An Investigation of the Importance of Mechanisms and Parameters in a Multicellular Developmental System

Tüze Kuyucu, Martin A. Trefzer, Julian F. Miller, Andy M. Tyrrell

***Abstract*—Multicellular organisms in biology possess invaluable characteristics, which artificial systems in engineering lack; such as adaptivity and robustness. Modelling biologically inspired multicellular developmental systems in evolutionary computation have been considered for a number of years, and there exist many developmental simulations that capture multicellular characteristics of biological organisms. However the relative importance of many mechanisms in such models is still poorly understood. This paper undertakes a detailed investigation of the importance of many mechanisms and parameters on the organisational behaviour of a gene regulatory network based artificial developmental system via classical pattern matching experiments. The work leads to an improved understanding of artificial multicellular development, which will assist in its utilization in the application of evolutionary computation. It may also provide a better understanding of mechanisms of biological development.**

***Index Terms*—Artificial Development, Gene Regulatory Network, Computational Evolution, Evolvability**

I. Introduction

THE process of designing an effective developmental model for Evolutionary Computation (EC) often reduces to implementing and sculpting the right biological mechanisms in an effective way. This aims to achieve an evolvable developmental system while maintaining a system that is not constrained or overwhelmed by undesirable biological processes. But how can we know which processes are useful and which are not? Implementations of artificial development in EC has been proposed since early 90's; e.g. [1], [2]. But most of the developmental models designed still rely on educated guesses, and various assumptions on the suitability of the biological developmental processes for EC applications. The need to investigate the behaviour and effective ways of implementing artificial development is already acknowledged by various researchers [3]–[7]. The physics of the digital medium where the artificial development is modelled is different from the biological counterpart. A biological cell has many advantages and disadvantages when compared with a digital cell, e.g. the ability to copy the genotype perfectly from one cell to another is a given in digital systems whereas small errors are inevitable in biological organisms. On the other hand, the cell growth in biology is naturally allowed by the physics of the organism whereas this has to be engineered in digital systems. Hence when implemented artificially, the usefulness of a mechanism can not always be directly correlated to biology.

The authors are with the Department of Electronics, University of York, UK. e-mail: tuze_kuyucu@hotmail.com, {mt540, jfm7, amt}@ohm.york.ac.uk

The aim of this paper is to undertake a thorough investigation of the effects of certain developmental parameters and mechanisms–*cell signalling, use of chemicals, methods of mapping development to a phenotype*–when implemented in an Artificial Developmental System (ADS), and demonstrate that implementations that rely on educated guesses and assumptions often do not provide the optimal designs. Although a large amount of experiments are undertaken, the investigations only examine part of a much larger space of possible developmental systems. The experiments undertaken test the ability of an ADS to achieve various organisational patterns. Hence, the experimental test set consists of only stable patterns also used in [8]. Although a more diverse experimental test set including computational problems other than stable patterns–such as neural nets and control problems–would provide a better insight into understanding the effects of the explored developmental parameters and mechanisms, and a better validity to the results obtained, such diversity goes beyond the scope of this paper. This paper solely aims to investigate the effects of certain developmental parameters and mechanisms on the multicellular organisational properties of a particular ADS. This paper aims to investigate the effects of certain developmental parameters and mechanisms on the multicellular organisational properties of a particular ADS.The intention is that such a detailed investigation will inform researchers about the relative importance of developmental mechanisms and parameter choices.

The ADS used in this paper is an already existing model that uses a cell chemistry approach to implement development [9]. Hence, the conclusions on the effects of the investigated parameters and mechanisms apply particularly to similar models of development that implement multicellular development at the cell-interactions level; this will be explained further in Section II as "micro-model", which provides a brief background on development. Section III provides a review of some of the relevant ADSs in literature, and provides discussions on some of the various decisions different researchers have taken while designing their systems. Sections IV introduces the ADS used in this paper as it was designed the first time it was introduced in [9]. Section V details the evolutionary algorithm used to optimize the ADS in the experiments presented in this paper. Section VI, presents investigations for a series of developmental mechanisms and parameters, where these parameters are tested for their effects on the evolvability of the developmental system to model various patterns. Finally, the paper is concluded with Section VII. Evolvability is

understood as the success rate of an evolutionary algorithm when optimizing for a certain task.

II. Background

"The development of multicellular organisms from a single cell–the fertilized egg–is a brilliant triumph of evolution" notes Wolpert in his book, Principles of Development [10]. The structure of a single cell may arguably be the most complex part of any organism, but a unicellular organism is vastly limited in the tasks it can achieve and is vulnerable to environmental threats. A multicellular organism is capable of multi-tasking using division of labour amongst the cells, and it is able to protect itself from environmental threats better than a unicellular organism would, since the loss of a cell or few cells does not necessarily harm the organism. Although multicellularity could have arisen through cell division failure or chance mutation in the evolutionary history of organisms, multicellularity is a key approach harnessed by biology to create complex and intelligent organisms capable of executing sophisticated behaviours and surviving harsh and changing environmental conditions [11]. Multicellular organisms are a product of a process called development that builds these organisms from a single cell. Wolpert defines development as "The process of gene activity that directs a sequence of cellular events in an organism which brings about the profound changes that occur to the organism" [10]. In other words, a simple description of development is that it is the step where all the genetic information gathered over the evolutionary history of an organism is put to use to create an adult organism from a single cell. However, development is also a mechanism that maintains the stability and functionality of an organism throughout its lifetime, and not merely the genotype-phenotype encoding mechanism of biology for creating complex multicellular organisms. Thanks to multicellular development, an organism is capable of surviving damage and loss of its physical parts, which otherwise would be lethal to the organism.

Multicellular biological organisms have been a topic of interest in the computer science and engineering fields as inspiration of models of intelligent systems. Their ability to be robust, adaptive, and scalable while they develop, make them interesting in computational intelligence as these properties are difficult to design in traditional approaches to computation or engineering. Biological organisms can grow from a single cell into a multicellular organism using the same genotype for all cells. These cells can then specialize to form different parts of an organism. Although the process of development in biology is clearly defined, its definition in Evolutionary Computation (EC) can be different depending on the author [3], [8], [12]–[18]. While some artificial algorithms try to closely model the biological development, others are simply inspired by a mechanism of biological development. In the latter cases, artificial development is defined by the source of its main inspiration and the task it is used for. Roggen's diffusion based developmental model [19], the self modifying cartesian genetic programming by Harding et al. [14], and Stanley's pattern producing networks [16] are examples of systems that use simple inspirations from few biological developmental mechanisms. Although most commonly referred to as artificial development, its name can also take many other forms; computational embryology, artificial embryology, artificial embryogeny, artificial ontogeny, computational development [12]. Formulating a correct definition of *artificial development* is not a simple task, and this is not the intention of this paper. In the remainder of this section this paper will refer to *artificial development* as a digital system that models the biological developmental process for the uses of understanding biology and/or aiding EC. Going back to Wolpert's definition of biological development [10], development in this paper will refer to the formation and maintenance of multicellularity in an organism. After a review of the benefits of multicellularity in EC, the known ways of modelling development in an artificial environment will be discussed.

A. Benefits of Multicellular Development

A multicellular design approach in EC benefiting from the decentralized organizational mechanism achieved in biological organisms could bring about scalability, fault tolerance and adaptivity to systems. These three possible benefits are discussed in further detail in this subsection.

1) Scalability: Over the years of EC research, the complexity of the evolved designs has not increased greatly. The inability of evolution to design systems at the desired level of complexity in a reasonable amount of time is a major problem. The ability to achieve higher complexity systems without a major increase in genotype size and within an acceptable time frame is referred to as *scalability*. Traditional system design methods in engineering and computer science build complex systems via the repeated use of functions or modules. Hence it has been acknowledged by many that introducing a mechanism that can achieve modular behaviour while evolving designs could relieve the scalability problem [20]–[27]. An example of the use of modules in EC is the Automatically Defined Functions (ADF) [23]; ADF introduces reuse of parts of the genetic code during evolution. This adds the concept of modularity to Genetic Programming (GP) [28] aiming to speed up evolution, and increase the achievable complexity. Koza showed that ADFs increase the evolutionary speed of GP. A similar modularity was introduced by Walker and Miller [27] for Cartesian Genetic Programming (CGP) [29] to speed up the evolution of more complex problems with CGP. It was shown that evolution of problems with modular CGP was much faster (20x in some cases), and scaled better for complex problems. The modularity in GP and CGP is done in a systematic way, where a modularity mechanism works in parallel with evolution to create modules from pieces of the evolved system which then can be reused by evolution. However, an explicitly defined mechanism that incorporates modularity into evolution can not solve the scalability problem. This is due to the presence of direct genotype[1]-phenotype[2] mapping. A direct genotype-phenotype encoding causes the

[1]Genetic information in a cell that is used to obtain a certain phenotype

[2]The physical form and characteristics of an organism; the target system in EC.

genotype to grow in proportion to the phenotype. This creates a large search space as the target system gets more complex. Hence a direct mapping from genotype to phenotype is less effective in designing complex systems [19], [20], [26], [30]–[33].

In nature, biological organisms achieve phenotypes specified by genes that are orders of magnitude smaller. An example of this is the human genome, which comprises approximately 30,000 genes, yet a human brain has roughly 10^{11} neurons [34], [35]. It is noteworthy that the number of distinct cell types in human body is only 256, and this number is as low as 13-15 for Hydra, which is a predatory animal with regenerative ability [36]. The extremely complex structural and behavioural architecture of biological organisms is not through intelligent design, but the heavy reuse of cells and genes. Biology achieves a highly scalable mapping via multicellularity and gene reuse; each cell has the same copy of genotype, and each gene in a given genotype may have different effects depending on when and where they are expressed. Also, the same set of genes are used over and over again in building phenotypic structures of similar characteristics, e.g. limbs in animals. Taking inspiration from biology, the idea of multicellularity in achieving complex systems in EC has been implemented by many researchers. Although the ability of artificial development to be scalable has been demonstrated by simple experiments [13], [19], [30], [31], [37], successful use of development in the design of systems at desired complexities that tackle real world problems is yet to be achieved. A recent achievement of an ADS has been presented by Harding et al. who evolved a Self-modifying Cartesian Genetic Programming that can find general solutions to parity and binary addition [38].

2) Fault Tolerance: Biological organisms are robust creatures that can achieve a very high level of fault tolerance. Regenerative ability of plants is an excellent example of fault tolerance and recovery in biological organisms. Plants cells are classified as totipotent[3], i.e. under the right conditions any plant cell would theoretically be able to grow into a fully developed adult plant [39]. Another example of a simple organism that has amazing regenerative properties is the Hydra, which has the ability to regenerate even when cut in half, producing two hydrae [40]. One of the main reasons for the type of repair and regeneration that happens in biological organisms is because of the lack of a central control mechanism. The ability of multiple cells to coordinate and organize themselves using various communication mechanisms provide an emergent adaptivity and fault tolerance to the whole organism.

Fault tolerant systems in electronics and computer science are highly important for remote, safety critical and hazardous applications. Almost all of the widely used techniques (N-modular redundancy [NMR] being the most popular) in achieving fault tolerance in electronics require a central control mechanism or a "golden" memory which is assumed to be failure-proof [41]. A de-centralized multicellular architecture can provide the system designed with redundancy, allowing the destruction of a number of cells before failure. In a multicellular design all cells are essentially identical to one another, hence a cell has the potential to change specialization and replace a damaged cell in order to recover from faults. This multiple redundant behaviour in development can be used to create a system free of a single point of failure if each cellular structure is represented by an independent piece of hardware; hence removing the weakest link present in traditional redundancy designs.

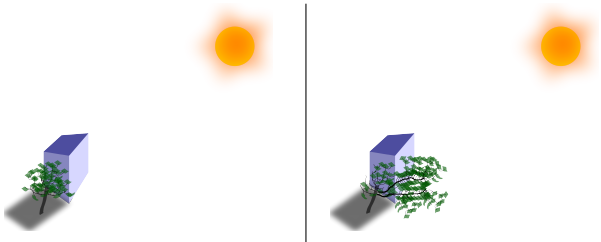
In addition to cell redundancy, biological organisms also have functional redundancy in their genetic code. In biology this functional redundancy arising from different genetic codes is referred to as *degeneracy*. Edelman and Gally [42] define degeneracy as; "the ability of elements that are structurally different to perform the same function or yield the same output", and they also note that degeneracy "is a well known characteristic of the genetic code and immune systems." Edelman and Gally emphasize that degeneracy is a key mechanism for the robustness of complex mechanisms and that it is almost directly related with complexity. In biological organisms degeneracy is present at almost every functional level; from genes to high level behaviours like body movements and social behaviours [42].

A developmental model can provide degeneracy both at the genotypic and phenotypic levels. Due to the indirect mapping of genes to the target phenotype, a developmental system can have multiple genes that perform the same function. Depending on their location in the organism each cell would have a different gene activity, but some of these cells would still have the same phenotypic functionality. Degeneracy in a developmental system can provide a powerful fault tolerance mechanism, as it provides robustness to genetic perturbations. A well explained example of gene redundancy in biology is the control of platelet activation by collagen [43].

Artificial development has been shown to provide a smoother degradation to perturbed genetic code [44]; when the genetic code of an artificial organism is altered before mapping the genome to the respective phenotype, the damaged genome will provide a phenotype that shows a more "graceful" degradation for a developmental system in comparison to direct mapping of the genome. In fact in some cases it was shown that a small number of gene knock-outs did not affect the overall result of gene expression [45]. It has also been demonstrated that a developmental system may be able to recover from transient changes in the phenotype, despite sometimes not being explicitly trained to do so [18], [19], [45]–[47]. However in order to benefit from the fault recovery properties of development, the developmental mechanism needs to be continuously running even when a fully functional phenotype is reached. In other words, the developmental system needs to have reached an *attractor*[4] that represents the desired phenotype. The ability of a developmental system to keep its phenotype unchanged after it has reached the target phenotypic form (i.e. represent the target phenotype as an attractor) is termed stability. To achieve a target phenotype at an attractor

[3]The ability of a cell to grow and generate all the specialized parts of an organism

[4]An attractor is a state to which a dynamical system settles after a time.



(a) A tree facing a wall that obstructs the sun. Only a very small portion of the tree from the right can see the sun.

(b) The tree leans and grows rightwards in order to receive more sunlight.

Fig. 1. The tree depicted in subfigure (a) uses the information from its leaves to maximize the received sunlight by changing its growth pattern as shown in subfigure (b).

state of a developmental system (i.e. finding a stable system) can be a harder task than achieving the target phenotype at a *transient* state. Unfortunately, solutions that occur at transient states are of no use for fault tolerance or adaptivity.

3) Adaptivity: Adaptive behaviour of biological organisms is another attractive quality that is aimed to be captured in EC. Designing systems that change their structure to adapt to their environments is a very challenging task, especially when a lot of the environmental factors can vary unpredictably. Multicellular organisms achieve adaptivity smoothly, and they change their structures or behaviours to fit the given environment for maximum survival chances. An example of adaptivity in biology is a changing plant structure depending on sunlight: if a plant "discovers" that there is an obstruction in the way that blocks the sun, the plant will grow in a way to maximise sunlight exposure, see Figure 1.

It is intended that by modelling multicellular development an adaptive system will be achieved. As mentioned in Section II-A2, in order to achieve an adaptive system the developmental system should be at a stable state when it achieves a functional phenotype. Once a stable state is achieved, the developmental process can run continuously in the background and adapt to environmental changes. A developmental system can have several attractors [17]; in an ideal case a dramatic change in the environment making the current configuration ineffective will cause the developmental system move to another attractor that would suit the current environmental conditions better.

Again, degeneracy in a developmental system can also allow the system to be more adaptive, because of the existence of multiple implementations of the same function. For example a change in an environmental condition may affect the activation of some genes in a developmental system, however the functionality of the organism would still be protected due to the existence of other genes that serve the same purpose as the affected genes. Systems with high degree of degeneracy have been observed to be very adaptable in biology as well, and favoured by natural selection [42].

B. Models of Artificial Development

In recent literature, developmental systems have sometimes been classified into two categories as: *Grammatical* or *Cell Chemistry* developmental models [6], [8], [12], [46]. However, there are developmental models that do not fit either of these two recent classifications, e.g. in [4] the developmental system is designed in a way to function without the use of cell chemistry but the developmental model is far from a grammatical implementation. The reason for classifying developmental models into *Grammatical* and *Cell Chemistry* is because the grammatical models of development follow a high level abstraction of biology, whereas the models that involve cell chemistry follow a low-level abstraction of biological development, and these two models of development cover most of the ADSs present in the literature. In order to make a similar but slightly clearer distinction amongst the present developmental models, the developmental systems are categorized into two different classes in this paper.

*1) **Macro-model Developmental Systems**:* A macro-model developmental system models the biological development at a high abstraction level, considering the overall behaviour of a biological organism or a developmental mechanism. A macro-model system's implementation is largely different to its biological inspiration, since the aim is to model the characteristic behaviour of the target developmental system/mechanism. Simply put a macro-model developmental system does not model individual cells in a multi-cellular organism, but provides a developmental behaviour in the system by the inclusion of time and ability to self modify over time. A widely known example of a macro-model developmental system is the Lindenmayer Systems (L-Systems) [48]. L-Systems, a parallel rewriting system, was introduced for modelling the growth processes of plant development [48]. L-systems model plant development using a set of rules via a grammar implementation, thus aiming to imitate biological development of plants using recursive functions. L-systems have been applied to circuit design problems [32], [49], neural networks [50], and 3D morphology design [51], [52]. Another example of a grammatical developmental system is Cellular Encoding (CE) [53]. CE was designed to be used in the design of neural networks. Using CE, a neural net would learn recurrence and solve large parity problems such as a 51-bit parity problem [53].

An example of a non-grammatical macro-model developmental system is the self-modifying Cartesian Genetic Programming (CGP), which models a CGP system that could alter its own structure over time after the evolution phase is complete [14], [38]. A macro-model developmental system should be computationally more efficient when compared to a micro-model developmental system in modelling developmental behaviour.

*2) **Micro-model Developmental Systems**:* A micro-model developmental system is a lower level model of the biological development that uses a bottom-up approach to modelling development. This category of developmental systems can also be seen as the more biologically plausible implementations, which imitate the biological development at a cellular level. Hence a micro-model developmental system involves the modelling of individual cells and their interactions, which together make up a whole organism. Each cell in a micro-

model developmental system has the same genotype and inter-cellular communication allows cells to specialize. All these cells together would form an organism which is the end product of development after each *developmental step*[5].

Although more biologically inspired, a micro-model developmental system does not necessarily model biological development perfectly. In fact there is much work in this type of artificial development with diverse design constraints, those that model biology closely [1], [13], [54]–[56], those that aim to model biological development in a simplistic fashion [19], [57], [58], and models that are "in-between" [3], [4], [9], [31], [59]. Mimicking biology closely should provide a developmental system with high evolvability [30], [60], whereas a simplistic model would reduce the number of complicated processes that exist in biological development, reducing simulation times drastically. The first and one of the simplest examples to micro-model developmental system is Cellular Automata (CA) [58]. CAs model biological systems with a grid of cells that determine their states using the local information from their neighbours and a global rule; this way, CAs effectively model inter-cellular communication and cell specialization.

