involved, how this impacts on neighbourhood structure is much more obscure and begins to muddy the simple search landscape analogy.

This analogy becomes even less plausible if we consider cases that are not simply directly encoded. In cases where the genetic representation is mapped in some indirect way
to the phenotypic representation there is no longer a fixed relationships between the space through which evolution traverses and the space in which evaluation occurs. Therefore relative positions and movement are meaningless unless you consider both spaces.

**No Free Lunch**

The No Free Lunch Theorem (NFLT) [137] shows that for all possible problems, every search algorithm will perform as well as any other, on average. From this, it can be inferred that the existence of some ubiquitously superior search process does not exist. Probably the most importance consequence of NFLT is that it has been made explicit that as much domain knowledge as possible should be used when implementing a search process to achieve an efficient optimisation or search.

**2.2.3 Encoding and Representation**

In most cases, especially for real world applications, the representation of an individual at the genetic level (genotype) has to be mapped to some useful solution with a measurable performance (phenotype). However, when theorising about the genetic algorithm, this mapping is often ignored. In such cases the mapping between genotype and phenotype can be considered to be direct, or almost direct, such that the genotype is essentially also the phenotype.

The *One Max* problem is a typical example where the objective is to maximise the number of 1’s in a binary vector of length $n$. For this problem a simple steepest ascent hill-climbing algorithm outperforms an sGA. Figure 2.2 shows the typical number of evaluations required to solve the *One Max* problem for various sizes of binary vector for random walk, steepest ascent hill climbing and an sGA. In this example the sGA utilises a population size equal to the length of the binary vector and utilises a single point mutation and crossover with roulette wheel selection. The performance of the sGA lies between the simple hill climbing algorithm and the exponential increases observed for a random walk.
These results highlight the point that evolutionary algorithms are not always the best option. For simpler problems less elaborate search techniques often result in better performance. Simulated evolutionary search process are often better suited to noisy search landscapes with multiple local optima and discontinuous regions [54, 105].

The results in figure 2.2 also demonstrate that in order for simulated evolution to tackle increasingly complex engineering and research problems the search space of a direct encoding would also become increasingly difficult to navigate. This has been a major influence upon the movement to more indirect encoding approaches. An indirect encoding describes any evolutionary system that requires a mapping process to determine the phenotype from the genotype (genotype-phenotype mapping). This effectively creates
two different spaces between which relationships are not necessarily preserved. These mappings can exist in many forms [124, 10, 18, 57].

2.2.4 Hybrid Evolutionary Systems

Hybrid evolutionary systems refer to those approaches that utilise an evolutionary technique in conjunction with some other search process. The advantages of hybrid systems are often very problem dependent but some of the most efficient evolutionary algorithms use a hybrid approach. The most successful hybrid approaches employ a combination of search techniques that overcome each other’s failings. In most cases this involves using a population based stochastic search process alongside a heuristic based local search process. These are commonly known as memetic algorithms [66].

Liang et al. [95] showed that combining a landscape approximating evolutionary algorithm with a simple local search technique allowed the hybrid search algorithm to traverse very noisy and rugged landscapes without getting stuck in local optima due to convergence.

Yao et al. [139] observed that the information held in each run of a simulated annealing algorithm could be utilised by connecting them together to form generations in a genetic algorithm. The results showed potential improvements in performance over both simulated annealing and genetic algorithm approaches alone.

Hacker et al. [62] showed how population information in a genetic algorithm can be used for regression analysis. This allows their hybrid approach to choose between local and global search depending upon the quality of the fit of a second order function to the population.

Typically, the successful application of a hybrid approach is extremely problem dependent and so finding an efficient algorithm can itself be a lengthy search process unless prior domain knowledge can be applied.
2.2.5 Evolvability

Evolvability can be defined as the ability of an entity in an evolutionary framework to continually create new and useful adaptations [79] and can be thought of as a measure of evolutionary activity. In a more specific sense it can also be thought of as an ability to continually produce variants from a population that are as fit if not fitter than any existing individuals in the population. Ensuring and sustaining this capability is the core of much research in the evolutionary computation field [4, 111, 120]. Altenberg suggests that evolvability should form a critical performance measure of an evolutionary algorithm [4].

Dawkins coined the term ‘Evolution of Evolvability’ referring to the ability of evolution to adapt to ensure a sustained capability for evolvability throughout the evolutionary process [40]. Several approaches have been taken to try to integrate this ability into a simulated evolutionary system as highlighted by Altenberg [4]. The following are just a small representative sample:

- Using linkage between genes, which is considered to be a measure of likelihood, that if a set of genes is manipulated by search operators, that those genes stay together. Hence it is often common to refer to sets of genes with a relatively strong linkage as a ‘building block’. In order for this approach to be of use an evolutionary system must have both a way of defining linkage and a way of enforcing behaviour of the search operators in respect of this linkage. Linkage learning refers to the ability to adapt the linkage values in an evolutionary system based upon performance feedback [65]. In essence, linkage learning should provide evolution with the ability to discern which sets of genes, when used in a given formation, become advantageous [111].