III. Micro-model Deveopmental Systems

The ADS used in this paper is a micro-model, using a bottom up approach in simulating biological development. Before describing the model and the design approach for the model, micro-model developmental systems are described in further detail in this section. Furthermore, some of the other micro-model developmental systems used in EC are also discussed, Table I summarizes these models.

Development of an organism is an ongoing process existing throughout the whole lifetime of the organism, even when a perfectly functional organism is present. The developmental system in an organism is supported by various processes; these processes keep the system stable in an ever changing and noisy environment. The core component of a biological developmental system is the *Gene Regulatory Network (GRN)*, Section III-A. GRN provides the control of a single cell, and the single cell is integrated into a multicellular environment via *cell signalling*, Section III-B.The emergence of multicellular organisms from a single cell is possible through the *growth and cell division* processes, Section III-C. These mechanisms, which are important part of micro-model developmental systems, are described and their uses in ADSs are discussed along with methods of genotype-phenotype mapping (Section III-D) and diffusion in ADSs(Section III-E).

A. Gene Regulatory Network

A Gene Regulatory Network (GRN) is the heart of a biological developmental system. It is a computation network formed by the interactions of genes that are part of a DNA[6]. These gene interactions control the transcription of genes that determine the developmental path of a cell and the organism in general. Multiple or single celled, every organism has a functional GRN that can respond to the environment. In a multicellular organism all the cells have an identical genotype, but various cells appear different; this is due to different genes being expressed in different cell types. Genes interact with each other via proteins (type of chemical) that control the activation of genes, and the proteins are produced by the active genes. This creates a self regulatory network of genes, which are also affected by proteins that may be produced due to a reaction to an environmental change or proteins that come from other cells. Proteins can cause a range of reactions in a cell that can directly or indirectly determine the cell behaviour. Figure 2 provides a simple diagram describing how a GRN works. For an in depth description and discussion of how GRNs work, and how they piece together with evolution and development, see [61].

Similar to the different artificial implementations of biological development, artificial implementations of biological GRNs can have various levels of detail with or without certain biological GRN mechanisms; GRNs have been modelled using simple boolean rules [2], [62], as well as detailed simulations [45], [56].

When the aim of modelling a GRN is accurate simulation of a biological network, as much detail as possible in the model needs to be included. Often differential equations describing the dynamic system formed by the interactions of chemicals and genes are used for the latter case. Some researchers have also looked at using differential equations for modelling GRNs as part of an ADS in EC [55], [63]. These detailed and accurate models provide interesting insights into the benefits of continuous models, but because such models are computationally expensive, simpler models may be more successful in solving computational problems.

Simpler and more abstract models of GRNs are more appropriate for evolutionary optimization and design where the intention is to acquire prevalent properties of GRNs rather than model them in detail. Rule based GRN models are the most common implementations in EC; this is true for most of the models listed in Table I. This is due to the highly flexible implementation of the rule based GRN models that allow easy modification after the initial design phase.

GRNs can be seen as the "*controller*" of the cell, hence in artificial development it is possible to use any processing mechanism to replace the GRN core of a biological cell in the simulated environment. There are different cell controllers in the developmental systems present in literature ranging from combinational circuits to bio-inspired GRNs, see Table I. With current knowledge it is not possible to tell what type of cell controller is best suited for an ADS designed for EC. The advantages of evolving a combinational circuit controller is its ability to be implemented in hardware, and also the well established successful training mechanisms for small combinational circuits such as [29], [64], [65]. Evolution of Artificial Neural Nets (ANN) have also been well developed, and there exists many tools and mechanisms for the evolution of complex artificial neural networks, such as [66], [67]. Although both combinational circuits and ANNs are more

[5]The time it takes to carry out all the developmental processes once, is referred to as a developmental step.

[6]Deoxyribonucleic acid: a double helix structured nucleic acid that contains all the genetic information used in the development of all biological organisms.

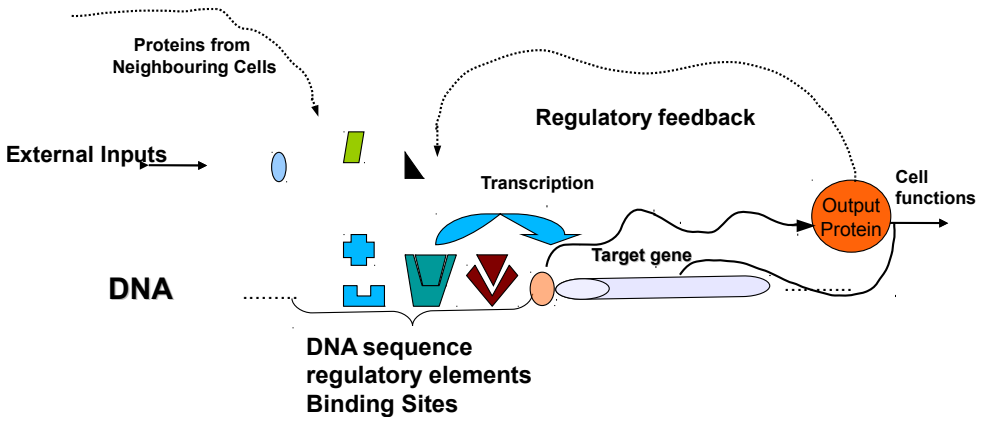


Fig. 2. A gene is activated by the correct matching of proteins that favour the transcription of the gene. The gene is transcribed when it becomes active producing a protein that may affect various functions of the cell it is produced in. Furthermore, the produced protein has the ability to bind genes within the cell to enhance or inhibit their activity or diffuse outside the cell to enter another cell. The resulting interactions of genes and proteins form a network of genes (GRN) creating a dynamical network. Hence, a gene's transcription is indirectly affected by the preconditional coding of the gene, activity of other genes, activity of genes in other cells, and environmental changes that may cause the production of further proteins.

widely used and established in EC, GRNs also have their advantages. GRNs (or GRN inspired computational models) are a common way of modelling a cell control mechanism within a developmental system. GRNs are highly dynamic computational networks that provide complex dynamics. A system with complex dynamics may benefit from the ability to change its state with the changing environmental conditions without the need to retrain, which can create an adaptive mechanism within the system. The highly dynamic nature of a GRN can provide multiple attractors for the same genotype allowing adaptation, differentiation, and robustness against disturbances and loss of functionality in genes [45]. Another advantage of a GRN based controller is that a GRN can be highly evolvable; a biologically inspired computational model may be more evolvable than a traditional computation model such as circuits. This is due to biological processes being a product of evolution itself [60].

In summary, different implementations of cell controllers aim to achieve similar objectives, and the choice of controller type is up to the researcher. In most of the models listed in Table I, the cell controller is a model of the biological GRNs. Although each of the GRN models use various implementation approaches, such as Differential Equations (DE) or rule-based formalisms, they all aim to create a dynamic network of self-regulatory genes. A numerical implementation of GRNs has advantages such as better precision in predictions (if modeled correctly). Whereas, a qualitative model like rule-based GRNs can incorporate more detailed and varied biological knowledge into the system. For a more detailed information on various ways of modelling GRNs see [68].

A developmental model that was not listed in Table I, but has been shown to provide successful results is the model designed by Roggen [19]. The system designed by Roggen lacks the use of any cell controller, and relies completely on the effects of growth and cell signalling (diffusion). In this model it was assumed that the cell controller is not needed for the developmental system designed. The system would always start with a group of pre-placed diffusers in a cellular grid, and the chemical diffusing from these diffusers would activate cells on contact. The state of each cell is determined by the concentration of the chemical present in the cell. Hence, there is a lack of gene processing unit in each cell, and the genotype may only represent the location of diffusers in the cell grid and the contents of a chemical level to function block translation table.

B. Cell Signalling

Cell signalling is an important part of development in multicellular organisms. It controls cellular activities and coordinates the cells in an organism. In biology, the cellular communication is established via the use of chemicals, which carry information depending on their type and intensity (concentration). In biology there are three main modes of signalling:

1) **Direct Contact:**
   This type of cell signalling only involves the cells in direct contact of each other. The signalling chemical does not diffuse from the cell producing it, but travels to the cell in direct contact of the host cell; hence no other cells would be affected by the signals. Fagotto in [69] refers to contact signalling in cells as "emerging as a major mechanism of communication in developing tissues". Thus contact signalling is a crucial part of the initial stages of multicellular development in biology.
2) **Short Distance:**
   Short distance signalling involves the diffusion of chemicals from a cell to its neighbours. The diffusing chemicals degrade quickly limiting their effective area to a small neighbourhood of cells. Short distance signalling complements contact signalling mechanism in embryonic patterning [69].
3) **Long Distance:**
   Long distance signalling can also be important for a

multicellular organism in regulating the development and coordination of the overall organism. In animals, the endocrine system is the mechanism that establishes a long distance chemical signalling for cells. In plants auxins are thought to be inter-cellular messengers for long distance signalling. Through auxin signalling, cellular patterning, and meristem and vascular development are thought to be mediated [70].

Coordination of large number of cells via simple chemical diffusion is not possible, and a more complex system like the endocrine or auxin is needed. It is noted in [69] that; "unrestricted diffusion is often undesirable in embryos in which small ensembles of pluripotent cells are required to respond only to local signals for proper patterning, even if they express a large number of different surface receptors." However long distance communication of cells is a more complex mechanism than the short distance and direct contact signalling mechanisms, and its role in development is less understood in biology.

Direct contact signalling is the more common signalling mechanism in ADSs that model multicellular development: it can be observed in Table I that the listed models prefer direct contact signalling more often than short distance (diffusion) signalling. Models that use direct contact signalling only are: [2]–[4], [31], [71]. The majority of these models, [2], [4], [71], that only use direct contact signalling are developmental models with similar characteristics to cellular automata, which specifically model the developmental effects of neighbourhood communication. Models that only implement direct contact signalling, also do not need to model graded chemicals in their system. The lack of simulating chemicals in an artificial model creates a simpler design.

An advantage of using boolean interactions instead of simulating graded chemicals in a developmental system is the decrease in the processing time and complexity of the developmental model [4], [31]. Other examples of ADSs using boolean interactions include [2], [30], [71]. In fact [30] and [71] only use the local phenotype state to model regulation rather than using explicit communication strategies. On the other hand modelling chemicals in a developmental system provides the ability to have more precise interactions amongst genes and cells. All the other researchers listed in Table I made use of the graded chemicals in their systems. In the earlier models, Kitano and Jakobi used a complex model of response strategies that may have over-complicated the system [54], [55]. Eggenberger's work that followed Kitano and Jakobi succeeded in providing impressive results, and this model had provided a much simpler response strategy [13].

Modelling chemicals in an ADS allows the use of communication via chemical diffusion in the system. Chemical diffusion has been used successfully as an organizing mechanism on its own [19], thus it is a mechanism worth implementing as part of a developmental system. Most of the developmental models presented in Table I also use graded chemicals as part of their system, and achieve promising results. Steiner et al. [63], Jakobi [54], and Zhan et al. [72] only use diffusion based communication mechanisms to achieve multicellular coordination and differentiation in their models. Kitano, [55], mentions that diffusion in correct proportions is beneficial to achieving useful developmental behaviour. However, developmental systems that do not use chemicals have also been shown to demonstrate scalability and stability in EC [4], [31], [71].

In the experiments provided by Flann [8] direct contact signalling was the more effective signalling method when compared with diffusion in achieving various patterns. All the pattern types; mosaic, patch, and border patterns were successfully achieved using contact signalling only. Whereas diffusion signalling was only effective for some patch patterns and the border patterns [8]. However, diffusion may be useful for controlling the growth of an organism. A gradient of chemicals can be created via diffusion amongst the cells that can provide the necessary information for stopping growth of a developmental organism before the entire organism space is filled. Miller had noted that without the presence of graded chemicals, it would be "unlikely, if not impossible, to achieve solutions that grow and then stop growing that meet the target objective" [18].

The need for graded chemicals have been partially investigated by Haddow and Hoye [4] after an earlier attempt by Miller [18]. Miller later realized that his experiments used boolean regulation for chemicals rather than graded regulation, i.e. the developmental model under test did not make use of the graded chemicals. Haddow and Hoye [4] concluded that increasing the number of chemicals in the developmental model inversely affected the performance of the system in achieving target patterns.

The GRN model presented by Haddow and Hoye uses a unique approach to modelling GRNs: only the cell state of the neighbouring cells are used for inter-cellular communication, and the chemicals that are produced inside a cell as a result of gene transcription are only used within the cell for self-regulation of genes. Unlike most other GRN models that use graded chemicals, Haddow and Hoye's model does not represent proteins as chemicals but as genetic actions, and there is no chemical diffusion or any other form of communication involving chemicals. Hence graded chemicals are only used for representing information within a cell. Due to the minimalistic role of chemicals, they increase the complexity of the search space rather than providing finer tuned information processing in the GRN and the developmental system [4].

TABLE I

LIST OF SOME OF THE MICRO-MODEL ARTIFICIAL DEVELOPMENTAL SYSTEMS SPECIFICALLY DESIGNED FOR COMPUTATIONAL PROBLEMS WITH THE COMMON DEVELOPMENTAL MECHANISMS USED BY EACH MODEL. EVIDENTLY THIS DOES NOT COMPREHENSIVELY COVER OF ALL THE MAJOR ARTIFICIAL DEVELOPMENTAL SYSTEMS. THERE WERE SEVERAL CRITERIA IN DETERMINING WHICH DEVELOPMENTAL SYSTEMS TO INCLUDE: CHRONOLOGICAL SPREAD OF THE DESIGNS AND A SPREAD OF RESEARCHERS INVOLVED IN THE DESIGN OF THE DEVELOPMENTAL SYSTEM WERE PART OF THE CRITERIA. ANOTHER CRITERION WAS TO INCLUDE SYSTEMS SPECIFICALLY DESIGNED TO SOLVE COMPUTATIONAL PROBLEMS USING MULTICELLULAR DEVELOPMENT. SPACE WAS ALSO A LIMITING FACTOR AS WELL AS THE LISTED CRITERIA. THE DEVELOPMENTAL SYSTEMS LISTED ON THIS TABLE WILL BE REFERENCED THROUGHOUT THE REST OF THE PAPER.

Developmental Model	**Cell Controller**	**Communication Mechanisms**	**Growth/ Division**	**Cell Structure**	**Graded Chemical Use**	**Target Phenotype**	**Stable**	**Robustness**
Dellaert and Beer 1994 [2]	RBN	direct	YES	emergent	none	2D patterns (control)	none	none
Jakobi 1995 [54]	GRN	diffusion	YES	functonal proteins	present (robot control)	ANN	none	none
Kitano 1995 [55]	GRN	diffusion & direct	YES	protein concentration	present	ANN	none	none
Eggenberger 1997 [13]	GRN	diffusion & direct	YES	functonal proteins	present	3D patterns	none	none
Bentley and Kumar 1999 [30]	rules (CA)	direct & routed	YES	emergent	none	2D patterns	none	none
Miller 2003 [59]	CGP (Circuits)	diffusion & direct	YES	circuit output (emergent)	present	2D patterns	upto 10 steps	transient fault recovery
Tufte and Haddow 2003 [71]	rules (CA)	direct	YES	emergent	none	2D patterns on hardware	none	none
Federici 2004 [46]	ANN	diffusion & direct	YES	NN output (emergent)	present	2D patterns	upto 2 steps	transient fault recovery
Gordon 2005 [31]	GRN	direct	YES	protein concentration	none	circuits	none	none
Haddow and Hoye 2007 [4]	GRN	direct	YES	functional proteins	optional	3D patterns	upto 10 steps	none
Devert et al. 2007 [3]	ANN	direct	NO	NN output (emergent)	present	2D patterns	stability checked	transient fault recovery
Steiner et al. 2007 [7], [63]	GRN	directable diffusion	YES	functional proteins	present	3D pattern	stability checked	resistance to unwanted mutations
Zhan et al. 2008 [72], [73]	GRN	controlled diffusion	YES	protein concentrations	present	Circuits	stable protein levels	transient recovery

C. Growth/Cell Division

Growth and cell division in biological development is a key process that enables a single cell organism to transform into a multicellular organism. Cell division creates a copy of a single cell with both the parent and daughter cells sharing the exact copy of the genome. The number of cells in an organism grows due to cell division. Growth is the change in the overall size of the organism due to multiplying numbers of cells as well as physical growth in the size of individual cells. An organism may grow to a predefined size that is mostly determined by the genotype e.g. animals, or its size may be highly dependent on the environmental conditions e.g. ivy. Hence depending on the nature of the organism and the environment, the growth and cell division processes may never stop.

All the ADSs listed in Table I feature some form of growth and cell division except the model designed by Devert et al [3]. Cell division and growth in ADSs is a way of controlling the number of cells active in the organism, and it provides the opportunity for a system to expand when extra resources are provided, hence adapting to its new environment. However, growth and cell division processes add extra complexity to the developmental system and they increase the simulation time and extend the developmental steps required to reach a mature organism. If the number of cells required for the target organism's size is known and is equal to the maximum number of cells available then eliminating the growth and cell division mechanisms from the developmental system may be beneficial.

D. Genotype-Phenotype Mappings

There are various implementations of mapping the developmental interactions to phenotype in the literature. Some systems (especially ones that do not use GRNs), interface the regulatory elements of the developmental system with the phenotype by using the phenotypic information as a regulatory element. In these models information from the phenotype is retrieved to be used as a regulatory input to the developmental system, and an output of the developmental system is used as a feedback to alter the phenotype directly. Dellaert and Beer use this approach in their Random Boolean Network (RBN) based developmental model [2], as well as Bentley and Kumar, Tufte and Haddow who use CAs in their developmental models [30], [71]. Miller's circuit based, and Federici's ANN based developmental models also interface phenotypic information directly to cell regulation [46], [59]. Haddow and Hoye also use this mechanism in their GRN based model [4].

In the remaining of the developmental systems listed in Table I, only artificial proteins/chemicals are used to define developmental environment, and only this information is used by the genes. Some of these systems use predefined proteins/chemicals as either structural or regulatory proteins, and others allow all proteins to perform both structural and regulatory functions. The system of Devert et al. is an exception to this, since it uses the chemicals from neighbours as inputs to a neural network in order to determine the phenotype and output chemicals of a cell [3]. In the rest of the approaches the structural proteins are interfaced to the phenotype in two different ways. They can either be used as functions that are part of the transcription process that make use of the genetic information in a gene to alter the phenotype [13], [54], [63] or they can be used as phenotypic outputs of the developmental system and mapped directly (or after some post processing, e.g. via a look up table) to the phenotype at the end of a developmental phase [31], [55], [72].

E. Diffusion

Table I lists various developmental models. More than half of these models implement some form of diffusion mechanism as a distant cell signalling mechanism. The only exception of a distant cell signalling mechanism that does not involve diffusion is the routed signalling implemented by Bentley and Kumar [30]. In this mechanism, two cells form direct connections with each other regardless of the distance between them. Not limiting the direct connections to the nearest neighbours, a symmetry breaking behaviour via a distant signalling mechanism is achieved. Since diffusion is a more popular and biologically inspired form of short-medium distance signalling, a simple diffusion mechanism is also modelled in the developmental system used here (same mechanism used in [59]); the mechanism involves constant diffusion of chemicals from each cell to their four cardinal neighbours.

The diffusion mechanisms implemented in the developmental systems displayed in Table I vary in their implementation, possibly providing different effects on the multicellular developmental system in each case. Zhan et al. [72] extends the model used by Miller [59] by the addition of a cell membrane and a chemical pathway. The cell membrane prevents the diffusion of chemicals below a certain threshold level, and the pathway provides a conversion table to convert the type of the chemicals to be diffused out to another chemical type. The cell membrane in [72] could be a good improvement since it provides a more controlled diffusion mechanism, and as it was discussed earlier; unrestricted diffusion in biological embryos is undesirable [69], which may also apply to ADSs. Steiner et al. [63] use a diffusion layer for the diffusing chemicals, the chemicals are diffused into the layer constantly but it is accompanied by cell adhesion and sorting which indirectly controls the direction of diffusion by moving cells within the organism.

Whether a more complex and bio-inspired diffusion mechanism is required for a more evolvable developmental system or a simple diffusion mechanism is sufficient is not clear without empirical data. Biological analogy could be used to suggest that uncontrolled diffusion is undesirable, but biological data can not always be relied on for models designed for EC. On the other hand most of the applications could be tackled by cell contact signalling [8], hence a diffusion mechanism may increase the ADS complexity and might not be beneficial at all.

IV. The Artificial Developmental System

As stated earlier, the ADS used in this paper is an already existing model that was designed to enable scalable evolution of digital systems such as circuits [9], [37], controllers [74],

and image compression tools [75]. The model that will be described in this section is the original design that was introduced in [9] without any current optimizations or simplifications. Therefore, the system described will have all the assumptions made in the initial design. Starting with an existing model that has been successfully used in previous works should demonstrate the available room for improvement in existing models, and the flaws of a design process that relies on assumptions and the existing literature. As noted earlier, the ADS used in this paper is a micro-model that uses an artificial GRN as the cell controller. The GRN is implemented as a rule based formalism.

Each gene is treated as a rule with a condition and a result (i.e. pre and post-conditions). A gene will be activated when the precondition is met, and when a gene is active its postcondition will be processed. The precondition of a gene specifies the proteins (and other chemicals) that need to be present or absent in order to activate the gene. Thus a protein (further detailed in Section IV-B) can inhibit, enhance or have no effect on the activation of a gene. The postconditional part of a gene specifies which protein to produce, and the action to be taken by the produced protein. In the model presented here some of the proteins have functional tasks, but all of the proteins participate in regulating the activation of genes. Hence, there is no difference in the regulatory or behavioural proteins as all proteins are regulatory. A short pseudo-code description of the simulation process of the GRN is detailed in Algorithm 1.

The model presented here makes use of graded chemical regulation of genes; all the proteins are treated as chemicals. Only direct contact and short distance signalling are used as cell signalling mechanisms in the ADS. Since there is still little understanding for the capabilities of ADSs in solving engineering and computer science applications, the described ADS is not expected to be capable of designing organisms complex enough to benefit from long distance signalling. As stated in Section III-B, long distance signalling is the least understood cell signalling mechanism in biology, and the least modelled communications in ADSs. Hence it is also convenient not to implement it as part of the ADS, which will result in a simpler design.

The direct contact mechanism implemented in the model here was inspired by the Plasmodesmata in plants. Plasmodesmata are microscopic channels that breach the cell walls of plant cells forming a tunnel between two cells, hence enabling the passage of chemicals between them [39]. There are two ways of creating plasmodesmata, they can either be formed during cell division or between two mature cells. As described in Section IV-B the direct contact signalling is implemented as a result of the synthesis of a plasmodesma protein, which can create a tunnel connecting the two cells. When a tunnel between two cells is established a free flow of chemicals between the two cells is allowed, hence the two cells share the same concentration of chemicals.

The other signalling mechanism implemented is the short (or medium) distance signalling, which is modelled as diffusion. The diffusion process is carried out for all chemicals available in every cell; half of an available chemical diffuses out of a cell equally to the four nearest neighbours, i.e. each neighbour obtains $\frac{1}{8}$ of the cell's chemicals. This diffusion model used is unrestricted diffusion, and although this model has been commonly used in developmental models before [47], [59], [72], [76], it may be undesirable in artificial development as well for similar reasons to biology. As mentioned earlier, in biology it is noted that an unrestricted constant diffusion is often undesirable [69].

The signalling mechanisms will be further investigated in Section VI to see whether there is a better model of these signalling mechanisms or whether they are needed at all. Change in chemical levels due to diffusion and direct contact signalling are adjusted at the end of a developmental step for all cells. Thus, the GRN always works with the chemical levels that are determined at the start of a developmental step; this aims to reduce the bias in the course of development due to the order of cell updates.

A. Gene Representation and Processing

The genotype is represented as a chain of binary strings, and each binary string represents an individual gene in the genome. A fixed length binary implementation of the genes was chosen to make the application of the GRN model straightforward on an embedded processor. Similar to the gene representation used by Gordon [31], each gene is modelled as a rule with two sections; pre and postconditional sections. The preconditional part of a rule specifies whether the activation of the rule could be enhanced, inhibited or unaffected by the presence of known chemicals. For a gene to be activated all the activating chemicals must be present and all the inhibiting chemicals must be absent. In other words a logic *AND* is used as the precondition function. Each chemical has a concentration, and it is considered to be present if its concentration is at or above a predetermined threshold level, otherwise it is considered absent. Chemical concentrations are represented as 8-bit integer values, hence the possible values are between 0 and 255. All of the processing in the ADS is done using binary and integer operations. The postcondition of a gene defines the chemical that is produced, and depending on the type of chemical produced, the information for the action it is going to take is also included within the postcondition. Encoding of an example gene is illustrated in Figure 3. More detail on the types of chemicals is provided in Section IV-B.

The GRN is processed synchronously, from the start of the genotype to the end. The GRN of each cell is also processed synchronously, starting from the top left cell and moving from one cell to another (left to right and top to bottom). This means that the results of a processed ADS are dependent on the order of genes in the genome as well as the order of cells in the organism. Although this diverts from biology, the processing of the ADS becomes straightforward. The GRN of each cell is processed every developmental time step which synchronises the organism. Every time step the developmental system progresses a step further, making it older. The changes to the chemical concentrations due to gene activations are done in real time, but the changes caused via cell signalling are recorded in a temporary buffer and made at the end of

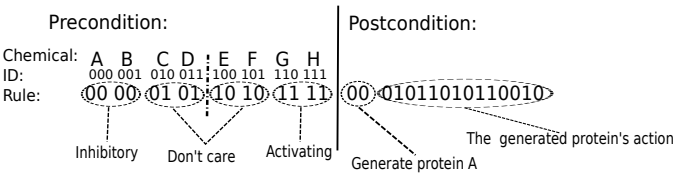


Fig. 3. An example gene of 32 bits is shown. The first 16 bits are reserved for the preconditional part, which specifies the rules to activate the gene. There are 8 chemicals defined in this figure; the first 4 being reserved for proteins, while the last 4 are messenger molecules, see Section IV-B. Each chemical's required presence or absence is specified by a 2 bit number, which provide two don't care states. In the event of a don't care state, the presence or absence of a chemical has no effect on the activation of the particular gene. The second 16 bits of the gene is reserved for the postconditional part, which provides the ID of the chemical produced as a 2 bit number (this means that only the first four chemicals [proteins] can be produced, i.e. the messenger molecules can not be produced via the activation of a gene), which is then followed by a 14 bit number. The last 14 bits in the gene define the action of the chemical produced if it has one (further explained in Subsection IV-B), if not the last 14 bits are treated as junk.

each developmental step. Whether a chemical will bind a gene is dependent on the concentration level of the chemical and the consumption rate of the chemical. In a cell (i), the concentration level (Z_{igtc}) of a chemical (c) by the time it reaches a gene (g) depends on its initial concentration (X_{itc}) at the start of the developmental step (t), the amount of the chemical used by the previous genes in the genome (B_{igtc}), and the concentration of chemical produced by the previous genes in the genome (O_{igtc}). At age 0 the concentration levels of all chemicals are 0, and the consumption rate (R_{gc}) of a chemical is a constant (either hard-coded or evolved) that specifies the amount of a chemical consumed if the chemical binds a gene.

$$Z_{igtc} = X_{itc} + O_{igtc} - B_{igtc}, \tag{1}$$

Whether a chemical will bind a gene is a boolean operation (Y_{igtc}), and it depends on available chemical concentration, and chemical consumption rate.

$$Y_{igtc} = \begin{cases} true & \frac{Z_{igtc}}{R_{gc}} \geqslant 1 \\ false & \frac{Z_{igtc}}{R_{gc}} < 1 \end{cases}. \tag{2}$$

B. Protein Synthesis

Biological protein synthesis involves the process of a protein transcribing a gene (binding the promoter sequence to activate the gene), where the genetic information (postconditional region) is copied into a mRNA (messenger RNA[7]). This is followed by the mRNA strand moving out of the nucleus of a cell to meet a complimentary tRNA (transfer RNA). Each tRNA molecule compliments the opposing bases on the mRNA strand perfectly. These in turn have the amino acid sequence to successfully code for a particular amino acid which forms a protein. Hence the amino acids (aka peptides) from the tRNA and mRNA combine to form a polypeptide chain (proteins), and can be used in a variety of structures such as enzymes and hormones to carry out a cell function. Although protein

[7]Ribonucleic acid: a nucleotide chain that is transcribed from and similar to DNA, but has small structural differences.

Algorithm 1 The pseudo-code for the simulation of Gene Regulatory Network for one time step.

```
ORGANISM stores all the cells available for development up to MAXCELL;
CELL is an individual entity that models a biological cell
for each CELL cell in ORGANISM do
  if cell is ALIVE then
    for each GENE g in GENOME do
      if (promoter chemicals for g) > PROMOTETHRESHOLD then
        promote ← true
      else
        promote ← false
      end if
      if (inhibitor chemicals for g) > INHIBITTHRESHOLD then
        promote ← false
      end if
      if promote then
        comment: Process the postcondition of g
        proteinProduced ← protein type specified by postcondition of g
        increase proteinProduced in cell by PROTEINPRODUCTIONRATE
        comment: Check if proteinProduced has any functions to carry out
        if proteinProduced = PLASMODESMA then
          tunnelDirection ←information from g on the direction of tunnel
          tunnelOpen ←information from g whether to block or form a tunnel
          if cell's neighbour in direction tunnelDirection is not ALIVE then
            if tunnelOpen then
              grow a new cell in direction tunnelDirection and
              form a plasmodesmata between cell and the new cell
            end if
          end if
        else if proteinProduced = STRUCTURING then
          extract information from g to modify cell structure
        else if proteinProduced = SENSOR then
          extract information from g for monitoring the cell surroundings
        end if
      end if
    end for
  end if
end for
```

synthesis in biology covers a series of complex processes, simply put it is the phase of building proteins.

In the ADS described here, protein synthesis does not involve the simulation of RNA molecules. Once a gene is activated with the correct binding of proteins the postconditional part of the gene is processed, i.e. the gene is transcribed. The first part of the postconditional part of a gene specifies the ID of the protein to be produced. The protein produced is supplied into the cell at a predetermined protein production rate. Once the protein produced is added to the cell protein repository, the rest of the gene postcondition is processed depending on the type of protein produced. The production (synthesis) of a protein triggers a protein action in the cell. For this action, the protein produced uses the remaining information provided by the gene it was produced by to coordinate its actions. The chemical types (including proteins) and their actions are detailed within Section IV-C.

C. Chemicals

The GRN system implementation used in the ADS described here makes use of various chemicals that are used for gene regulation; affecting cellular functions, and communication. Chemicals are classified in two groups as *proteins* and *molecules*. The proteins build the organism, create a regulatory network, and are used for multicellular communication. The molecules are used as part of the regulatory system as well, and they help monitor the environmental changes for regulatory adaptation. The proposed GRN system has four types of proteins:

1) ***Plasmodesma Protein****:* Plasmodesma proteins are inspired by the *Plasmodesmata* in plants (similar to the gap junction in animals). When a plasmodesma protein is synthesised, it forms a tunnel in one of the four cardinal directions (North, South, East, West), and if the neighbouring cell also has formed a tunnel in the corresponding direction, the two tunnels join together allowing the passage of proteins between the two cells (the joint tunnels are termed plasmodesmata in this paper). After the two cells are connected by plasmodesmata (tunnels), they share the same protein distribution. If the neighbouring cell space (of the cell where the plasmodesma is produced) is not occupied, a cell division is initiated. During the cell division plasmodesmata is created connecting the two daughter cells (one of which now fills the empty cell space). Although in plants the plasmodesmata size is known to shrink as the cells mature (thus filtering larger proteins) [39], this is omitted in our developmental system for simplicity. However, the callose deposition is simulated. The callose deposition causes the blockage of the plasmodesmata, which prevents the transport of all proteins through the plasmodesmata. In our system this is initiated by a specifically encoded plasmodesma protein.

2) ***Structuring Protein****:* Structuring proteins are the type of proteins that build and change the physical structure of the cell. When a structuring protein is produced, it uses the information provided by the gene to alter the physical structure within the cell. The physical structure of a cell refers to the cell's role in the application domain, e.g. a component part of a digital circuit, a control system or an image pixel. When a structuring protein is produced, the cell phenotype is altered. For example, if the cell phenotype is represented as a 32-bit integer, a transcribed structuring protein may make bitwise changes in the 32-bit number using the postconditional information provided by the gene producing the structuring protein, see Figure 3. This provides an emergent structure for the cells in the developmental system, aiming to create an effective mapping of developmental dynamics to organism structure.

There is still little understanding on how to map the dynamics of a developmental system to engineering applications. Hence, it is not possible to distinguish between the different methods of processing and mapping the developmental information to a phenotype. Using specific proteins that can alter the phenotypic structure of the cell during the ongoing developmental process has the potential to extract more information from the dynamics of the ADS rather than using the protein concentration values at the end of a developmental stage. The evolvability of the latter two methods of phenotypic mapping in creating patterns will be briefly investigated in Section VI.

We reviewed various genotype-phenotype mappings used in micro-model developmental systems in Section III-D. However, in such developmental approaches it is still not known how best to map the dynamics of a developmental system to engineering applications. Hence, it is not possible to dis-

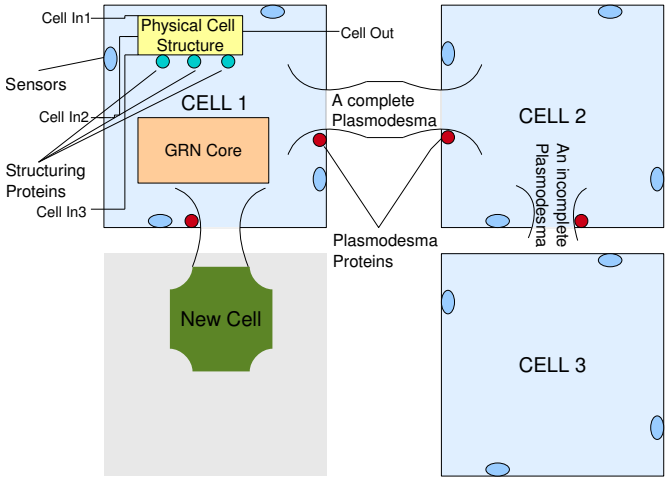


Fig. 4. In a multicellular environment using the 4 basic protein types a cell is able to: interact with its environment, grow, structure itself, and form a multicellular organism. The basic functions of the listed proteins are demonstrated in this figure. Only cell 1 is drawn completely, certain components are omitted in other cells. In the actual implementation of the organism there are no spaces between cells, they are only separated by their borders. In the example above, cells 1 and 2 both have active plasmodesma proteins, which cause the formation of a channel on both cells towards the other, creating a plasmodesmata to allow free movement of proteins from one cell to other. Cells 1 and 2 both also have active plasmodesma proteins on their southern sides. Cell 1's southern neighbour does not exist, so the active plasmodesma protein initiates a growth process in that direction. However, cell 2's southern neighbour is an alive cell with no plasmodesma protein, thus cell 2 forms an unconnected channel on its southern wall. The 4 sensors drawn monitor the outside activity on 4 sides of each cell and produce sensor proteins with the changing environment. The Structuring proteins are produced by the GRN to change the physical structure of the cell, which is connected to the physical inputs and outputs of the cell.

tinguish between the different methods of processing and mapping the developmental information to a phenotype.

*3) **Sensor Protein**:* The sensor proteins are produced by the GRN to act as sensors around the cell by monitoring the external environmental changes in the phenotype domain. The sensor proteins produce different kinds of messenger molecules for different outside activity. This enables environmental factors to affect the GRN activity, possibly creating a more interactive developmental system.

*4) **Regulatory Protein**:* Although every protein is a regulatory protein, there are proteins that only exist for regulatory purposes. These proteins, just like every other protein, control the GRN activity by their presence or absence, but they do not have any other purpose. Thus when a regulatory protein is produced by a gene, the information provided by the postcondition of the gene is discarded. The unused parts of the gene provide a neutral search space, which can have positive effects on evolvability. It has been shown that neutrality provides stability via preventing deleterious mutations, and helps evolution avoid local optima, thus creating a smoother search space [77], [78].

Apart from the four proteins introduced, there is another type of chemical called a messenger molecule; molecules have different regulatory properties compared to proteins. The *messenger molecules* are produced by the sensor proteins that monitor the phenotype domain when present. The messenger molecules can not be produced directly as a result of gene activity (protein synthesis) since their purpose is to regulate the gene activity via the environmental response, but they can still bind to activate or deactivate genes.

It is intended that these chemical types will give the cells the ability to form an interactive multicellular organism that can achieve scalability, adaptivity, and fault tolerance. Figure 4 portrays the role of these chemicals in a simple multicellular organism.

The developmental system presented here features cell division and growth as these are inspirational parts of biological development, and there are no known comparisons in literature on the effect of the presence or lack of cell division and growth.

Pseudo code summarizing and describing the implementation of the ADS is presented in Algorithm 2. This algorithm only details the functions that are executed during each developmental step. It refers to the gene regulatory activity in each cell as a function call to **GRN()**. The pseudo-code description of the gene regulatory network simulation in the organism is detailed in Algorithm 1. Lines 8–13 of Algorithm 2 detail the cell signalling procedures. When two cells are connected via tunnels (plasmodesmata) the concentrations of their chemicals are combined and shared equally to the two cells. This process is carried out by individual cells, and each cell only controls the tunnels to its east and south neighbours. Since the processing is done from top left most to the bottom right most cell, there is no need for each cell to check all four directions for the contact cell signalling process. Lines 15–18 detail the diffusion process where each chemical is constantly diffused out to four nearest neighbours. During the ongoing diffusion process, the changes to the chemical concentrations are not recorded on the actual values but stored in temporary variables. These variables are then later used at line 23 to update the actual chemical concentrations for each cell. This way the order of update of cells does not bias the diffusion process. By the end of the Algorithm 2 at line 30 the cell structure, which is a single or an array of integers, is translated for the target system. If the target system is a reconfigurable hardware device, the integer values are converted to a binary string to configure a part of the reconfigurable device. If the target system is an image, and the cell represents a pixel, then the integer value[8] is used to represent a pixel colour for the cell.