- Rather than use blind search operators to produce offspring, one option is to use machine learning techniques. The Learnable Evolution Model (LEM) [99] works
by utilising a machine learning method which differentiates between the better individuals and poorer individuals in a population. It then utilises these differences to generate new individuals. Various machine learning techniques can be applied but their success has been shown to be highly problem dependent.

- Altenberg’s ‘Constructional Selection’ [5, 3, 4] is a hypothesis about how the mapping process between genotype and phenotype impacts upon the evolvability of an evolutionary system. He argues that selection pressure during the creation of novel genes constrains the mapping process in order to make it naturally more evolvable. The premise is that all genes in a genome at some point must have had some selective advantage in order to become a stable part of the genome and so must be related to some useful phenotypic variation. However new genes create new variational possibilities for evolution and so in this manner variational properties of the genotype can become correlated with variational properties of the phenotype. This hypothesis is unique in that it not only accounts for, but also relies upon, a disassociation between the genotype and phenotype and so may prove important to the work presented in this thesis.

2.2.6 Neutrality

The evolutionary rate is a measure of how often a particular characteristic changes in an evolutionary system. Kimura [78] showed that the evolutionary rate at the molecular level across mammalian evolution was surprisingly and consistently higher than expected. He suggested that this be reconciled by assuming a large percentage of these changes, or mutations, have a neutral effect such that there is no change in the function or appearance of an organism. This is now known to be broadly correct and neutral evolution has become an extremely important tool for evolutionary biologists with much discussion about its purpose [125].
Miglino et al [100] argued that neutral changes could be functional in the sense that they enable *pre-adaptations* which subsequently become the basis for an adaptive change. This may explain the unsteady discontinuous rate of change, which is observed in natural evolution at the phenotypic level, interspersed with dramatic and abrupt adaptations.

The concept of neutrality also has important implications for computational search processes because it changes the nature of local optima. In a search space where no neutrality exists local optima are typically considered to be single points in the space upon which a converging evolutionary process can easily become trapped. However, neutrality means that the optima may not be localised in the same way, in fact they can be vastly distributed across the entire search space. What happens to a population, which inhabits a neutral network, is largely dependent upon how the evolutionary process has been defined. Typically, the result is a random drifting (neutral drift) of the population until a path to a fitter individual is found and then adaptive evolution recommences. This enables the population to appear converged on local optima at the phenotypic level whilst neutral drift allows for genetic divergence.

During neutral evolution, by definition, there can be no directed selection pressure. However selection is still performed, albeit randomly, and this does affect the population structure. Wilke [135] argued that some of these randomly selected parents will be more likely to produce viable offspring than others. The more densely connected the neutral network an individual occupies the more likely that its offspring will be neutral rather than deleterious. This will place a bias on evolution toward more densely connected neutral networks. However the effect of neutrality on a search process is still an open question.

Hogeweg [69] showed that, using a model of morphogenesis, the behaviour of mutated individuals during neutral evolution, which was termed the *mutational shadow* of the neutral path, was noticeably different than during adaptive evolution.

There have been many arguments for how neutrality may result in an improved
Evolvability. Ebner et al. [44] suggested that the redundancy causing the neutrality also increases the likelihood that evolution can find a smooth and amiable search path and that neutral networks help to prevent early convergence. Smith et al. [120] conducted an investigation into whether neutral evolution altered the evolvability of an evolutionary system for the evolution of robot controllers. Their results suggested that neutral evolution was not performing any “useful” task and made little difference to the evolvability of an evolutionary system. This conclusion may, of course, be entirely problem dependent.

2.2.7 Computational Power

In order to make comparisons between different evolutionary approaches and other search processes, and to evaluate the characteristics of such processes, some measure of performance is required. For a computational process the amount of computational resources required to complete the task is a common measure. The computational cost, or performance, of a simulated evolutionary process has been measured in numerous ways. The following are the three most common methods:

- **Number of evaluations** is probably the most widely used and respected measure of evolutionary performance however there are cases where the computational cost of the evaluation process varies across individuals. This is typical of EAs that incorporate a mapping process between the genotype and phenotype.

- **Number of generations** can be an effective measure of performance but only if the number and computational cost of these evaluations is consistent across generations.

- **Computational time** is the most direct way of measuring performance but results can be misleading unless the processing power allocated to simulating evolution can be guaranteed to be consistent, something which is very hard to achieve across various processors using modern operating systems.
Overall, accurately measuring the performance of a simulated evolutionary system is only really practical for very simple evolutionary simulations, using direct encodings on simple problems, where the exceptions discussed above rarely occur. For more complex approaches accurate measures become much more difficult to obtain.

In the majority of cases it is the evaluation process that accounts for most of the computational costs of an evolutionary approach. In cases where this evaluation requires the modelling of complex or dynamical systems these costs can be high. Several approaches have been taken in order to either reduce or overcome the computational costs of such processes. The most obvious is to simply distribute the evaluation of a population of individuals over several processors, termed Parallel Genetic Algorithms (PGA’s) [108, 35]. Probably the most extreme example of this is by Astor and Adami [6, 7] who utilised a distributed client-server system that allowed the evaluation process to be distributed from a server to clients across a network of machines. Their aim was to set up a community of Internet users who donate CPU time.