Algorithm 2 The pseudo-code for the simulation of artificial development for one time step.

```
1:  ORGANISM stores all the cells available for development
    up to MAXCELL
2:  CELL is an individual entity that models a biological cell
3:  call function GRN(); comment: see Algorithm 1
4:  for each CELL cell in ORGANISM do
5:    if cell is ALIVE then
6:      comment: Direct Contact signaling.
7:      comment: Only need to check EAST and SOUTH
        as the previous cells were to the NORTH and WEST
8:      if cell has a PLASMODESMATA to EAST then
9:        share chemicals with the neighbouring cell on the
          EAST
10:     end if
11:     if cell has a PLASMODESMATA to SOUTH then
12:       share chemicals with the neighbouring cell on the
          SOUTH
13:     end if
14:     comment: Diffusion
15:     for each CHEMICAL c in CHEMICALS do
16:       divide 1/2 of c to each neighbouring cell
17:       store changes to chemicals in a buffer available for
          each cell
18:     end for
19:   end if
20: end for
21: for each CELL cell in ORGANISM do
22:   comment: Update Cell States
23:   update the actual chemical values from the buffers
24:   reset the buffers to 0
25:   adjust the chemical values to be within 0 to
      MAX_CHEMICAL_LEVEL
26:   if cell was newly created then
27:     set cell state to ALIVE
28:   end if
29:   comment: Build Target Structure - problem dependent
30:   translate cell's structure for the target system
31: end for
```

[8]The integer value representing the cell function needs to be kept within 0 to max number of possible colours for the target image. A modulo operation is used to ensure this.

V. The Evolutionary Algorithm Framework

In the experiments described here, the resulting developmental systems are always evolved using an evolutionary algorithm. Evolution is used to find the genetic information (genome) that can represent the desired system after being fully developed. An Evolutionary Strategy (ES) with a population size of 7 and an elite size of 2 is used, i.e. ES(2+5). There is no crossover implemented but an adaptable mutation rate in the 0.5-10% range that changes with respect to the *rate of change* in the fitness, see Algorithm 3. The use of adaptive mutation rate in ES has been previously shown to boost the algorithm performance. More information and various investigations into self-adaptive evolutionary parameters such as mutation and crossover can be found in [79].

Algorithm 3 The pseudo-code for the variable mutation rate. This is part of a function that is called once every evolutionary generation.

comment: At the start of evolution convergence is set to 1 and Mutation Rate is set to 0.5%
if one of the top two individuals from the previous evolutionary generation is now the best individual **then**
  $convergence--$;
else
  $convergence++$;
end if
if $convergence = 0$ **then**
  comment: Mutated individuals are all worse; decrease the mutation rate
  if $MutationRate > 0$ **then**
    $MutationRate--$;
  end if
  $convergence \leftarrow 1$;
else if $convergence > 0$ **then**
  comment: Mutation did not hurt; increase the mutation rate
  $MutationRate++$;
  $convergence \leftarrow 1$;
end if
if $MutationRate < 0.5\%$ **then**
  $MutationRate \leftarrow 0.5\%$;
else if $MutationRate > 10\%$ **then**
  $MutationRate \leftarrow 10\%$;
end if

A. Selection

A refined fitness function is used for all the experiments discussed in this paper; further detailed in each experimental section. All fitness functions used in the experiments here work by penalizing the bad solutions. Hence the lower the penalty given to an individual the better it's considered to be. Each generation is formed by carrying over the two best performing individuals, and creating 3 offspring from the best and 2 offspring from the second best individuals. An equally performing offspring is always favoured over a parent in the next generation's selection process.

VI. Investigation

In the developmental system presented, the tuning of the developmental system and the inclusion/design of multicellular developmental mechanisms (e.g. cell signalling) was mostly done by making educated guesses on the behaviour and effects of these mechanisms and parameters. Although there are various implementations and uses of developmental systems in literature, there is not much exhaustive work on the investigation of the effects of developmental parameters and mechanisms on artificial multicellular development. Without thorough investigation the tuning of a complex dynamical system like an ADS is not feasible, and before using an ADS to solve complex problems, it is important to understand the effects of mechanisms and parameters that might have a drastic impact on an ADS's performance.

This section investigates the importance of various mechanisms and parameters on the ability of the ADS to successfully achieve cellular organisation that matches a given ordered pattern. These mechanisms and parameters are:

- *Diffusion*
- *Direct contact signalling*
- *Protein production and chemical consumption rates*
- *Chemical thresholds for gene binding*
- *Various chemical/protein types*
- *The methods of mapping development to a phenotype*
- *Preconditional and postconditional decision methods*
- *Artificial simulation of food reliance*

The problem of explaining how genetic mechanisms can produce the many differentiated patterns of cells found in multicellular biological organisms remains a fundamental research challenge in molecular and developmental biology [10]. Inspired by this we have devised a series of benchmarks based on pattern formation. Although developing spatial patterns does not appear directly useful in engineering applications, it proves to be very useful for examining and comparing micro-model developmental approaches and ascertaining the importance of mechanisms and parameters. We use three types of patterns, which were also examined in [8], namely:

- *Mosaic patterns*
- *Border patterns*
- *Patch patterns*

Three main pattern types were selected in order to test the ADS on building ordered structures of different types. ADSs have been well established in successfully matching ordered patterns, and ordered patterns are a common part of biological development as well. Hence, using ordered patterns of different regularities is a convenient and intuitive way of providing initial benchmarking experiments to ADSs. The experiments involving these spatial patterns do not intend to demonstrate the capabilities of the provided ADS in machine learning or artificial intelligence. In fact the simplicity of these experimental tests are a key part of the investigations undertaken, since it is important to fully understand the changes in the behaviour of the ADS with different mechanisms.

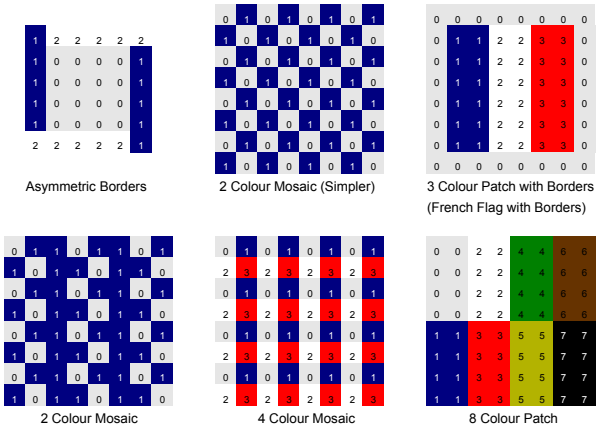
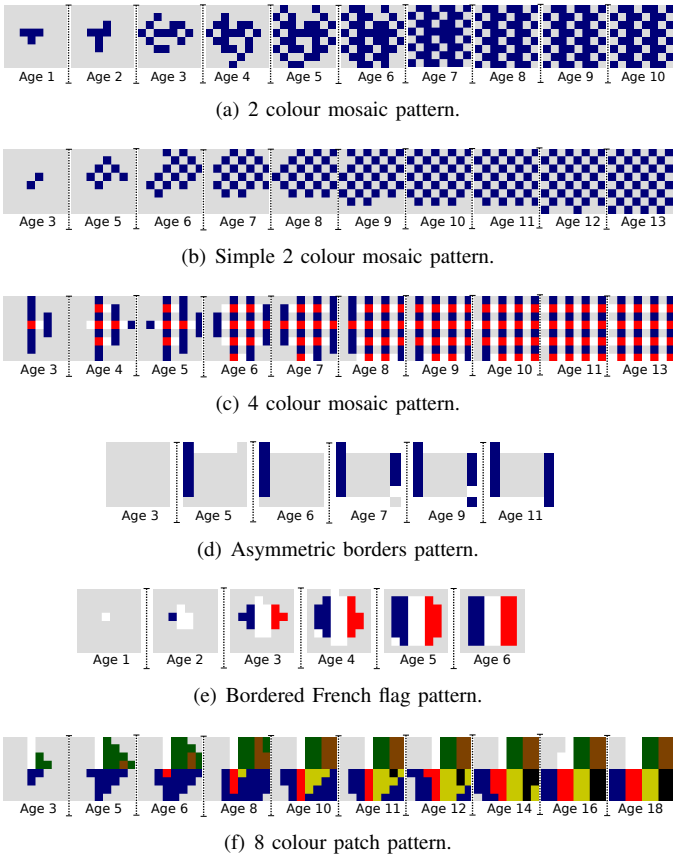


Fig. 5. The patterns used for investigating the effects of developmental mechanisms and parameters. A border pattern with asymmetric borders of size 6*x*6 is the first pattern. The next three patterns (size 8*x*8) are of mosaic nature with different complexities. The simple 2 colour (simplest of the three) and 4 colour mosaic patterns have a 2*x*2 size for the periodic pattern, whereas the 2 colour mosaic pattern (most complex of the three) has a 4*x*2 size for its periodic pattern. The next two patterns (size 8*x*8) can be classified as patch patterns, one being the popular French flag pattern and the other one a patch pattern of 8 colours. A border around the French flag pattern is added making it a hybrid – bordered patch pattern.



(a) 2 colour mosaic pattern.

(b) Simple 2 colour mosaic pattern.

(c) 4 colour mosaic pattern.

(d) Asymmetric borders pattern.

(e) Bordered French flag pattern.

(f) 8 colour patch pattern.

Fig. 6. Developmental path of some evolved organisms that form the patterns in Figure 5 are shown.

The independence of mechanisms investigated can not be guaranteed but using a well understood problem domain for artificial development provides an easy to digest investigation that is open to future extensions.

These three types of patterns demonstrate different ordering characteristics and complexities in patterns for the experiments testing the characteristics of the ADS. A mosaic pattern is a periodic pattern that repeats a smaller motif over the whole pattern area. A border pattern is a group of cells (pixels) isolated by a thin layer of different type of cells. Finally, a patch pattern is the division of different groups of cells into an aperiodic partitioning of groups of cells of the same type. These three types of patterns are also common in natural development, and they can be seen in various parts of different developmental organisms [8]. Out of these three categories, six patterns are chosen as test patterns for all the experimental scenarios. These patterns are illustrated in further detail in Figure 5. Each pixel of a pattern is represented by a cell, which specifies the colour of the pixel as the cell type. For all the patterns except the 8 patches pattern in Figure 5, the maximum number of colour types is set to 4 (i.e. colours range from $0 \rightarrow 3$). For 8 patches pattern the maximum number of colour types is increased to 8.

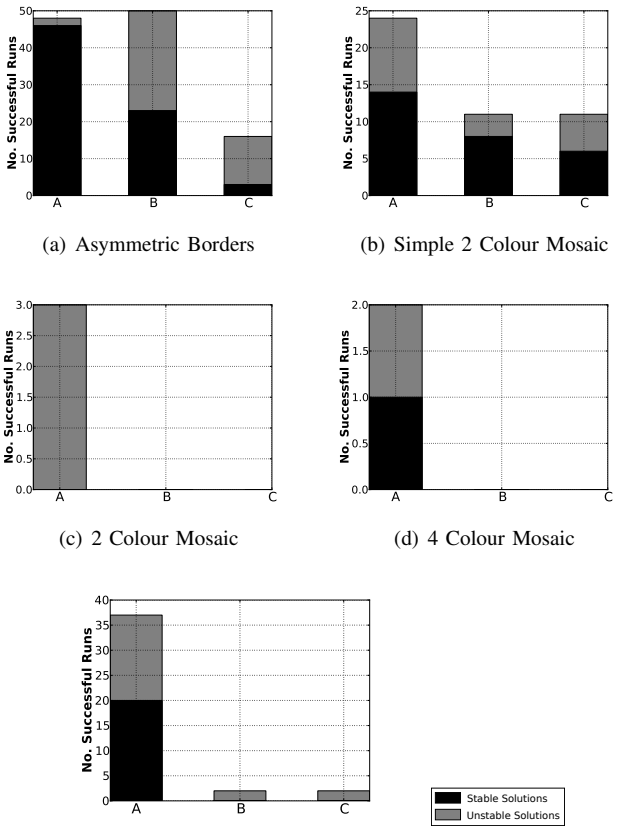
The experiments described in this section are composed of 50 evolutionary runs for each case, each evolutionary run being limited to 1 million evolutionary generations. The evolutionary algorithm is the one described in Section V. Non-deterministic maturing is used for all the runs, and the maximum maturing age is limited to 30. This means that the organism would not always be expected to mature at a fixed developmental step, but it would be allowed to mature at any developmental step between a minimum (3 in the experiments presented here) and a maximum (30 in the experiments presented here). Each run that achieves the perfect representation of the target pattern is evolved further (until the generation limit is reached) until a stable solution is found. To test the stability of a run, the organism is developed for ten extra developmental steps (e.g. $19 \rightarrow 28$) and for each step matched to the target pattern. If an organism can form a pattern for *ten* consecutive developmental steps, it is marked as a stable solution. Of course in reality not all the solutions marked as stable are in fact stable. Testing a candidate solution for 100% stability is too time consuming to be incorporated into the evolutionary phase. The test used here filters most of the transient solutions, and encourages (but not force) evolution to find stable solutions. Figure 6 illustrates the developmental path of the target patterns from a single cell, as it can be seen each organism has a different maturing age. The developmental path of some of the organisms achieving imperfect solutions in the experiments provided are shown at the end of this section in figures 19 and 20

In the rest of this section, the experiments for each investigation are presented and analysed in respective subsections. The experimental data may be presented and analyzed in 3 different forms. The first form presents the number of successful runs (as well as the stablized runs) for each experiment in bar chart form. These bar charts provide a simple comparison betwen different test cases in terms of finding the perfect solution to the given problem. For experiments where the number of successful runs does not provide sufficient information, the fitness information (or sometimes the number of evolutionary

generations to find successful solutions) is used to create box and whisker plots for each experimental case.

A box plot displays a variety of information about the sample data being plotted. The median of a sample data is shown as a line in the middle of a box. The 25^{th} and 75^{th} percentiles of a sample are drawn as the lower and upper lines of the box. The distance between the lower and upper lines of the box is referred to as the inter-quartile range. The rest of a sample is shown by the whiskers of the plot, which covers up to 1.5 times the inter-quartile range of the sample. If there are any data outside the range covered by the whiskers, these are then marked by diamond symbols above and below the whiskers and they are referred to as outliers. The notches in the box plot, first introduced by [80], are a graphic confidence interval about the median of a sample. A small confidence interval demonstrates a good sample of data. A side-by-side comparison of two notched box plots can provide information on the statistical difference between the two data sets. If the notches of the two data sets do not overlap the medians are significantly different at a 95% confidence.

To provide a clearer differentitation among some specific cases, statistical and scientific significance comparison tests are used; the ranksum (p-value) and Vargha-Delaney A statistic tests. A 5% significance level is used for all the rank-sum tests presented in this section, hence a p-value of $< 5\%$ indicates a statistically significant difference in the results of the two ADS versions being compared. To analyse scientific significance, the Vargha-Delaney A statistic is calculated, which is a measure of effect size compared to stochastic noise [81]. This statistic is independent of the sample size and has a range of 0-1: a value of 0.5 indicates identical performance between the two samples. Values smaller or larger than 0.5 suggests increasingly large effect sizes; a value larger than 0.5 indicating a better performance for the first of the two samples. An A statistic value greater than 0.64, or less than 0.36, indicates a "medium" or "large" effect size [81]. Any comparison demonstrating "medium" or "large" effect sizes is considered to be scientifically significant.



(a) Asymmetric Borders (b) Simple 2 Colour Mosaic

(c) 2 Colour Mosaic (d) 4 Colour Mosaic

(e) French Flag

Fig. 7. Bar charts displaying the number of successful runs out of fifty runs for five of the six experimental patterns tried by developmental models using different contact signalling mechanisms. For the 8 patches pattern none of the models returned a successful run. Grey Bars show the total number of successful runs and black bars show the successful runs that are also stable. **A** indicates controlled tunnels, **B** simple tunnels, and **C** no tunnels.

A. Direct Contact Signalling

Contact signalling is the most popular cell signalling model in ADSs, see Table I. It is simple to implement, and biological evidence strongly suggests that direct contact signalling is an essential part of development [69]. Experiments in this section will investigate the usefulness of direct contact signalling. Three versions of the developmental model with different contact signalling mechanisms are investigated in the experiments. In the ADS used here, contact signalling is established by the mutual transcription of plasmodesma proteins (described in Section IV), which form connections between two neighbouring cells. These connections are formed as plasmodesmata, a tunnel, between the two cells.

- **Simple tunnels**: The initially designed version of plasmodesmata between two cells involved a "simple tunnel"; no filtering of chemicals were done, and all the chemicals available in both cells were entirely shared amongst the two cells.
- **Controlled tunnels**: A more complex version of the original design. When a plasmodesmata is formed, the chemicals are not freely shared between the two cells. A chemical may pass through the tunnel only as a result of further gene regulation. A chemical is transferred through an existing tunnel when a plasmodesma protein coded with the concentration and identity of the chemical to transfer through the tunnel is expressed by a gene. This approach of contact cell signalling is more complex than the original design, but it provides individual cells the chance to protect their specialization in case of an emerging plasmodesmata with their neighbour. In the original design, the cells that are connected together were forced to share all their chemicals creating two identical cells at the cost of loss of the identity of both cells, causing loss of information. Simple tunnels may have had adverse effects on the ADS performance by making cell specialization a tougher task.
- **No contact signalling**: In this case direct contact signalling is disabled, i.e. no plasmodesmata (tunnels), leaving diffusion as the only cell signalling process within the organism.

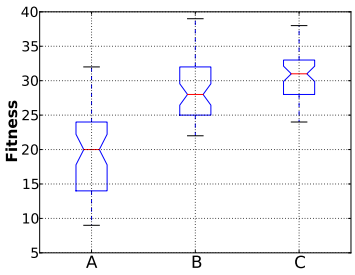


Fig. 8. A fitness box and whisker plot of the different contact signalling models for 8 patches pattern. The box plot shows the best (minimum) fitness achieved as well as the fitness distribution via quartiles.

1) Analysis: The success rates of the experiments are presented in Figure 7. All of the experiments suggest that a contact signalling mechanism is essential for effective multicellular development. The developmental system had a poor performance for all the patterns when it lacked contact signalling (labelled as *No Tunnelling*); for some patterns the success rate was 0/50. This strongly suggests that a simple diffusion mechanism on its own is unable to organize a multicellular system effectively.

The newly introduced direct contact signalling model (marked as *controlledTunnels*) improved the success rate as well as the number of stable solutions in all patterns (except 8 patches pattern). In most cases the *simple tunnels* model is unable to provide an effective signalling mechanism. This suggests that even though the *controlled tunnels* model is a more complex signalling model than the *simple tunnels*, the developmental system greatly benefits from the larger control over the shared chemicals between individual cells.

Since none of the versions of the ADS were able to find a perfect match for the 8 patches pattern, a box and whisker plot (referred to as box plot) for the fitness achieved over the 50 runs for each experiment is displayed in Figure 8.

The best fitness is achieved by the *controlled tunnels* contact signalling mechanism at 9; meaning there were 9 mismatched pixels from the target image. The *controlled tunnels* also achieves the best fitness distribution among the three signalling models, *simple tunnels* having the second best distribution. The box plots indicate that *controlled tunnels* mechanism provides a significantly different data sample from the other two mechanisms, which has the lowest median value at 20. The *simple tunnels* and *controlled tunnels* were analysed for scientific and statistical significance with rank-sum (p-value) and Vargha-Delaney A statistic tests as well.