This distribution of the processing task can be made more analogous to nature by using the island model, sometimes referred to as the course grained PGA. This is where a set of populations, called islands, are essentially exposed to independent evolutionary regimes which may have differing selection pressures and search operators. By migrating individuals between islands, which are selected and timed according to a predetermined set of rules, the aim is to positively influence the evolution of each population. The overall objective is to produce an evolutionary system that obtains improved results over an evolutionary system that contains only a single population. Island model approaches are now quite commonplace in all aspects of the evolutionary computation field and have even shown improvements over a single population approach when simulated on a single processor [38, 134].
There is also a fine grained approach to PGA’s. In a similar fashion to the island model, the population is split into a set of sub-populations. These sub-populations are arranged into a grid with a set neighbourhood structure. Migrations are only allowed to occur between neighbours. Ideally each sub-population should be as small as possible, preferably containing as little as one individual. As with the island model each sub-population may be handled by separate processors but there are also benefits from simulation on a single processor [35].

The fact that a simple change in the representation of population structure on a single processor can have such an impact suggests that in most cases simulated evolution is far from optimal in terms of usage of computational resources.

### 2.2.8 Open-Ended Evolution

Most simulated evolutionary systems are directed toward some optimal solution utilising a selection bias based on some evaluation method. Many researchers argue that it is this directing of evolution that limits its capabilities. In natural evolution such direction does not exist since the selection bias of an individual is an inherent property of the evolutionary system itself. This form of evolution has no definitive end, with the potential for continuous evolutionary emergence of novel adaptations and behaviours, and so is referred to as open-ended evolution [123].

There have been various attempts at simulating an open-ended evolutionary system, Tierra being probably the most well cited example [113]. Based on simple self-replicating programs that compete for CPU time, these programs evolve to form more complex functions such as to develop attack and defence mechanisms. However with the possible exception of “Geb” by Channon [37, 36] none of these attempts have truly achieved open-ended evolution with the number of novel evolutionary adaptations flagging as the evolutionary process concludes [127, 114].
2.2.9 Observations about the Inherent Nature of Simulated Evolution

There are a number of disparities across the simulated evolutionary and natural evolutionary fields. One stark difference is how population diversity is considered. In nature, population diversity is a highly variable and transient thing [125]. It is also highly dependent upon how it is measured, for example at the genotype or phenotype level. However in simulated evolution it is almost always argued that diversity should be maintained and maximised in some way until the global optima has been discovered. It is argued that, in terms of search landscapes, spreading the population over more of the landscape will help prevent the search process from early convergence to local optima. However this analogy can be flawed [39] especially when neutral landscapes exist since it is possible for a population to fracture and thus diverge in different ways across differing neutral networks [52].

Species Adaptation Genetic Algorithms (SAGA) are a good example of this point since they use a relatively converged population as compared to a standard GA [67]. Local optima are still a problem for SAGA, however, for a large class of problems, SAGA is able to use its mutation driven approach to move around the search space utilising the neutral networks introduced by genetic redundancy and the genotype to phenotype mapping process.

Introducing such complex mappings as a developmental process to the genotype-phenotype map is very often naively viewed as an added complication that makes simulating and understanding evolution even more difficult. However, it is important to recognise that this mapping process can effectively be used to direct the evolutionary path and so without it, there may be no possibility of truly understanding evolutionary systems. Wagner and Altenberg [133] used the analogy of a monkey and a verse of
Shakespeare to describe the importance of a mapping process. If a monkey randomly presses keys on a typewriter it is far more likely to produce a verse of Shakespeare than if it was given a pencil and paper. This is due to mechanical constraints of the typewriter to produce only viable text characters. In the same fashion genetic variation can have a constrained effect on the phenotype via the genotype-phenotype map.

One could argue that the evolutionary search process itself is not the really interesting part of evolution. After all it is just another form of search comparable in many ways to other search processes. What is really interesting about evolution is the systems and behaviours that have been observed to have built up around it in both natural and simulated evolutionary systems.

2.3 Developmental Biology

Developmental biology is the study of how organisms grow and develop. It incorporates the study of genetics, cell structure, cell mechanisms, and the function and anatomy of organs and tissue. When the relationship with evolutionary processes is also taken into account the field is expanded and termed evolutionary developmental biology (evo-devo). The following section describes just a small but relevant part of this field, only entering into more detailed analysis where deemed necessary.

2.3.1 Genetics

Genetic Regulatory Networks

Sequencing the genes from an organism alone will tell you very little about how that organism functions and how it came to be. In order to do this a more functional understanding of how each of these genes interact is required. Genetic regulatory networks (GRN’s) are collections of genes which interact with each other and with other substances to form a complex network.
New technologies, such as micro-array devices, allow us to determine gene interactions on an increasingly larger scale and the data that they produce can be used to reconstruct the GRN in which the genes interact. Computational systems are used to infer the overall genetic networks from these observed interactions [98]. Ironically, one of the most promising ways of reconstructing GRN’s from this new data has proven to be simulated evolutionary search processes [38].