The comparison of *simple tunnels* with *controlled tunnels* yield a p-value of $< 10^{-6}$ and an A statistic of 0.104, which suggests a statistically and scientifically significant difference between the performance provided by the two methods, *controlled tunnels* being significantly better.

2) Conclusions: Observing the data provided by figures 7 and 8, we can conclude that contact signalling for multicellular organization is essential. However a careful design of contact signalling is required in order to fully benefit from its use in multicellular organization. Since the *controlled tunnels* version of the contact cell signalling mechanism used in the experiments here proved to be the most effective, the rest of the experiments presented in this section use an ADS with *controlled tunnels* contact signalling mechanism.

B. Diffusion

Previously, Section III-E discussed the simulation of diffusion as a cell signalling mechanism in ADSs. It was noted that although it may not be desirable to have uncontrolled diffusion, many successful implementations of ADSs exist that rely purely on chemical diffusion. The need for diffusion, and how it should be implemented, as a cell signalling mechanism for a successful ADS is not evident. The experiments in this subsection intend to investigate these questions.

Six different diffusion models are investigated in this section for their effects on the developmental system's ability to achieve the patterns shown in Figure 5.

- **Unrestricted constant diffusion**: The diffusion mechanism that was included in the initial design. The diffusion process is carried out for all chemicals available in every cell each developmental step; half of an available chemical diffuses out of a cell equally to the four nearest neighbours, i.e. each neighbour obtains $\frac{1}{8}$ of the cell's chemicals each developmental step.
- **Generic diffusion proteins**: In this case the diffusion process only takes place when a diffusion protein is produced by a gene. When the dedicated diffusion protein is produced by a gene, the diffusion protein uses the information provided by the postconditional part of the gene to determine the amount of chemical concentration (same for each chemical type) to diffuse out. Therefore the diffusion of chemicals is controlled by gene regulation within a cell.
- **Chemical Specific diffusion proteins** (diffusion protein [DP]): In this case when the dedicated diffusion protein is produced by a gene, the diffusion protein uses the further information provided by the postconditional part of the gene to determine how much of which chemical to diffuse out. Therefore the diffusion of each chemical is controlled individually by the gene regulation within a cell.
- **Diffusion layer**: An extra layer for the diffusing chemicals is included to simulate a more realistic diffusion of chemicals throughout the organism. A chemical only diffuses from a direction of high concentration to a low concentration and not vice versa in this case. The amount of chemical that diffuses is described by Equation 3: the flow (ΔY_{pat}) of chemical a from position p to position q is calculated as their difference in their concentrations Y. DC is the diffusion constant.
- **Diffusers**: These are chemical sources placed around the organism diffusing out chemicals at a constant rate. These diffusers are used as well as an unrestricted constant diffusion of chemicals from the cells themselves. The idea of diffusers was used in Roggen's morphogenesis model [19].
- **No diffusion**: Diffusion mechanism is completely removed from the system, only leaving cell to cell contact signalling.

TABLE II
THE DIFFUSION MECHANISMS USED IN THE EXPERIMENTS.

Reference in Plots	Mechanism
A	Constant diffusion
B	No diffusion
C	Generic diffusion proteins
D	Chemical specific diffusion proteins (DP)
E	Diffusion layer with evolved Diffusion Constant (DC)
F	Diffusion layer with fixed DC
G	Diffusion layer that uses DP
H	Diffusion layer with evolved DC that uses DP
I	Diffusion layer with evolved DC that uses generic DP
J	Diffusers in all cells - 127
K	Diffusers in all cells with no don't care genes - 127
L	Single diffuser - 127
M	Single diffuser with no don't care genes - 127
N	Single diffuser - 255

$$\Delta Y_{pat} = (Y_{pat} - Y_{qat})/DC \quad (3)$$

1) Analysis: Figure 9 contains bar charts displaying the number of runs the developmental system was successfully evolved to match the patterns shown in Figure 5 for different diffusion mechanisms. Unlike the contact signalling experiments there is no single diffusion mechanism that outperforms the other mechanisms in achieving all the experimental patterns. However just like contact signalling, diffusion had a big impact on the ADS performance. Most of the problems suffered from uncontrolled presence (except the bordered French flag pattern) or total absence (except asymmetric borders pattern) of diffusion.

Unrestricted constant diffusion was only favoured by the popular French flag pattern, and had negative effects on the formation of all other patterns. Although lack of diffusion reduced the performance of the developmental system in general, its effects were not as big as having constant diffusion. Once more, introducing a more controlled method of communication seemed to improve the ADS performance in general. Figure 9 shows that 8 patches and 4 colour mosaic patterns benefited from the presence of an extra diffusion layer in the system whereas 2 colour mosaic, simple 2 colour mosaic, and 8 patches patterns benefited from the use of diffusion proteins. Use of diffusion proteins that control the diffusion of single chemicals rather than the diffusion of all chemicals at once slightly improved the performance in 4 colour mosaic, 8 patches, and asymmetric borders patterns due to the higher precision control in the ADS; whereas the performance deteriorated in other cases due to the increase in the required number of genes in the control of chemical diffusion. The use of diffusers did not have any notable benefits, in fact in most of the cases the performance of the ADS deteriorated when diffusers were included in the system.

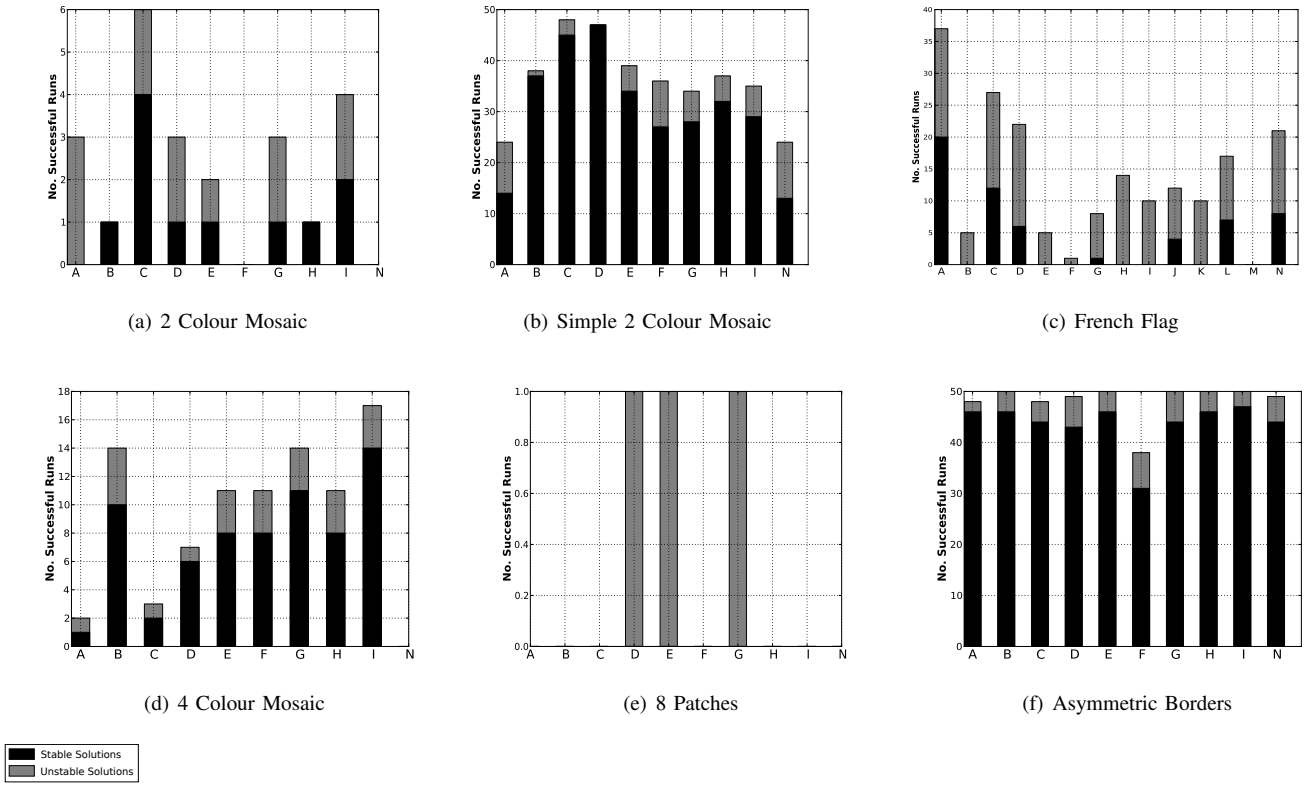
For some of the experiments with a diffusion layer, the diffusion constant was evolved instead of being pre-set. For most of the experiments this provided an improvement in evolutionary performance, although not a big improvement.

Deciding whether the use of a diffusion layer, diffusion protein, or both for controlling the diffusion process to achieve the overall best diffusion mechanism for the ADS is not very clear from the bar charts shown in Figure 9. To provide more information on the performance, Figure 10 provides box plots on the fitness distribution on all the runs of each experiment. Since almost all the experiments on asymmetric borders pattern had high success rates (i.e. fitness 0), the box plots on the distribution of the number of generations to achieve stable solutions is shown instead of the fitness distribution.

Looking closely at the results of all the options for diffusion mechanisms in Figure 10, using a diffusion protein as the diffusion control mechanism seems to give the best results. But in every pattern except the French flag, the confidence interval of the best diffusion protein runs overlap with the confidence interval of the best diffusion layer runs. Hence it is worth taking a closer look at the best cases in a side by side significance comparison for a better conclusion. Table III compares chemical specific diffusion protein with six different versions of the diffusion mechanisms using Vargha Delaney A statistic and Mann-Whitney-Wilcoxon p-value. As described earlier a p-value of $< 5\%$ is considered to represent a statistically significant difference between the two data sets being compared, as well as an A value above 0.64 or below 0.36 representing a scientifically significant difference.

2) Conclusions: It is clear that a total presence of constant diffusion or total absence of diffusion are both undesirable. It is most advantegous to include a diffusion mechanism in a micro-model developmental system that can easily be controlled by the developmental system itself.

Addition of a diffusion layer or a diffusion protein in the ADS improves the overall ADS performance compared to constant or no diffusion. However, the use of diffusion protein on its own seems to be sufficient, and the addition of a diffusion layer –a more complex mechanism– does not seem to make much difference. Hence the optimal diffusion mechanism is a diffusion protein in which the GRN controls the chemical diffusion. Chemical specific diffusion protein introduces extra complexity to the system, reducing the ADS performance when compared to the generic diffusion protein in some of the experimental patterns, but in the experiments provided in this section the effects of this complexity is mostly negligible. When a larger number of chemicals are used, these effects may become more significant. It is noteworthy that in the most complex of all the patterns tried in this section, the 8 patch pattern, the use of chemical specific diffusion protein made a significant improvement in the fitness distribution. The performance difference between the two different implementations of the diffusion protein are almost marginal, but there was one case (the 8 patch pattern) where the chemical specific diffusion protein provided better results with a scientific significance. The performance differences between the two implementations of the diffusion protein in all other experimental problems were below the A statistic significance threshold.



(a) 2 Colour Mosaic (b) Simple 2 Colour Mosaic (c) French Flag

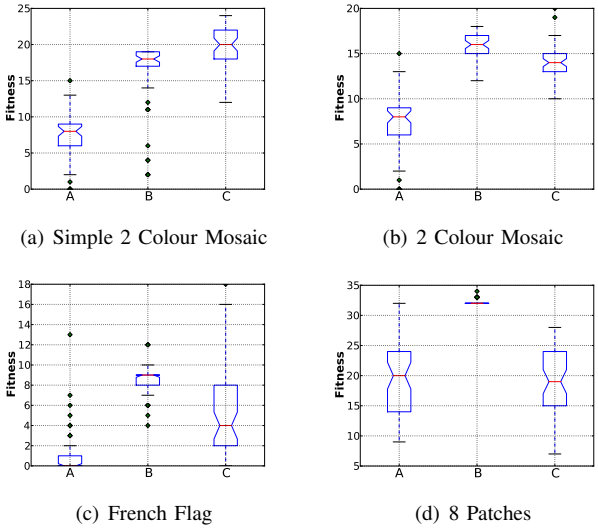
(d) 4 Colour Mosaic (e) 8 Patches (f) Asymmetric Borders

Fig. 9. Bar charts displaying the number of successful runs out of fifty runs for different diffusion mechanisms in achieving the mosaic patterns. The aliases used in the x-axis are explained in Table II.

C. Mapping The Cell Phenotype

The function of the cell (referred to as the cell phenotype from now on), which is part of the target function for the ADS (e.g. a pixel value in the case of patterns) can be built in a number of ways. The common method of obtaining the cell phenotype is to use a mechanism within the developmental processes of the cell to provide a structural output as an emergent result of the developmental processes. The most popular way of building the cell phenotype in systems that use GRNs is to use specialized proteins that change the cell phenotype when they are being expressed during the developmental process. Other developmental systems that do not use GRNs tend to use the evolved developmental cell program both for regulatory purposes and building the cell phenotype. Therefore the cell phenotype is determined by one or more of the outputs of the cell program. The behaviour of the cell phenotype in these cases is directly affected by the developmental processes. There are few other methods of building the cell phenotype, and most of these methods are an emergent result of the developmental progress of the organism. This is the case for all except three of the developmental models listed in Table I; Gordon [31], Kitano [55] and Zhan et al. [72] do not use integrated mechanisms within development that build the cell phenotype as an emergent result of the respective developmental system. Instead they use the chemical concentrations at the end of a developmental step or phase to map the cell phenotype (in the application domain).

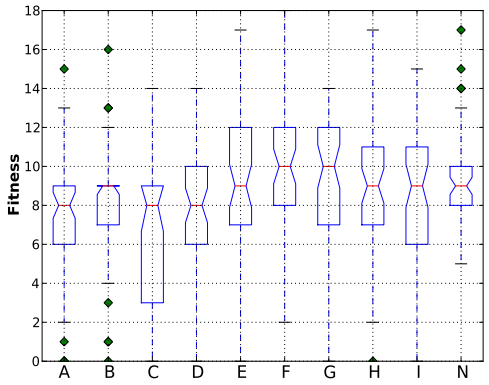
In this section some experiments were carried out to investigate whether there is an advantage of using an extra mechanism within the GRN based developmental system to build the cell function, or whether it is as good to make use of the chemical concentrations within each cell at the end of a developmental step to build a cell function for every cell. Using the chemical concentrations at the end of a



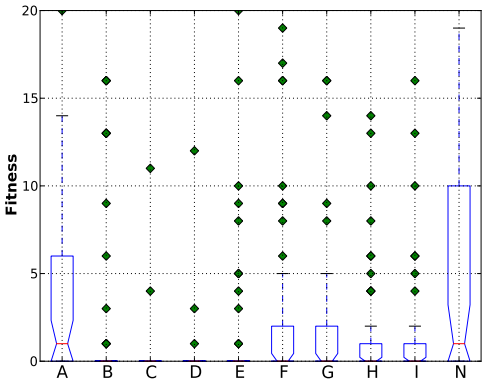
(a) Simple 2 Colour Mosaic (b) 2 Colour Mosaic

(c) French Flag (d) 8 Patches

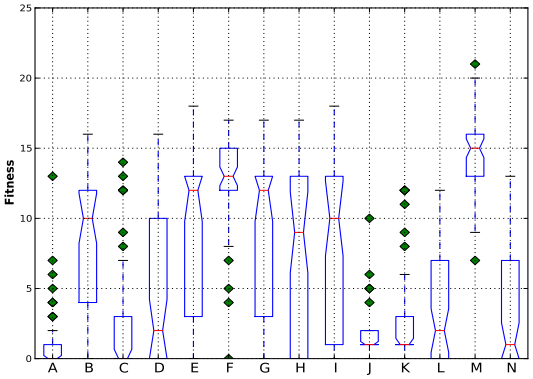
Fig. 11. Box and whisker plots on the fitness distribution of the different methods of constructing cell phenotype. The aliases used in the x-axis are explained in Table IV.



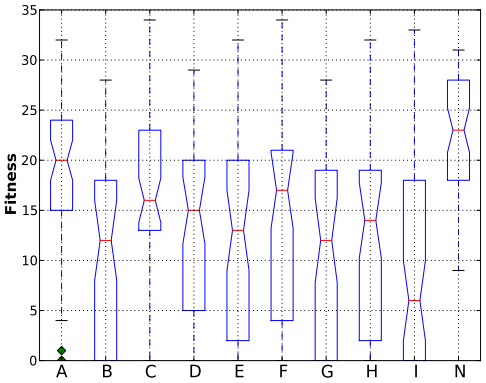
(a) 2 Colour Mosaic



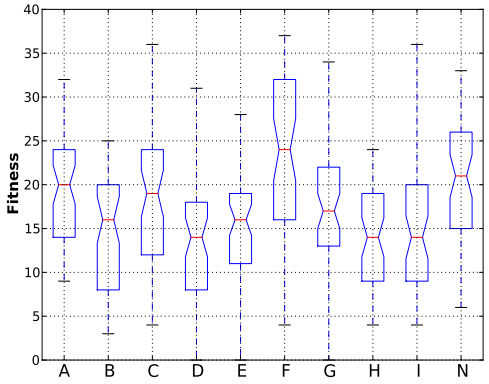
(b) Simple 2 Colour Mosaic



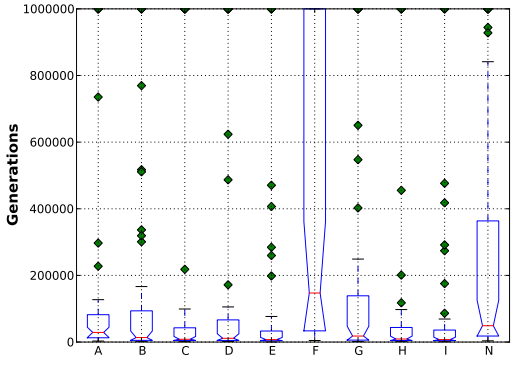
(c) French Flag



(d) 4 Colour Mosaic



(e) 8 Patches



(f) Asymmetric Borders

Fig. 10. Box plots of fitness distribution in fifty runs for the different diffusion mechanisms in achieving all the patterns except the asymmetric borders pattern; box plots of the number of evolutionary generations in finding stable solutions are shown for the asymmetric borders pattern. The aliases used in the x-axis are explained in Table II.

developmental step to create a cell function is not biologically plausible, however it simplifies the cell structuring process. Hence once more this imposes a question on the evolvability of bio-inspired vs simplistic mechanisms.

The experiments in this section only use the French flag, simple 2 colour mosaic, 2 colour mosaic and the 8 patches patterns as test patterns. There are three versions of cell structuring methods tested.

- **The structuring protein**: This uses a dedicated protein to change the cell phenotype when it is produced during the gene processing. This is part of the initial design of the developmental system described in Section IV-B.

TABLE III
STATISTICAL COMPARISON OF DIFFUSION MECHANISM USING CHEMICAL SPECIFIC DIFFUSION PROTEIN (LABELLED AS "DIFFUSION PROTEIN") WITH SIX OTHER DIFFUSION MECHANISMS. SCIENTIFICALLY BETTER CASES FOR THE CHEMICAL SPECIFIC DIFFUSION PROTEIN ARE MARKED WITH TICK MARKS, AND WORSE CASES ARE MARKED WITH CROSSES; EXPLAINED FURTHER BELOW THE TABLE. FOR THE STATISTICAL SIGNIFICANCE THE ACTUAL P-VALUES ARE SHOWN, AND THE STATISTICALLY SIGNIFICANT DIFFERENCE IN THE EFFECTS OF THE TWO MECHANISMS ARE RECORDED IN BOLD WHITE TEXT. FOR ALL EXAMPLES EXCEPT ASYMMETRIC BORDERS, THE FINAL FITNESS DISTRIBUTION FOR EACH EXPERIMENT IS USED. FOR THE ASYMMETRIC BORDERS, NUMBER OF EVOLUTIONARY GENERATIONS REQUIRED TO ACHIEVE STABLE SOLUTIONS IS USED TO CALCULATE THE A STATISTIC AND THE P-VALUE.

Diffusion Protein vs	Statistic	2 Colour Mosaic	4 Colour Mosaic	Simple 2 Colour Mosaic	8 Patches	Bordered French Flag	Asym. Borders
Generic Diffusion Protein	A Statistic	XX	✓	=	✓✓✓	X	X
	p-value	**0.024**	0.062	0.062	**0.001**	**0.048**	0.277
Diffusion Layer-EvoDC	A Statistic	✓✓	X	✓	✓	✓✓✓	X
	p-value	**0.037**	0.326	**0.035**	0.105	**5.1×10^{-5}**	0.101
Diffusion Layer-DP-EvoDC	A Statistic	✓	X	✓✓	✓	✓✓✓	X
	p-value	0.243	0.343	**0.014**	0.264	**0.006**	0.239
Constant Diffusion	A Statistic	X	✓✓✓	✓✓✓	✓✓✓	XXX	✓✓
	p-value	0.271	**0.001**	**3.7×10^{-6}**	**6.3×10^{-5}**	**0.0003**	**0.015**
No Diffusion	A Statistic	✓	X	✓✓	✓	✓✓✓	✓
	p-value	0.488	0.100	**0.0212**	0.174	**0.0003**	0.483
Diffusion Layer-Generic DP-EvoDC	A Statistic	✓	XX	✓✓	✓	✓✓✓	X
	p-value	0.191	**0.015**	**0.006**	0.229	**0.001**	0.177

A Statistic	Legend	Meaning
0.5	=	Same Performance
0.4 - 0.5	X	Slightly Worse
0.36 - 0.5	XX	Worse
<0.36	XXX	Significantly Worse

A Statistic	Legend	Meaning
0.5-0.6	✓	Slightly Better
0.6-0.64	✓✓	Better
>0.64	✓✓✓	Significantly Better

TABLE V
STATISTICAL COMPARISON OF DEVELOPMENTAL SYSTEM USING THE CONCENTRATION OF A SINGLE PROTEIN FOR CELL STRUCTURING TO THE DEVELOPMENTAL SYSTEM USING STRUCTURING PROTEIN.

Statistical Comparison	French Flag	Simple 2 Colour M.	2 Colour M.	8 Patches
Vargha Delaney A Statistic	0.160	0.012	0.035	0.527
p-value	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	0.387

TABLE IV
THE LABELS FOR EACH STRUCTURING MECHANISM USED IN THE EXPERIMENTS.

Reference in Plots	Mechanism
A	Structuring proteins
B	Concentration of all proteins
C	Concentration of a dedicated protein

- **Concentration of a dedicated protein**: In this case, the structuring protein has no action when it is produced during the gene processing stages. At the end of a developmental step, the concentration of one of the proteins is used as an output; the concentration value is divided by the *maximum* chemical concentration level, multiplied by the number of possible cell functions and rounded down to an integer value in order to obtain the cell phenotype.
- **Concentration of all chemicals**: In this case all the chemicals in the system are used for determining the cell function. The chemical with highest concentration is used to set the cell function, and each chemical corresponds to a specific cell function.

For all the experiments the ADS used contact signalling with "controlled tunnels" as discussed in Section VI-A, and a constant diffusion mechanism as described in Section III-B.

1) Analysis: In all test cases the runs where multiple and single protein concentrations were used to define cell functions showed poor success rates. Therefore, rather than

TABLE VI
STATISTICAL TEST RESULTS FOR THE DEVELOPMENTAL SYSTEM THAT USES THE *concentration* OF MULTIPLE PROTEINS TO MAP THE CELL PHENOTYPE COMPARED WITH THE DEVELOPMENTAL SYSTEM WITH STRUCTURING PROTEIN.

Statistical Comparison	French Flag	Simple 2 Colour M.	2 Colour M.	8 Patches
Vargha Delaney A Statistic	0.036	0.121	0.018	0.018
p-value	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$

using success rate bar charts, the final best fitness values are used to create box plots (Figure 11). In almost every case using a structuring protein to build the cell phenotype resulted in better performance. Further statistical analysis of structuring protein mechanism vs the other two methods of mapping the cell phenotype are shown in Tables V and VI.

In all cases except one using the structuring protein proved to be significantly better. The exceptional case is the 8 Patches problem where the use of single protein concentration gives slightly better results. However, in this case the use of single protein concentration does not provide a scientifically significant improvement (A statistic of 0.527 against structuring protein), and the distribution difference from the case with structuring protein is not statistically significant (p-value of 0.387).

2) Conclusions: From the observations made, using protein concentrations to build the cell phenotype made the developmental system's ability to create specialized cells harder. It was observed that the cell specializations arise from the different developmental path taken by each cell rather than their final developmental state. For example every cell in an ADS might always end up having similar concentration levels of chemicals at the end of each developmental step, but the changes to the chemical concentration levels during the processing of the GRN would be different for each cell. Thus using information relating to changes in the chemical levels during the processing of the GRN rather than the chemical levels at the end of a developmental step provides a more precise way of differentiating a cell.

In conclusion, using an emergent mechanism employing structuring proteins to build the cell phenotype proved to be more successful than using the final protein concentrations at the end of a developmental phase.

D. Protein Production and Chemical Consumption Rates

In a GRN system the interactions between chemicals and genes are the direct determinants of the computational pathway the GRN system undergoes, and the final states it stabilizes to. As seen earlier (in Figure 2), the presence of chemicals determine whether a gene will be activated, and the activation of a gene increases the amount of a protein (which is a chemical) available in the system, which also regulates the activation of genes. The amount of chemical needed to activate a gene and the amount of protein produced after the activation of a gene affects the number of genes that will be active at one time as well as the number of genes that will never be active within the GRN. The amount of chemical needed to activate a gene is referred to as "Gene Activation Threshold"; the minimum concentration of a chemical required to bind a gene. The amount of protein produced after the activation of a gene is referred to as "Protein Production Rate". When a chemical binds to a gene, a certain amount of that chemical is "used up" by the gene, this is referred to as "Chemical Consumption Rate". These three values directly or indirectly control the gene activation that can be evolved during the evolution of the developmental system or they can be set to a constant value before the evolution phase. Although each target organism might have a different optimal set of values for the gene activation threshold, protein production and chemical consumption rates, evolving these values might not be beneficial as it increases the dimensionality of the genotype.

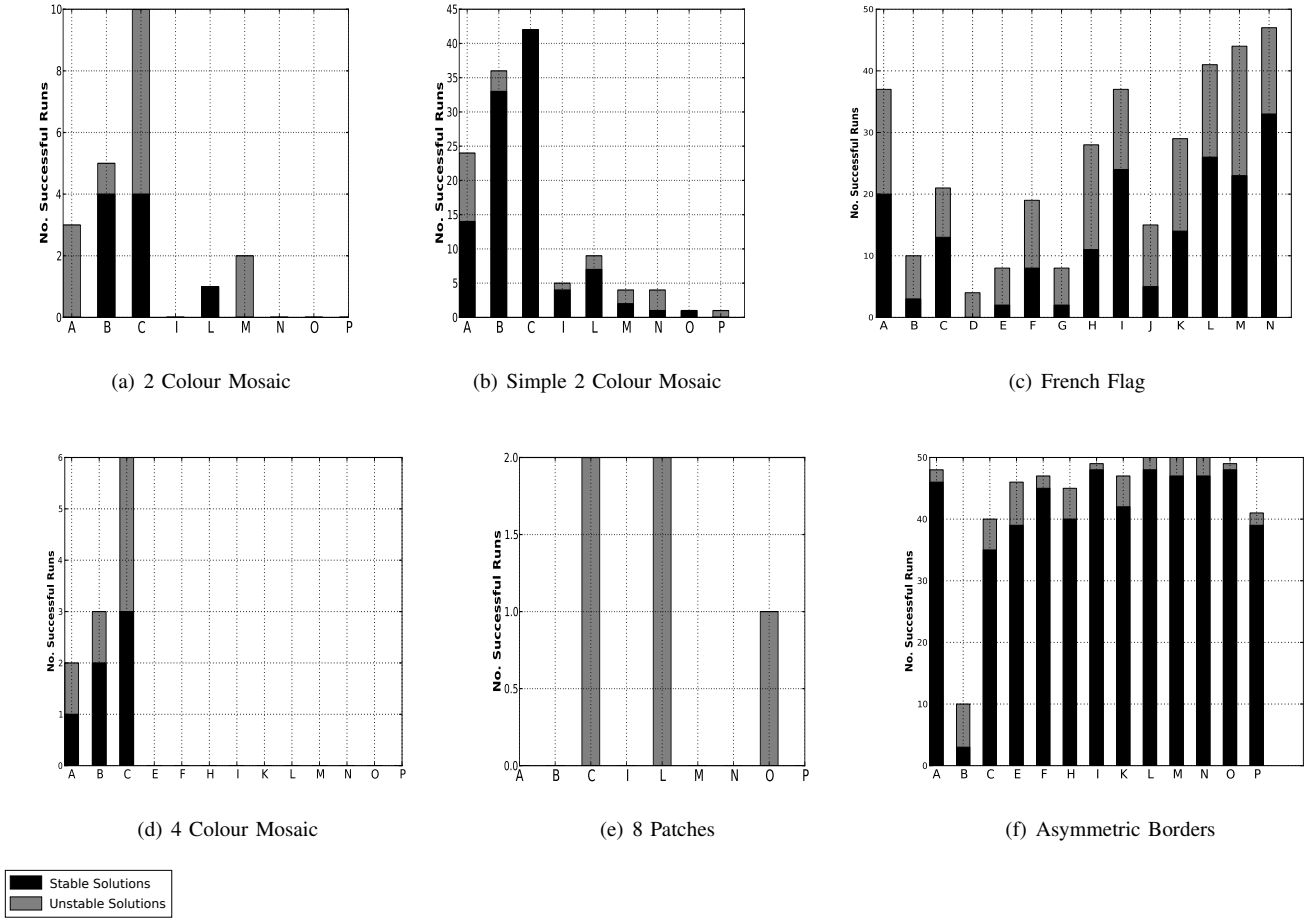
TABLE VII
THE LABELS FOR EACH PROTEIN PRODUCTION AND CHEMICAL CONSUMPTION VALUES USED IN THE EXPERIMENTS.

Reference	Mechanism
A	Original setting - Prod: $\frac{255}{96}$,Cons: $\frac{255}{48}$, Plas prod: $\frac{255}{24}$
B	A single evolved value
C	A value evolved per gene
D	Production: $\frac{255}{12}$, Consumption: $\frac{255}{6}$
E	Production: $\frac{255}{24}$, Consumption: $\frac{255}{12}$
F	Production: $\frac{255}{48}$, Consumption: $\frac{255}{24}$
G	Production: $\frac{255}{12}$, Consumption: $\frac{255}{12}$
H	Production: $\frac{255}{24}$, Consumption: $\frac{255}{24}$
I	Production: $\frac{255}{48}$, Consumption: $\frac{255}{48}$
J	Production: $\frac{255}{12}$, Consumption: $\frac{255}{24}$
K	Production: $\frac{255}{24}$, Consumption: $\frac{255}{48}$
L	Production: $\frac{255}{48}$, Consumption: $\frac{255}{96}$
M	Production: $\frac{255}{48}$, Consumption: $\frac{255}{96}$, Plasmodesma prod: $\frac{255}{192}$
N	Production: $\frac{255}{64}$, Consumption: $\frac{255}{128}$
O	Production: $\frac{255}{64}$, Consumption: $\frac{255}{96}$
P	Production: $\frac{255}{64}$, Consumption: $\frac{255}{128}$, Plasmodesma prod: $\frac{255}{255}$

In order to find out the most evolvable approach to determining the developmental parameters of gene activation thresholds (investigated in Subsection VI-E), protein production and consumption rates (investigated in this subsection) for a developmental system, a series of experiments involving all six patterns shown in Figure 5 were performed.

Three different experiments involving setting the protein production and chemical consumption rates are investigated.

- **Preset values**: A protein production rate that is the same for all genes in the GRN, and a chemical consumption rate that is also the same for all genes and chemicals are initialized at the start of an experiment. Various combinations of different values for both are used for all the experiments.
- **Evolved values**: A protein production rate that is the same for all genes in the GRN is evolved, as well as a chemical consumption rate that is also the same for all genes and chemicals. Therefore only two extra values



(a) 2 Colour Mosaic (b) Simple 2 Colour Mosaic (c) French Flag

(d) 4 Colour Mosaic (e) 8 Patches (f) Asymmetric Borders

Fig. 12. Success rates of experiments with different protein production and chemical consumption rates are displayed. The aliases used in the x-axis are explained in Table VII.

need to be evolved for each developmental system.

- **Evolved values per gene**: A protein production rate and a chemical consumption rate are evolved for each gene. This provides a finer tuning of the protein production and chemical consumption rates but increases the evolutionary search space further by the increasing number of genes.

The experiments provided in this section use "controlled tunnels" as the contact signalling mechanism, and a constant diffusion mechanism. Table VII provides the details for the labels used in the plots used for the protein production and chemical consumption rates experiments. These rates are detailed in the table as; "Production:X" for protein production rate of X, "Consumption: Y" for chemical consumption rate of Y, and "Plasmodesma prod: Z" for the production rate of plasmodesma protein –for experiments that used a different production rate for plasmosdema protein (the contact cell signalling an growth protein). One of the experimental cases in the plots presented in this section is detailed as "Original setting"; this represents a protein production rate of $\frac{255}{96}$ for the plasmodesma protein and $\frac{255}{24}$ for the rest of the proteins, and a chemical consumption rate of $\frac{255}{48}$ for all chemicals (255 is the maximum chemical concentration). The "Original setting" is used for the experiments in the other sections of this paper. The "Original setting" was obtained during the initial design as a result of manual tuning of the system.

Figure 12 shows the results of the experiments. Various combinations of protein production rate at the activation of a gene and chemical consumption rate at the binding of a chemical to a gene are tested. The aim is to see whether there exist a single best combination for these developmental parameters or whether they need to be tuned for each application. If the latter, then evolving the protein production and chemical consumption rates will create a more adaptive system by allowing evolution to tune the ADS. If all the applications can be easily developed with a single set of these parameters then using a fixed set of optimised values would be more beneficial as it would reduce the load on the evolutionary algorithm.

The protein production and chemical consumption rates are evolved with two different approaches. The first approach evolves a protein production and a chemical consumption rate for the whole organism. So, every cell and every gene use the same pair of values for protein production and chemical consumption rates. The second approach evolves the protein production and chemical consumption rates for each gene. Hence, the genome length is extended in the second approach to accommodate these rates for each gene. Evolving these rates per gene will provide evolution the ability to tune these

parameters at a finer level, but if the effect of these parameters is not large then the evolutionary performance will deteriorate due to the increase in evolutionary search space.

1) Analysis: The results in Figure 12 show that different patterns have different optimal protein production and chemical consumption rate values. A compromise setting between the patch and border patterns would be a protein production rate at $\frac{255}{48}$ and chemical consumption rate at $\frac{255}{96}$, which provides not the best but close to best performance for all three patterns. But for the mosaic patterns, this combination of protein production and chemical consumption rates is far from ideal. Evolving these parameters is the best compromise amongst all six patterns, and evolving a set of these rates per gene rather than a single set for the whole organism seems to make a significant difference. For a better understanding of the performance difference obtained via the evolution of these rates, fitness box plots of the results are provided in Figure 13, with the exception of asymmetric borders pattern. For the asymmetric borders pattern the number evolutionary generations to achieve stable solutions is used rather than final fitness values.

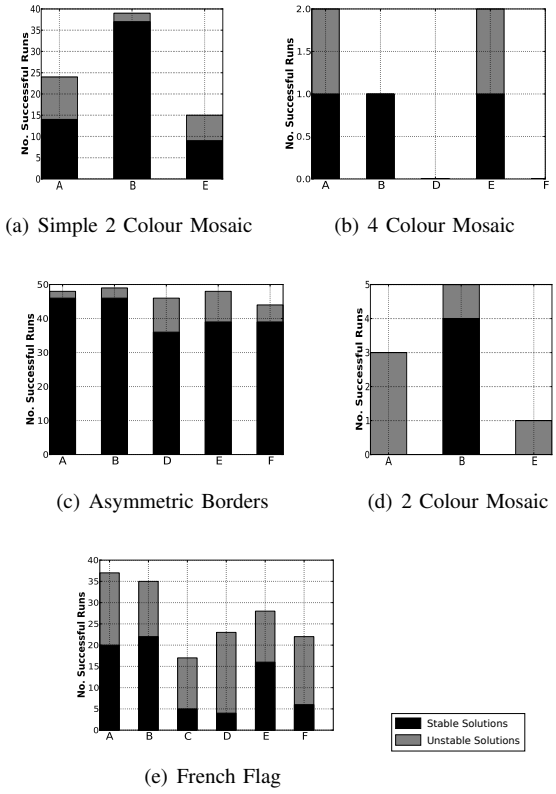
2) Conclusions: Although evolving the protein production and chemical consumption parameters for the ADS during the evolution phase made a large performance improvement for the mosaic patterns, this is not true for the patch and border patterns, see Figure 13. However, even though the patch and border patterns suffer from the use of evolved protein production and chemical consumption parameters, overall, it is worth evolving these parameters for each gene. By evolving the protein production and chemical consumption parameters for each gene:

1) The performance loss in the worst case problems (e.g. evolving asymmetric borders pattern) is balanced by the time it would take to determine the optimal combination of these parameters before running experiments for each problem.
2) There is more control for evolution to fine tune the ADS.
3) Due the the drastic effects these rates might have on the evolability of the ADS, evolving these rates per gene resulted in better results when compared with the evolution of these rates as global values for all genes.

E. Gene Binding Threshold

Three different approaches of setting the chemical concentration threshold for gene binding were investigated.

- **Pre-set thresholds**: The minimum concentration required for a chemical to bind a gene is preset to a constant value before evolving the developmental system. All the genes and chemicals use the same threshold value.
- **Evolved thresholds per gene**: The minimum concentration required for a chemical to bind a gene is evolved for each gene during the evolution phase.
- **Soft Thresholds**: The idea of relaxed threshold limits are implemented in a simplistic way. The "Original setting" is used for this case but when a protein concentration drops or increases to the chemical threshold level, the switching from present to absent or absent to present is not done immediately but a further X amount of chemical concentration increase (if switching to present state) or decrease (if switching to absent state) is allowed. Concentration X is determined via a random number generator that returns a value between 0 and 15.