The Operon model was proposed by François Jacob and Jacques Monod and is used to explain the operation of genes and how they can express, and be expressed by, each other [2] such that genes are either active or inactive, depending on the state of other genes. This demonstrates how genes are able to form GRN structures. An Operon is a set of genes that are transcribed in order to produce proteins or ribonucleic acid molecules (RNA). There are two classes of genes, structural genes, which code for proteins that are required for the operation of the cell, and regulatory genes, which code for regulatory proteins whose function is to regulate the expression of other genes. The regulatory part of an operon consists of a promotor and operator. The promotor is where gene expression is begun. The operator can control whether or not the remaining operon genes are expressed.

The action of a regulatory gene is dependent upon the presence or absence of a relevant regulatory protein. If the action of the regulatory protein is to bond with the operator in order to prevent gene expression then it is known as a repressor. This form of regulation is known as negative regulation of which there are two forms:

- **Negative Inducible Operons** refer to the case where the regulatory repressor protein is normally able to bond with the operator preventing expression of the genes in the operon. However, if an inducer molecule is present, it binds to the repressor protein preventing it from bonding with the operator and allowing the expression of the genes in the Operon.

- **Negative Repressible Operons** describe the opposite case. In order for the repressor
protein to be able to bond with the operator an effector protein must be present. Both repressor and effector proteins must be present to prevent expression of the genes within the Operon.

Operons may also be positively regulated such that bonding of an activator protein with the operator causes the expression of the genes within the Operon. Again two forms of positive regulation exist:

- **Positive Inducible Operons** have activator proteins which are usually unable to bond to the operator. An inducer protein must bond with the activator in order for the activator to bond with the operator. Both activator and inducer must be present to ensure the expression of genes within the operon.

- **Positive Repressible Operons** describe the opposite case in which the activator protein is normally bound to the operator. However, the existence of a effector protein can prevent bonding of the activator to the operator and thus preventing expression of the genes within the Operon.

Figure 2.3 shows the case of a negative repressible Operon in which no effector substance is present. This prevents the activation of the repressor and so the repressor does not bond to the operator site of the Operon. This allows RNA polymerase to bond to the promoter and so the genes are transcribed and thus expressed. In figure 2.4, the effector is present and so activates the repressor protein. This allows the repressor to bond to the operator site and so preventing transcription of the genes.

Since the only function of a gene, if expressed, is to specify some given protein, and the transcription and thus expression of these genes is dependent upon proteins, then proteins are how the genetic regulatory network must interact within itself and the external environment.
Figure 2.3: Model of negative repressible Operon in which the absence of an effector prevents the repressor from bonding with the operator gene and so RNA polymerase bonds with the promoter gene allowing the regulated genes to be expressed.

Figure 2.4: Model of negative repressible Operon in which the presence of an effector allows the repressor to bond with the operator blocking the bonding of RNA polymerase and thus expression of the regulated genes.

**Heterochrony and Canalization**

The timing of events in cell development can be critical. Changing these timings can have a huge impact upon the evolved phenotype, known as heterochronic change [116]. Heterochrony can present itself as a change in the timing of a biological developmental process, as a change in the time at which a process starts or stops or as a change in the speed at which a process is conducted.

A striking example is the Hercules beetle, *Dynastes hercules*, which shows a severe dimorphism in the physical appearance between male and female due to the growth of a gigantic horn-like protrusion from the head of the male. This is caused by a biological
development process simply being allowed to continue for a greater length of time than it
occurs in the female of the species [43].

Heterochrony offers the possibility of enabling large changes in a single phenotypic
characteristic from small genetic alterations with minimal interference with other
biological developmental processes. However, the potential for large changes due to small
genetic mutations raises the issue of how damaging such mutations could potentially be.

Heterochrony plays a large part in the development and evolution of neuronal
structures and as such as been the target as several researchers in this area [32, 33].

As well as providing mechanisms for phenotypic variation, such as heterochrony,
evolution has formed the ability to control these mechanisms. Canalisation, in a
evolutionary sense, refers to the suppression of phenotypic variation [125]. Genotypic
canalisation refers to the ability to constrain the effects of genetic mutations on the
phenotype. If used correctly it can prevent damaging mutations whilst still enabling
useful or neutral mutations to occur. Environmental Canalisation refers to the resistance
of a phenotype to variation by environmental influences.

2.3.2 Cell Function

Cell Division

Cell division is of obvious key importance to the growth of multi-cellular organisms. It
involves the replication of cells by physical division. Cells divide by first duplicating the
genetic material held within the cell and then, by a process known a cytokinesis, the
cell physically divides into two with each side holding a copy of the duplicated genetic
information. This whole process is triggered by chemical interactions in the cell as a
consequence of changes in the cell state. In most cases, cell mass is doubled and vital
components duplicated before division. In this case daughter and parent cells can be
considered almost identical. There are exceptions, such as cleavage, where division occurs
with no increase in mass. In some cells, especially those that form in layers, there is a polarity in the internal structure of the cell. In these cases it is common for the division process to be orientated relative to this polarisation [2].

**Cell Differentiation**

Cell differentiation is the process by which a cell becomes specialised in form or function. Essentially a cell is considered to have differentiated if it’s form or function has changed in some manner. Examples of cell types include neurons, epidermis and blood cells.

Some cells can only differentiate into a single cell type (*unipotent*) or a small group of cell types (*multipotent*), whilst other cells can differentiate into a large variety of differing cell types (*pluripotent*). Cells that appear to have the potential to differentiate into any form of cell type found within the organism are *totipotent*, more commonly known as *stem cells*. In general cells that are more highly differentiated find it more difficult, or unlikely, to redifferentiate to a different cell type. Stem cells are the least highly differentiated cells and can differentiate to form many different cell types.

Cell differentiation is an important process in enabling multi-cellular organisms to consist of a large variety of highly functionally specialised cells. This ability to redifferentiate is also a key aspect of multi-cellular growth. This has been shown to be critical to the regenerative capabilities of some creatures such as the newt which can regenerate limbs which are subject to damage [28]. This is because the cells have enough plasticity to allow them to re-differentiate into stem-like forms of cell which then go on to re-grow the damaged component. This capability is, of course, of vast interest to those aspiring to enhance these biological traits or implement similar capabilities into man-made structures such as electronic hardware [101, 102].

The important aspects of cell differentiation are not just simply the function that a differentiated cell performs but also how and why a cell is instructed to differentiate. This
leads to the question of what external influences motivate this instruction?

Spatial differentiation refers to situations where differentiation is obviously instantiated due to cell position. Wolpert [138] suggested that cells exploit positional information in the form of thresholding of diffused chemicals to perform this task. This process, along with the diffusion of chemicals, will be explained in further detail in chapter 3.

**Morphogenesis**

The capability of cells to differentiate based on positional information highlights the capability of multi-cellular organisms to organise their internal shape, form and structure. This process is known as morphogenesis and is highly dependent on the presence of diffusible chemicals, called morphogens. These morphogens control the topological structuring of cells based on chemical concentration patterns.

The diffusion rate of a chemical determines how far the impact of that chemical can reach. Some can be extremely long range spreading throughout the mass of an organism whilst others only reach neighbouring cells. Morphogens may also undergo a variety of chemical reactions with each other to produce spatial concentration patterns. Turing [129] gave some of the earliest mathematical explanations for this behaviour with the use of reaction-diffusion equations which have since been shown capable of replicating many of the patterns observed in nature [47, 63].

Morphogenesis combined with differentiation is the key to understanding the biological development of a multi-cellular organism but these processes in turn are highly dependent on how cells interact and communicate.

**Cell Adhesion**

Cell Adhesion refers to the ability of cells to form a physical bond with other cells and material. This is critically important to the morphogenesis of 3-dimensional multi-cellular structure since it enables the creation of solid tissue and the ability to apply force to the
physical environment.

Cells also have the ability to build structures outside of the cell, known as the extra-
cellular matrix, to which they can adhere. These cell structures are typically known as
connective tissue and include bones, ligaments and tendons. This provides a fundamental
physical framework that helps to give a multi-cellular organism strength and agility.

Cell adhesion is not commonly simulated in cell development models. However,
Hogeweg [68, 70] showed that cell adhesion could allow cells to differentiate and thus
form patterns and structures without the need for morphogens or other gradient based
processes.

Cell Signalling

One can argue that, in order for numerous cells to form a multi-cellular structure with
behaviour consistent with a single organism, there must exist a capability for cells to
communicate. Cell signalling provides the ability to communicate and there are two
primary methods through which it is conducted.

Direct cell-cell communication describes the case when a cell directs the form of
communication to a specific cell. Since, typically, the extra-cellular environment has
no means to support this, direct contact between cells is required. Information is then
either passed directly through the cell wall to an internal signal receptor or via signal
receptors on the surface of the cell.

Indirect cell-cell communication prevents direct control over which cells receive the
signal but allows longer-range communication. Cells secrete cell-signalling molecules
that diffuse in the extra-cellular environment. Cells can receive these diffused signalling
molecules in the same way as direct cell-cell communication, either at the cell surface or
within the body of the cell.

Of course, as in all areas of biology, there are exceptions to these cases where evolution
has adapted cells for a specific function. For example neuronal cells are specialised in enabling direct cell-cell interactions but over relatively larger distances by using axonal and dendritic protrusions from the cells that use a mixture of chemical and electrical signals. Other examples include endocrine cells which secrete cell signalling molecules into the blood stream carrying the signalling molecule quickly around the body enabling fast communication with cells throughout the body [2].