(a) Simple 2 Colour Mosaic (b) 4 Colour Mosaic

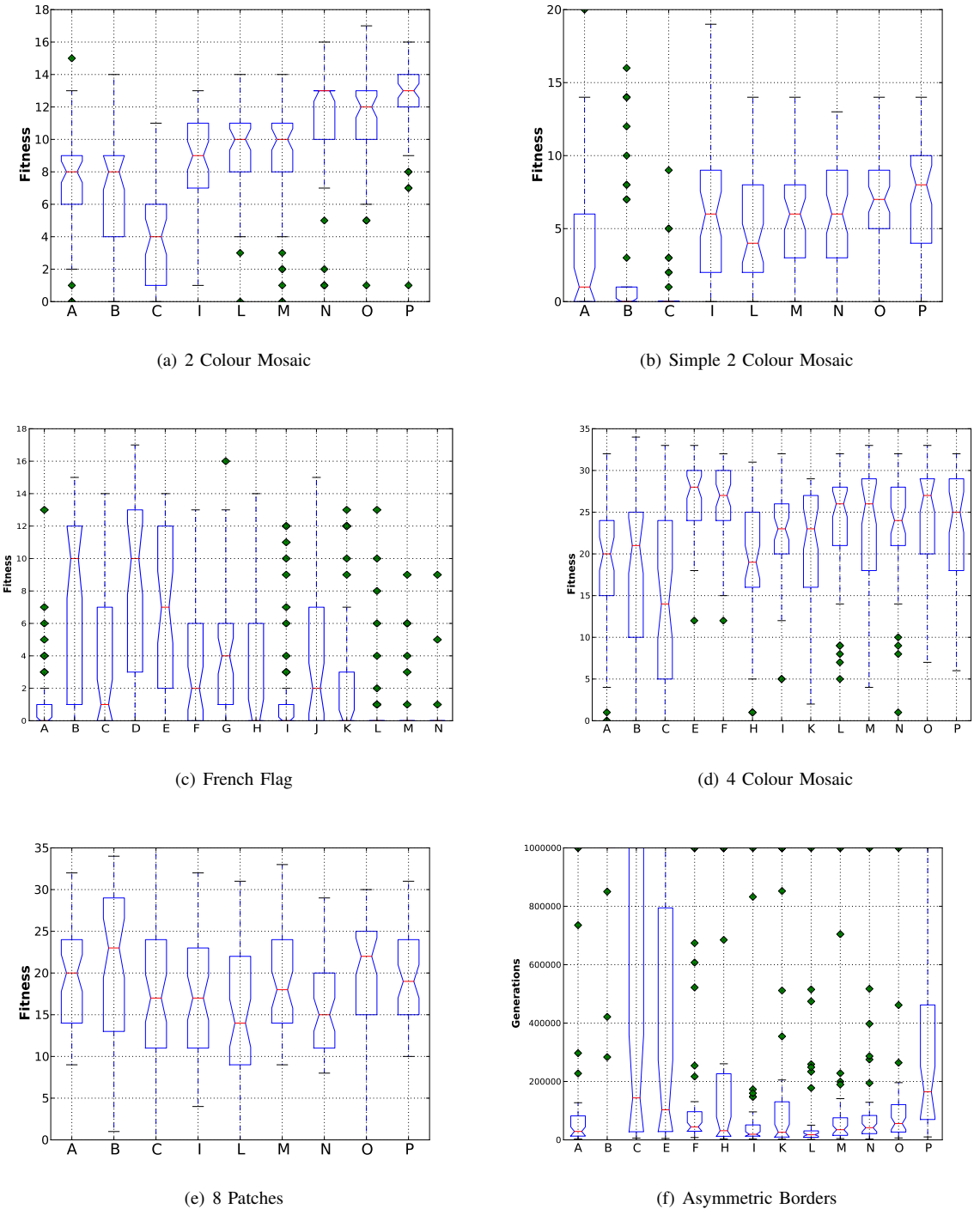
(c) Asymmetric Borders (d) 2 Colour Mosaic

(e) French Flag

Fig. 14. Success rates of experiments with different chemical to gene binding thresholds. Only the most successful settings are shown for some of the test patterns e.g. 2 colour mosaic. 8 patches pattern is omitted because none of the runs in this section returned a success. The aliases used in the x-axis are explained in Table VIII.

The experiments provided in this section use "controlled tunnels" as the contact signalling mechanism, and a constant diffusion mechanism. Table VIII provides the details for the labels used in the plots for the chemical to gene binding thresholds. These rates are detailed in the table as; "Gene inhibition thresh: X" X being the minimum concentration required of a chemical to bind an inhibitory site of a gene, and "Activation thresh: Y" Y being the minimum concentration required of a chemical to bind an excitatory site of a gene. One of the experimental cases in the plots presented in this section is labelled as "Original setting"; this represents a chemical to gene binding threshold level of 127 ($\frac{max}{2}$) for both inhibiting and enhancing chemicals. The "Original setting" is used for the experiments in the other sections of this paper.

1) Analysis and Conclusions: Evolving the chemical to gene binding thresholds improved the performance in most patterns when compared to experiments with pre-set values, and in two cases (French flag and 4 colour mosaic) the results were comparable to the best case ("Original setting"), see figures 14 and 15. Although none of the experiments returned a perfect matching result for the 8 patches pattern,



(a) 2 Colour Mosaic

(b) Simple 2 Colour Mosaic

(c) French Flag

(d) 4 Colour Mosaic

(e) 8 Patches

(f) Asymmetric Borders

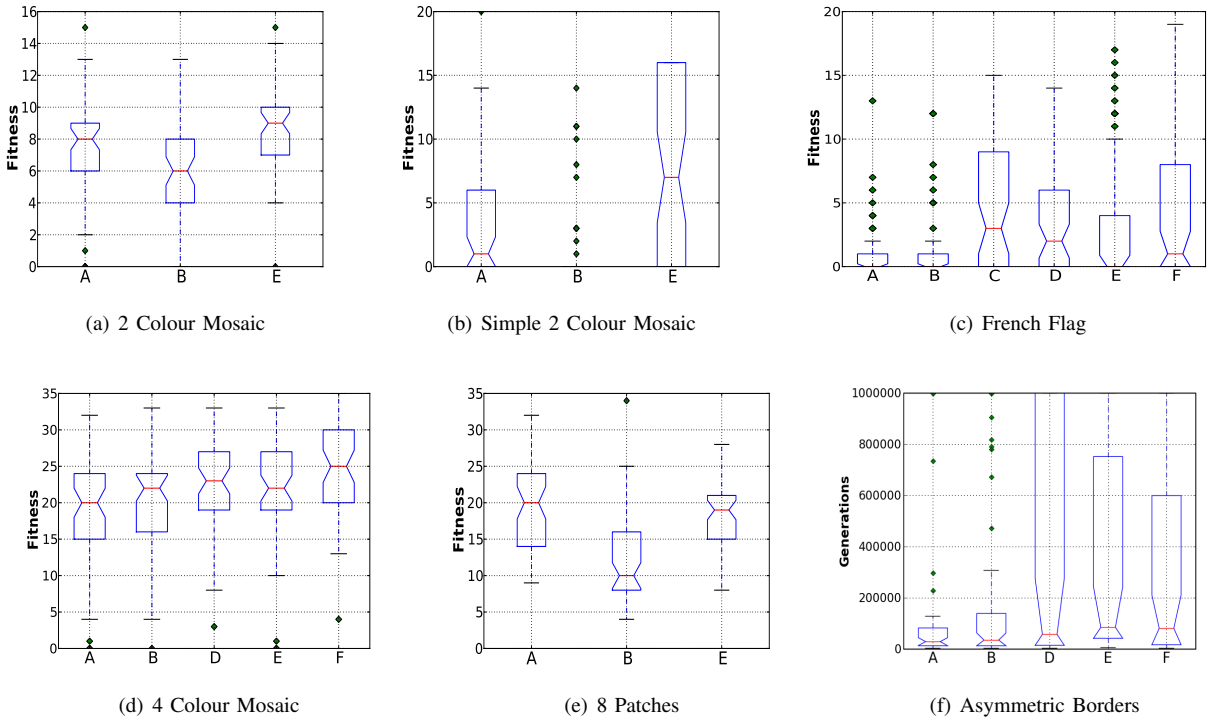
Fig. 13. Box plot of fitness values (the number of evolutionary generations required to achieve stable solutions for the asymmetric borders pattern) reached at the end of each run out of fifty runs of each experiment with different protein production and chemical consumption rates are displayed. The aliases used in the x-axis are explained in Table VII.

evolving the chemical to gene binding thresholds improved the fitness distribution (with the best fitness being 4 mismatched pixels) when compared to pre-set cases, see Figure 15. The "soft thresholds" implementation have only complicated the chemical to gene binding processes and had no benefits in any of the tested cases.

Once more evolving a developmental parameter improved the overall evolvability of the ADS.

F. Other Mechanisms

The use of extra mechanisms in an ADS may enrich the model by providing access to areas of design space that were not accessible before. Some mechanisms may allow the developmental system to reach certain points in the design space



(a) 2 Colour Mosaic (b) Simple 2 Colour Mosaic (c) French Flag

(d) 4 Colour Mosaic (e) 8 Patches (f) Asymmetric Borders

Fig. 15. Fitness box plots of every pattern except the asymmetric borders pattern are displayed in this figure for different chemical to gene binding threshold experiments. Due to the high success rate of most experiments in achieving the asymmetric borders pattern, the number of evolutionary generations to achieve stable results is used for its box plots. The aliases used in the x-axis are explained in Table VIII.

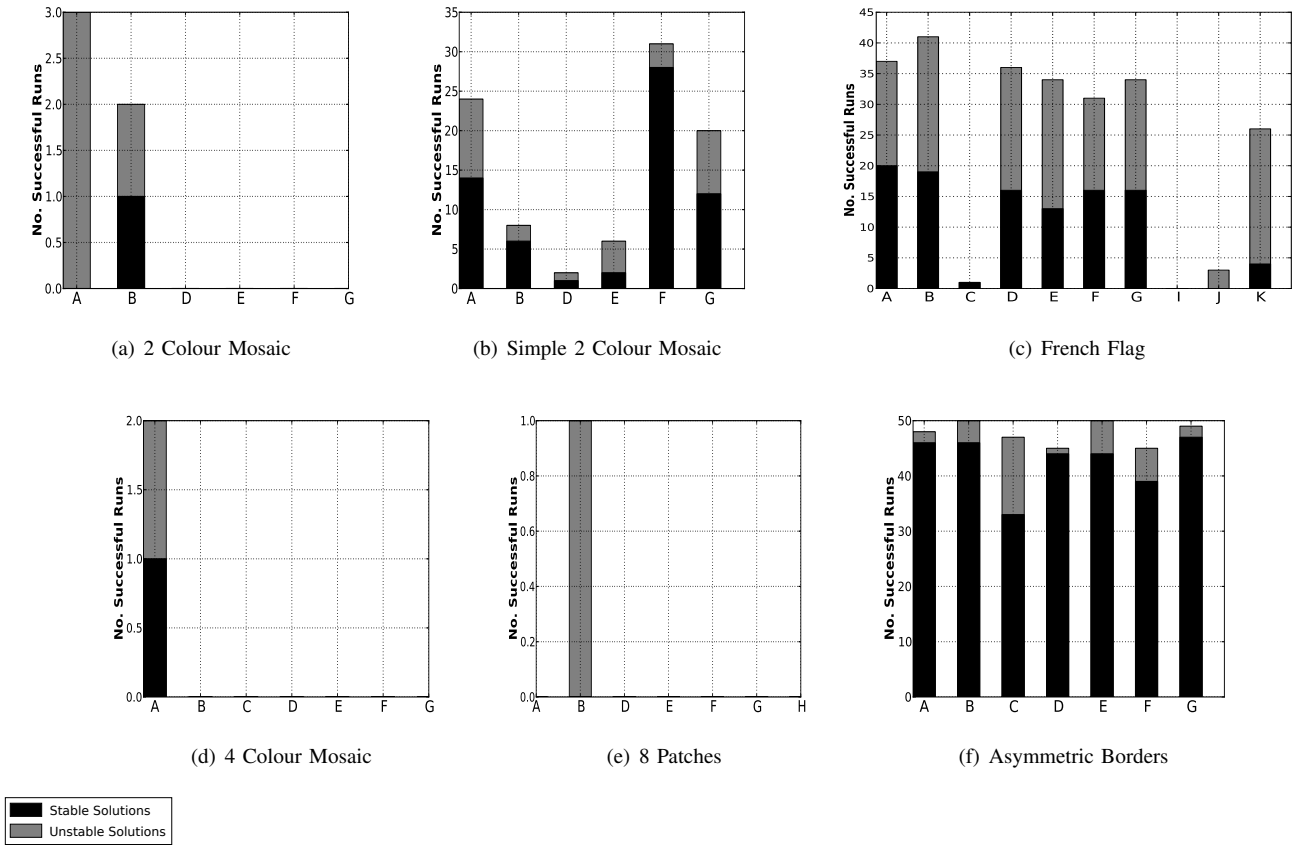
TABLE VIII
THE LABELS FOR PROTEIN CONCENTRATION THRESHOLD VALUES USED IN THE EXPERIMENTS FOR ACTIVATING OR INHIBITING A GENE.

Reference in Plots	Mechanism
A	Original setting
B	Evolved thresholds
C	Gene inhibition thresh: $\frac{255}{4}$, Activation thresh: $\frac{255}{4}$
D	Gene inhibition thresh: $\frac{255}{2}$, Activation thresh: $\frac{255}{4}$
E	Gene inhibition thresh: $\frac{255}{4}$, Activation thresh: $\frac{255}{2}$
F	Soft thresholds at $\frac{255}{2}$

more easily by creating a less rugged search space. On the other hand extra mechanisms may increase complexity of the ADS hence slowing it down. Mechanisms that only duplicate the already existing abilities of a system may create larger biases towards a specific area of a design space forming a more rugged search space. Whether the inclusion of a mechanism in an ADS is beneficial is often not clear. A biologically inspired developmental system is highly dynamic, and it is poorly understood. Predicting the changes in the dynamics of a dynamical system is a challenging task. Extensive empirical data on the behaviour of the ADS with and without the mechanism in question can help with the decision on whether the mechanism is worth keeping as part of the ADS.

Six developmental mechanisms that were not used in the previous experiments are investigated in this section.

- **Local proteins**: The effect of proteins that are only involved with the regulation of the GRN within the source cell is investigated. This is similar to the role of the chemicals in Haddow and Hoye's model [4]. The local proteins are not used for cell communication, hence they are not diffused or allowed to pass onto the neighbouring cells via plasmodesmata (tunnels). In [4] it was demonstrated that chemicals which are only used for local cell regulation increase the complexity of the search space without any visible advantages.
- **Messenger Molecules**: In Section IV-C the role of messenger molecules were explained, but for the experiments presented thus far they were not used. The idea of messenger molecules is to provide a more adaptive system in changing environments, which does not really correspond to stable patterns. Nonetheless the effect of messenger molecules on the evolvability of the ADS for evolving stable patterns is investigated. In the pattern forming experiments presented here, the sensor proteins are used to monitor the phenotype of the neighbouring cells and produce messenger molecules accordingly. The use of phenotypic information from the neighbouring cells to regulate the developmental system has already been used in many developmental systems before. In fact, almost half of the ADSs listed in Table I include this mechanism in their systems [2], [4], [30], [46], [59], [71].
- **Voter decision mechanism**: The decision mechanism on the activation of a gene in the initial design is simply a conjuctive expression. As an alternative mechanism a voter is used for deciding the activation of a gene. The



(a) 2 Colour Mosaic (b) Simple 2 Colour Mosaic (c) French Flag

(d) 4 Colour Mosaic (e) 8 Patches (f) Asymmetric Borders

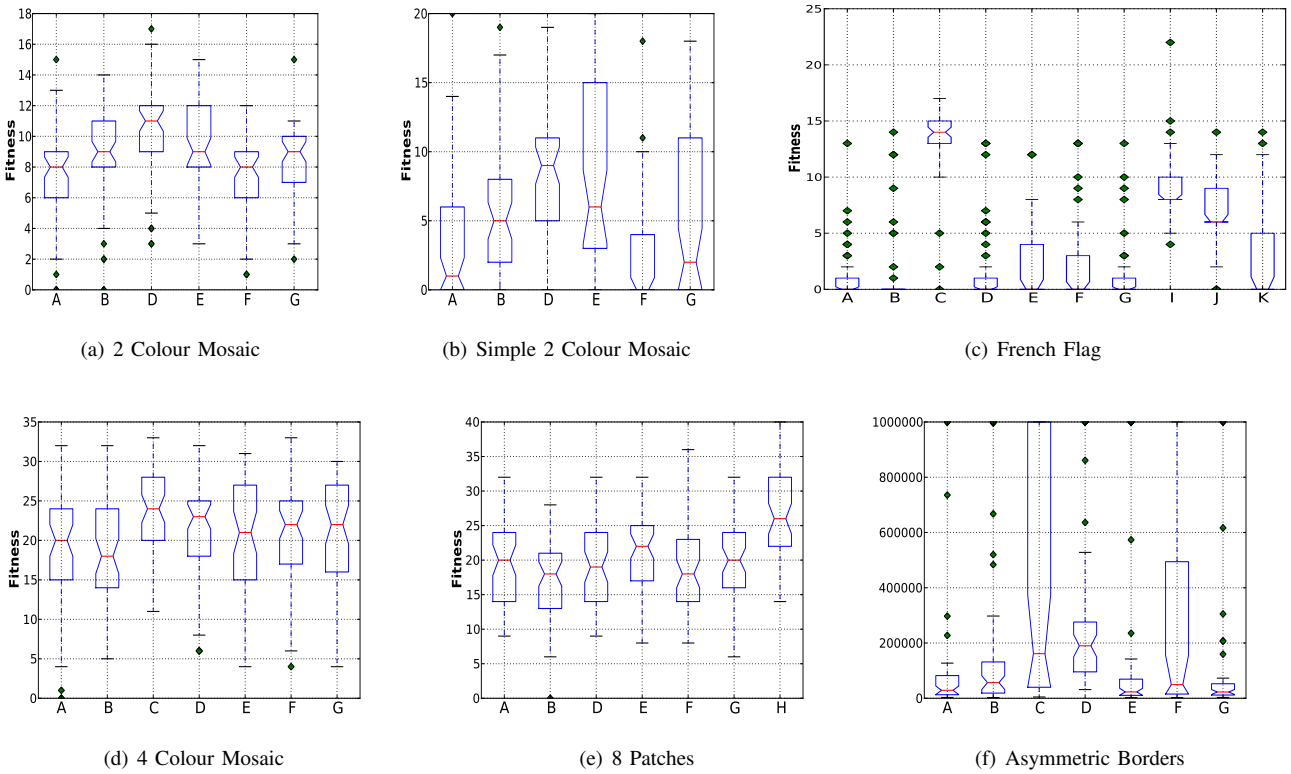
Fig. 16. The success rates achieved by the different developmental mechanisms. The aliases used in the x-axis are explained in Table IX.

voter mechanism counts the number of bound enhancers and inhibitors, and activates the gene when there are higher number of enhancing bindings or zero inhibitors. The aim of the voter decision mechanism is to provide a smoother transition between active and inactive genes rather than a sharp transition provided by a conjuctive expression.