2.4 Modelling Evolutionary Biological Development

Scientists have been trying to explain the purpose of, or reason for, the biological cell growth process for centuries. In 1828 von Baer, a comparative developmental biologist, announced that the embryo passed through several stages of development which, in the early stages, was similar amongst most species. The concept of embryonic growth was not new but it was thought that it resembled the adult in a smaller or more primitive form [125].

As the science behind this process has been uncovered we have been able to explain in much greater detail the process of evolutionary biological development [2]. However, understanding how this process occurs has not been able to clearly explain why it occurs. It is for this reason that the modelling of these cell development processes has become such an important part of increasing our understanding from both the scientific and engineering viewpoints.

This section will review some of the key previous attempts at producing a model of cell growth, and will highlight the strengths, weaknesses and similarities between these attempts.
<table>
<thead>
<tr>
<th>Rewrite rules</th>
<th>Steps</th>
<th>Resultant string</th>
</tr>
</thead>
<tbody>
<tr>
<td>A → AB</td>
<td>0</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>AB</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>ABCBA</td>
</tr>
<tr>
<td>B → CBA</td>
<td>3</td>
<td>ABCBACCCBAABABCBBAABBCBA</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>ABCBACCCBAABCCBAABABCBABCBAABCBABCBA</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>ABCBACCCBAABCCBAABABCBABCBAABCBABCBAABCBABCBA</td>
</tr>
</tbody>
</table>

Table 2.1: Simple example of L-System with re-write rules.

2.4.1 Rewrite Grammars

One of the earliest forms of what is considered to be biological developmental modelling was performed by Lindenmayer who developed a mathematical construct called *L-Systems* [96]. The purpose of L-systems was to provide a formal description of the processes involved in cell development of plants. The approach consists of a predefined set of symbolic re-write rules. Table 2.1 shows an example set of symbolic rewrite rules and their resultant effect upon a simple initial string of symbols. Translated into plant characteristics, such as branch lengths and angles, this simple rewrite grammar approach is able to express growing forms extremely similar to that of real plants.

Boers [19] extended this approach by applying a graph grammar to L-Systems which meant that rewrite rules become context sensitive to surrounding symbols. He then went on to prove that his *G2L-Systems* representation could encode any form of graph.

Kitano used rewrite grammars to produce graph representations by utilising a connectivity matrix representation [80]. It involved rewriting elements of a connectivity matrix that describe the topology of an artificial neural network (ANN). A rewrite rule is represented as a symbol with a respective $2 \times 2$ matrix. The matrices contain elements that are either further symbols or some form of primitive. The rewrite rules are applied, creating an increasingly larger matrix, until primitives have replaced all symbols. The result is a connectivity matrix for an ANN where Boolean primitives allow for the
encoding of network topology whilst real valued primitives can express weighting on those connections. The size of the ANN is dependent upon the size of the matrix that is in turn dependent upon the number of rewrites required to remove all symbols. It has not been made clear if this approach can express all possible network topologies, unlike G2L-Systems [92].

Cellular Encoding [59] enables the growth of ANN’s using a rewrite grammar in which individual symbols represent neurons rather than the connections between them. This work has been extended in numerous ways, such as to include the evolutionary optimisation of the rewrite grammars themselves, incorporating the representation of weights and biases of neurons into the rewrite grammar itself [60], and to use edges rather than nodes as the basis for the grammar [97]. This is one of the earliest attempts, utilising a developmental principle, that could produce networks of neurons sufficiently large enough to tackle non-trivial problems. Generally, cellular encoding has a much greater computational expense than optimising a directly encoded ANN, but has been shown to be able to find optimal structure for problems without requiring any predetermined domain knowledge [61].

### 2.4.2 Cell Development Models

Moving away from the rewrite grammar approach are the systems based on cell models. These are systems that represent each cell as its own independent entity with its own state, but with the same set of developmental rules as all other cells in the system. This is a more refined approximation of biological development than that of rewrite grammars.

The utilisation of a grid-based environment, where cells occupy elements of the grid, is a popular approach to modelling spatial development. It automatically defines a fixed neighbourhood topology and is simple to represent as a data structure. These can be considered as extensions to the Cellular Automaton (CA) [136] which also utilise a regular
grid of cells, where each cell is in one of a finite number of states. The grid can be of various dimensionalities. The model is discrete in terms of both cell states and time, such that a cell’s next state is dependent upon some set of rules applied to the cell, neighbours and/or previous states.

The differences between such a cell model and CA are in the addition of growth of cells rather than the fixed cell numbers used in a CA, and also the ability of a cell to interact with cells outside of its neighbourhood topology.

The process of updating the states of cells (elements) across a grid can be generalised using a large scale chaos approach by means of a coupled map lattice representation [77]. The coupled map lattice represents a dynamical system with discrete time and space, and a continuous state. It allows the modelling of physical phenomena such as convection, reaction and diffusion whereby the state of each new element is determined by the previous state of the element and its neighbours based on the dynamics of the phenomena being modelled.

Kitano [81] used and extended this approach with a globally coupled map to allow the elements new state to be dependent not only upon previous and neighbouring states but also on global variables such as time, and average energy of the systems. He called this a *super coupled map* and results suggested that there was a specific region in the window of parameters within which elements would form self organized clusters of various guises.

However, there are significant problems attached to such grid structures. When a growth process is introduced through cell division it becomes awkward if all neighbouring elements of a dividing cell are already occupied. The most common solution is to allow cell overwriting in which the existing cell is simply over-written if a new cell is to be placed in that location on the grid [22, 102].

Dellaert and Beer [41] came up with a novel solution to this in which the entire space is fully occupied at all times. Figure 2.5 shows an example of how they used space
Figure 2.5: Two ways to represent cell division in the grid based spatial developmental process.

partitioning to handle cell division without needing to worry about cell-overwriting, whilst still retaining a fixed neighbourhood structure. This is compared to cell division for cells of fixed size where the division of a cell into an already occupied area has to be accounted for.

One common approximation, which is made amongst many simulations of cell development, is their representation in 2-dimensional space. Agarwal [1] believed this to be a problem, highlighting that many biological development processes occur, and can only be described, in 3-dimensional space. Why the penchant for 2-dimensional representations exists may be for numerous reasons:

- Typically, simple cell development experiments are preformed on Petri-dish apparatus.

- Visualisation of cell structures resulting from computer simulations are much easier to produce in 2-dimensions.

- The extension to higher dimensionality may massively increase computational costs of simulation.
There is no doubt that nature is massively parallel and this includes the interaction and function of cells. Modelling such process in an inherently serial machine will always cause problems. The simple option is to update the state of each cell in serial [1]. This often leads to the appearance of serial artefacts in the developing cells. In the best case these can appear as simple oddities in the division of cells such as a bias towards cells that are processed earlier or later in the growth process. In the worst case a single cell can divide continuously through the serial process causing the interaction between cell state and environment to be quashed. A better approach is to simulate the parallel update of cell states. This requires extra overheads since each cell state must be stored twice, one for the current state and another for the new state.

There has been some impressively thorough and biologically plausible, yet computationally feasible, cell development models formulated. Fleischer [47] produced a model which was capable of describing self-organising 3-dimensional cell structures with interactions of both a physical and chemical nature. He used an evolutionary approach to find parameter settings for his model that produced suitable results. Kumar [87, 88, 90] created the evolutionary development system (EDS) in which he recognises that the cell structure is not the only essential element of a cell development model and the proteins and their interactions with genes and the cell are key too. The EDS is capable of representing proteins that can synthesise, interact and decay.

However, aside from improving our understanding of natural systems, biological plausibility may not be the best approach to creating developmental evolutionary systems that can construct useful solutions. For example, Miller et al [104, 101, 102] concentrate on producing characteristics which are highly sought after by engineers such as self-maintenance, repair and regeneration without any justification for biological plausibility.
2.4.3 Developmental Artificial Neural Networks

The simulated evolution of artificial networks (EANNs) has become a research field in its own right [140] and, as an extension, the simulation of biological developmental growth of ANNs has been the subject of much research [92, 33] for a number of possible reasons.

- The design of complex ANNs by hand is difficult and more optimal or promising architectures may be missed. However the evolution of large complex networks of real neurons is as equally difficult and so it is hoped that taking inspiration from natural developmental processes may result in a more scalable system for the evolutionary discovery of ANN structures [7].

- The addition of a growth process to EANNs may provide a more biologically plausible way of describing the architectural structure of neurons in the brain [118].

- It may introduce modularity or compartmentalise the networks to improve their ability to generalise and so produce ANNs which are exceptionally good at dealing with complex, noisy or deceptive data [50].

- It may provide the ability to form genetic representations which allow crossover to be more effective rather than destructive, as is often the case with a direct encoding, allowing the useful exchange of information between individuals [140].

- It has long been recognised that development plays a large role in the structure and function of the brain. Neurons are particularly unusual types of cells in that their life cycle is very different to that of other tissue cells. Modelling the effects of developmental growth may give us insights into how specific areas of the brain form and how in turn that affects functional and learning capabilities [32, 34].

With the exception of grammar based approaches such as cellular encoding discussed in section 2.4.1, most approaches to the growth of ANNs are based on cell development
models. This usually consists of a grid based environment with provision for neuritic growth to support axonal and dendritic structures [74, 7]. However other methods have included fixing the neuron positions and adapting only the neuritic growth [118].

Research into the timing and movement of neuron positions during the development of the brain has shown to be critical in the final functionality of neuronal structures in the brain. Some work has attempted to incorporate this into models of neural development by enabling cell adhesion chemicals [32] and physical forces to play a role.

There has been some limited success with these approaches, primarily in the area of evolving simple control systems for autonomous agents [41].

2.4.4 Classifying Models of Evolutionary Biological Development

As is evident in the previous sections, there are many approaches to modelling biological developmental processes in an evolutionary system. Some of these differ in fundamental ways and there are arguments about which are the more powerful, expressive or biologically plausible.

Bentley and Kumar [12, 18, 89], who introduced the term ‘Computational Embryogeny’, categorised such models into three approaches and discussed and compared each:

- **External Embryogenies** refer to systems that use a pre-defined growth process that is dependent upon parameters stored in the genotype in order to grow the resulting phenotype. However the process itself cannot be evolved since it is external to the genotype and so outside of the evolutionary process.

- **Explicit Embryogenies** applies to systems where the growth process is explicitly defined in the genome for each stage of growth. Since the rules applied at each stage of the growth process are explicitly defined, these rules form a tree like structure.
Using a GP like approach to evolution is therefore most common and this allows for the rules themselves to be evolved.

- **Implicit Embryogenies** are the most similar to that observed in real biological systems. The growth process is implicitly defined by the set of rules that make up the genome. How these rules interact to form an embryogeny is dependent upon the state of the system upon which they are being applied.

Each approach was tested against a tile tessellation problem where a tile consisted of the layout of components on a grid and fitness was measured as the number of grid elements left empty when the evolved tile is tessellated. Results showed that the implicit embryogeny was by far the most successful with exceptional scalability in terms of the number of elements of which an evolved tile consists.

With some further analysis it is possible to pinpoint the differences in the structure of the dynamics in each case as shown in figure 2.6. For example, in an implicit mapping approach, the phenotype is dependent upon the mapping process that in turn is dependent upon the genotype. However, the genotype is also dependent upon the state of the phenotype and so this encapsulates the genome, mapping process and the phenotype within a complex non-linear dynamical system. This means that not only can the mapping process itself evolve, but also it can evolve to be adaptive in its application to the phenotype. In an explicit approach the genotype is no longer dependent upon the state of the phenotype. Thus the loop between genotype and phenotype is broken. This allows the mapping process to evolve but without the ability to be adaptive during application.

With an external approach, neither the genotype nor the phenotype interacts in a loop with the mapping process and so is held outside of the evolutionary process preventing the mapping process itself from being evolved.

Since the mapping process effectively defines the representation used to transform the genotype into a phenotype, an external embryogeny results in a fixed representation,
explicit embryogeny allows the representation to evolve, and implicit embryogeny allows the representation to evolve but also allows it to be adaptive in its application.

Implicit embryogeny has certainly exhibited much greater potential for evolving scalable solutions to problems [12], but it has not been explored as intensively as other forms of embryogeny. This may be, in part, due to the discontinuity it introduces into the landscape of search whereby neighbouring genotypes can result in very different phenotypes. These noisy and discontinuous search spaces are very difficult for any search process.

Stanley and Miikkulainen [124] suggests a multidimensional classification based on ‘5 dimensions of variability’. These are:

- **Cell Fate** refers to the role a cell will take in the adult phenotype and can be
defined by a single or several processes.

- **Targeting** refers to the ability to locate and control the connectivity of a cell. At one extreme the targeting is genetically pre-defined, at the other it is determined by several processes.

- **Heterochrony** is the change in the timing of a developmental process. Some systems have no way of altering these timings whilst others have many.

- **Canalization** is the sensitivity of the genotype to mutations.

- **Complexification** refers to the way in which the complexity of the genetic representation can be varied.

The paper then goes on to categorise the majority of influential papers in the field by this method and concludes that there does not seem to be any correlation between either grammatical or cell model approaches in terms of the distribution within these 5 dimensions. Although it provides some useful insight by comparing such a large number of embryogeny models in a structured manner, in comparison to Bentley and Kumar’s work, this form of classification seems more contrived and weaker in its justification.

### 2.4.5 Models of Genetic Regulatory Networks

Genetic Regulatory Networks (GRN’s) are a fundamental feature of natural developmental processes, and we are just beginning to unravel how these networks of genes interact with evolution. In essence, a genetic regulatory network is a complex protein factory. It produces proteins that can interact with and alter the surrounding environment, but it is also dependent upon proteins for its function. In this manner, the GRN forms a bond between the genotypic and phenotypic representations.

A common reason for modelling a GRN in a computational system is to enable modularity in the mapping process from genotype to phenotype. Bongard [20, 21] created
an *Artificial Ontogeny* system which evolved virtual embodied agents and argued that by using a GRN as the basis for representation it enabled more modular agents to evolve and with greater complexity. Many other researchers have taken a similar approach of combining simulated evolution with genetic regulatory network models, commonly concentrating on the development of neural network robot controllers [74].

Artificial regulatory network models have shown how simple gene interactions through protein concentrations are capable of function approximations such as the sinusoidal, exponential and sigmoid functions [91].

Proteins are large polymer molecules that are made from long chains of amino acids. These chains fold up into a complex 3-dimensional shape that differs from protein to protein. It is this shape and how it interacts with others which defines the proteins functionality [26].

In order to represent the numerous and complex interacting structures of proteins in nature, Bentley has suggested modelling genetic regulatory networks using protein representations based on fractals, or more specifically a subset of the Mandelbrot set [14, 13, 15, 16]. The idea is to represent a protein using three real values that record the position and size of the sample from the Mandelbrot set. This enables a large array of proteins with varying similarities and differences to be produced using only three real values.