- **Protein Consuming Genes**: Although the genes consume proteins when proteins bind to a gene, protein consuming genes use up proteins as the result of their postcondition as well. The ability to further consume a protein may allow the GRN to regulate the protein chemical levels more precisely, and create redundant behaviour for further gene degeneracy in the system.
- **Unproductive Genes**: Genes that do nothing when activated are named unproductive genes. Similar to protein consuming genes these genes may be useful for protein regulation.
- **Food reliance**: A new type of chemical that is modelled to act as artificial food for the cells in the ADS is introduced. The food chemical is required by the cells to process genes, and every time a gene is activated the food chemical is used up. The absence of food chemical prevents the activation of genes. This chemical is diffused within the organism but it is not shared between neighbouring cells via contact cell signalling. The food chemical is supplied via an outside source which is located around the organism, and the source supplies a constant amount of food chemical every developmental step to the cells on the outside borders of the organism. The food chemical can not bind the genes and it can not be produced by the active genes. In Table IX, the food reliance is given a "supply level", which is the amount of food chemical being supplied by the outside source every developmental step.

TABLE IX
THE LABELS FOR THE CHARTS AND PLOTS OF EXPERIMENTS WITH VARIOUS DEVELOPMENTAL MECHANISMS.

Reference in Plots	Mechanism
A	Original setting
B	2 local proteins
C	Voter decision mechanism
D	4 messenger molecules
E	Food reliance - supply level 255
F	Protein consuming genes
G	Unproductive Genes
H	8 messenger molecules
I	Food reliance - supply level 32
J	Food reliance - supply level 64
K	Food reliance - supply level 128



(a) 2 Colour Mosaic (b) Simple 2 Colour Mosaic (c) French Flag

(d) 4 Colour Mosaic (e) 8 Patches (f) Asymmetric Borders

Fig. 17. The box plots of the results from the experiments with the different developmental mechanisms. The aliases used in the x-axis are explained in Table IX.

1) Analysis: Unlike most of the previous experiments the inclusion of different mechanisms presented in this subsection did not have a large effect in the performance of the ADS. The patch patterns slightly benefited from the use of "local proteins", whereas the mosaic patterns suffered from their use, see Figure 16. The same is true for "protein consuming genes", which gave the "simple 2 colour mosaic" pattern a boost in performance, and deteriorated the performance of the other patterns. It is not surprising that the messenger molecules does not improve the evolvability of ADS in pattern experiments, since these chemicals are designed to adapt the ADS in a changing environment.

2) Conclusions: In conclusion, none of the mechanisms described in this section made any real improvement on the evolvability of the ADS; some of them simply increased the biases in achieving certain patterns. The best overall performance was achieved without the use of any of these mechanisms. Figure 17 illustrates the changes in the evolvability of the ADS with the new mechanisms well; most of the time the use of new mechanisms did not affect the fitness distribution (number of generations for the asymmetric borders pattern). Only changing the decision mechanism to a voter (not shown in all the plots) had really deteriorated the ADS performance.

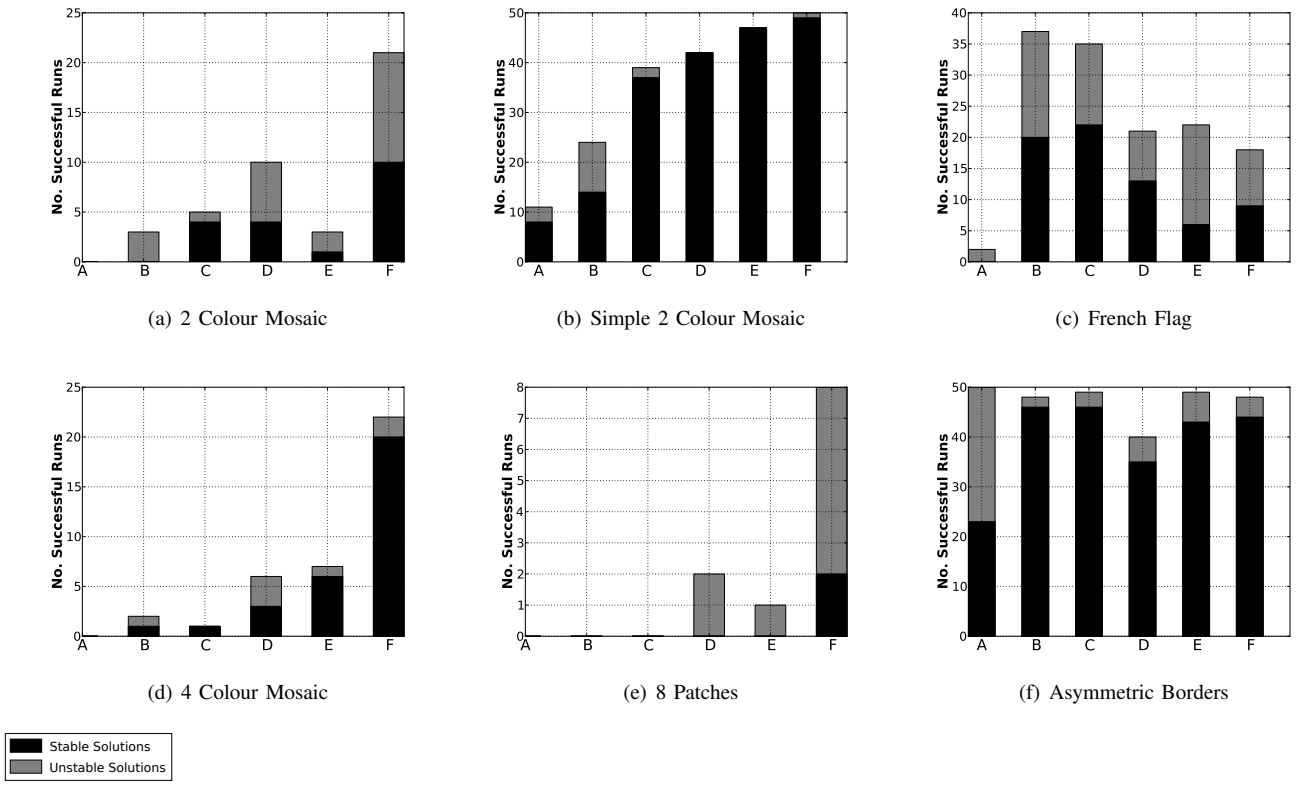
G. Improving the ADS

The previous subsections have investigated various mechanisms and developmental parameters that affect the evolvability of a bio-inspired GRN based multicellular ADS in mimicking the organisation of simple ordered patterns. It was shown that several of these mechanisms (e.g. cell signalling, adaptive parameters, etc.) had drastic effects on the evolvability of the multicellular ADS. Table XI summarises the results of these experiments.

In this section the best of the mechanisms that have significant effects on the evolvability of the ADS are shown side by side along with the original design of the ADS, and with a combination of all these "best" mechanisms. The aim is to obtain an idea on the amount of effect one mechanism may have on the evolvability of the ADS in comparison to the others, as well as demonstrating the amount of improvement that can be achieved via the use of correct combination of developmental mechanisms in an ADS.

Figure 18 displays the number of successful runs (as well as the stable ones) out of fifty runs for each version of ADS in solving the six different patterns shown in Figure 5. "Original setting" in the charts refer to the initial design of the developmental system; "simple tunnels", constant diffusion, pre-set chemical vs gene interaction constants. All the other experiments use "controlled tunnels", and have only one other mechanism implemented differently from the "controlled tunnels" experiments. In a final case all the improved mechanisms are used as part of the ADS.

1) Analysis: The mosaic and the 8 patches are the patterns that generally benefit from the combination of all the improved mechanisms. In most cases the initial design of the



(a) 2 Colour Mosaic (b) Simple 2 Colour Mosaic (c) French Flag

(d) 4 Colour Mosaic (e) 8 Patches (f) Asymmetric Borders

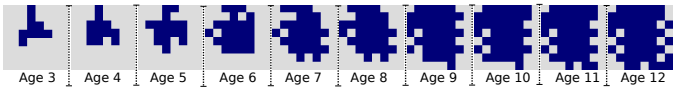
Fig. 18. Combination of all the best mechanisms compared to the "original setting" and each best mechanism. The aliases used in the x-axis are explained in Table X.

ADS was unable to find any perfectly matching patterns in fifty runs, and for the patterns it did, the success rate was low (except asymmetric borders pattern). French flag pattern benefited mainly from a single different implementation of a mechanism –the "controlled tunnels" implementation of contact signalling–, and all the other "improved" mechanisms lowered the performance of the ADS in achieving French flag patterns. Asymmetric borders pattern was generally not affected by the changes made to the ADS, however the ADS achieved more stable asymmetric borders patterns with the use of "controlled tunnels" contact signalling. The most complex patterns – 8 patches, 4 colour mosaic and 2 colour mosaic– benefited from the combination of all the "improved" mechanisms greatly, obtaining an improvement of 100-400% from the best case achieved among the other experimental runs. Using the combination of all the improved mechanisms, stable solutions were found for the 8 patches pattern for the first time. The development of few of the evolved organisms from the final experiments are displayed in Figure 19. The illustrated examples show an example developmental path to obtaining the desired patterns, as a contrast the developmental path of the organisms with the lowest fitness scores are shown as well. Figure 20 shows the developmental path of organisms obtaining a median fitness score, to provide an idea about what an imperfect solution generally looked like. The development of organisms that match the asymmetric borders and simple 2 colour mosaic patterns do not exist in Figure 20 because the median fitness scored achieved in the experiments for these patterns are 0, i.e. perfect match.

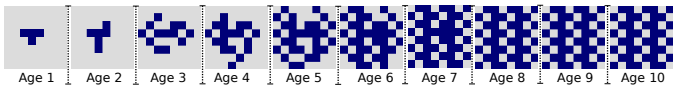
TABLE X
THE LABELS FOR THE CHARTS DISPLAYING THE RESULTS FOR EXPERIMENTS WITH THE BEST MECHANISMS.

Reference	Mechanism
A	Original setting
B	Controlled tunnels
C	Evolved binding thresholds
D	Evolved chemical production and consumption rates per gene
E	Diffusion protein
F	Combination of B,C,D,E

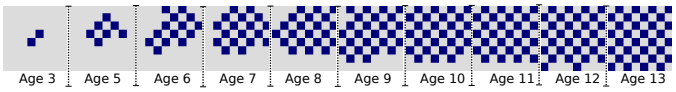
2) Conclusions: It seems that the contact signalling is the most important multicellular developmental mechanism investigated here, and a careful design of contact signalling is important in obtaining an evolvable ADS. However contact signalling alone is not sufficient for a flexible ADS in tackling evolutionary computation problems (even small patterns!), and a careful design of other mechanisms is as important. The importance of distant signalling, such as diffusion, should be more apparent in problems that require the use of larger number of cells. Also, the important role of contact signalling should not diminish even when larger organisms are developed, since localized organisation of cells would almost always be required. It is essential when solving unknown problems that the ADS is not biased on a specific type of multicellular ordering, and it is equally evolvable for all evolutionary



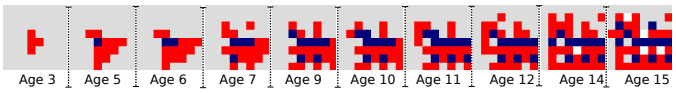
(a) Development of the worst 2 colour mosaic pattern with 13 incorrect pixels.



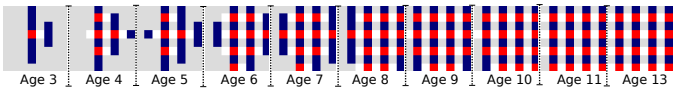
(b) Development of a perfect matching 2 colour mosaic pattern.



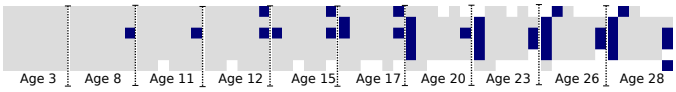
(c) Development of a perfect matching simple 2 colour mosaic pattern.



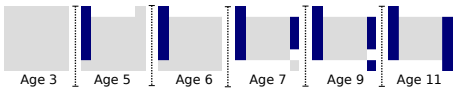
(d) Development of the worst 4 colour mosaic pattern with 25 incorrect pixels.



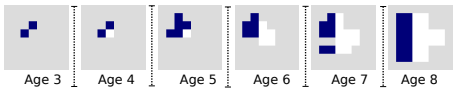
(e) Development of a perfect matching 4 colour mosaic pattern.



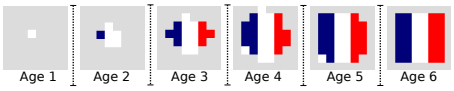
(f) Development of the worst asymmetric borders pattern with 5 incorrect pixels.



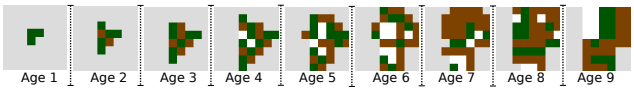
(g) Development of a perfect matching asymmetric borders pattern.



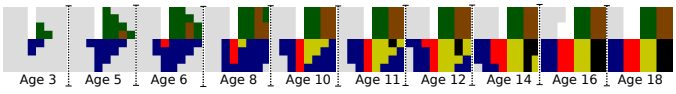
(h) Development of the worst bordered French flag pattern with 12 incorrect pixels.



(i) Development of a perfect matching bordered French flag pattern.



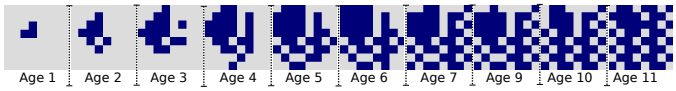
(j) Development of the worst 8 patch pattern with 32 incorrect pixels.



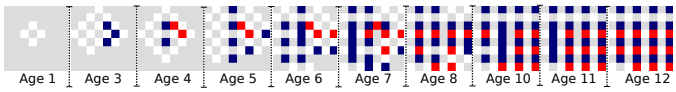
(k) Development of a perfect matching 8 patch pattern.

Fig. 19. The development of one of the worst and the best organisms from the experiments with the improved ADS (case F of this section). All of the organisms evolved can match the simple 2 colour mosaic pattern perfectly, therefore only one of the organisms are displayed.

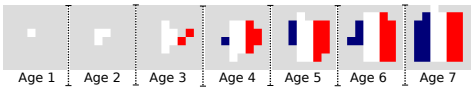
computation problems.



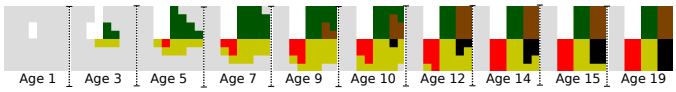
(a) 2 colour mosaic pattern with 2 incorrect pixels.



(b) 4 colour mosaic pattern with 5 incorrect pixels.



(c) French flag pattern with 2 incorrect pixels.



(d) 8 patch pattern with 8 incorrect pixels.

Fig. 20. The development of one of the organisms that achieve median fitness scores in matching the respective patterns they are evolved to match.

H. Limitations of the Current Results and Scope for Future Work

The experiments show that a version of an ADS may perform well on certain problems and equally poorly on others. This strongly suggests that using a single class of experimental problems for the empirical validation of an evolutionary system is not sufficient and often misleading. Although, a variety of organisational patterns were used for the investigations presented, there are aspects of artificial development that are not covered and the results obtained here can not be generalized. The aspects of development not covered by the current work include fault tolerance, adaptivitiy, and development of dynamic phenotypes. Investigation of the highlighted mechanisms on the evolvability of developmental systems exhibiting such properties would be an important step in generalizing the results obtained here. The fact that patterns of different ordering provided different results in some of the experiments strongly suggest that it is important to repeat these experiments on other problem domains as well. Although this will be time consuming, and probably a repetition of many of the conclusions obtained here is likely, such an undertaking is worthy for the certainty and solid understanding of the developmental mechanisms and parameters in EC.

Few of the important mechanisms and parameters in multicellular development were explored by the investigations provided in this paper, but many mechanisms were left out and even more mechanisms and parameters that need detailed investigations were found. Growth and cell division, cell movement and adhesion, importance of the order of cell processing in a developmental step, the use of signalling pathways, long distance cell signalling (i.e. auxins and endocrine) are some of the important and only partially explored mechanisms that can have a significant effect on the performance of an ADS, and these mechanisms would be worth investigating in future. It was also found in the experiments provided in this paper that

TABLE XI
SUMMARY OF THE INVESTIGATIONS DONE ON THE EVOLVABILITY OF A MULTICELLULAR ADS IN FORMING VARIOUS PATTERNS.

Mechanism	Result
Contact Signalling	It was shown that contact signalling is an essential part of an ADS, and its absence made pattern formation almost impossible for some examples. It is also important to design a contact signalling mechanism carefully, in order to allow the ADS to be able to control the intensity of signalling.
Diffusion	It was shown that both the lack of and constant presence of diffusion is undesirable. The best overall performance was obtained via a simple method that gave ADS control for the diffusion process without the need for more complicated mechanisms.
Gene-Chemical Interactions	The optimal values for ADS parameters were shown to be problem dependent. Most of the time a non optimal combination of these parameters was shown to be undesirable. Evolving these parameters during the evolution of the genotype of the ADS provided the best overall performance.
Extra Chemicals	The use of messenger molecules were introduced to monitor the phenotypes of the neighbouring cells. Local proteins were introduced to regulate the dynamics within a cell without the direct interaction of other cells, and food chemical was introduced to create extra level of control on the number of active genes. All these chemicals provide alternative ways of guiding the ADS by gene regulation, but none showed any real improvement in the overall performance of the ADS.
Mapping Development	It was observed that a cell has more information in the oscillations of its chemicals' concentrations that arise during gene interactions rather than the concentration of its chemicals at the end of a developmental step. Thus it was concluded that using the final concentrations of chemicals to map the cell phenotype is inefficient.
Decisions	Only two different decision mechanisms for activating a gene were compared: a conjunctive expression and a voter. There were large negative effects of using a voter decision mechanism when compared to a conjunctive expression. It would be worth investigating the effects of a larger number of mechanisms.
Chemical Control	The results showed that uncontrolled supply of chemicals via diffusers or constant diffusion is not desired, and in most cases the performance of the ADS deteriorates under such conditions. Protein consuming genes and unproductive genes were introduced as redundant controls of chemical supplies. But the results showed that the ADS did not need need these mechanisms.

a detailed investigation of the decision mechanisms used in the rule based GRN models and the techniques of mapping a developmental system to a phenotype may provide important advancements in the use of ADS in EC.

It would be also worth investigating and including fitness function independent evolvability measures into future investigations as well as fitness function dependent measures for even more reliable results and conclusions. Jin and Trommler developed a fitness function independent evolvability measure for GRN based ADSs, where the GRN is modelled using differential equations [82]. It should be possible to extend the method presented for developmental systems using rule-based GRN models as well.

VII. CONCLUSIONS

A new method of classification was introduced in the background section in order to provide a clearer distinction in the existing artificial implementations of development. This new method divides ADSs into two categories depending on their sources of inspiration: macro-modelling was used to refer to ADSs that model the overall behaviour of biological development (taking a "high-level" view), and micro-modelling was used to refer to ADSs that model biological development at a small scale (taking a "low-level" view), i.e. modelling cells and their interactions that lead to multicellular development. A GRN based multicellular ADS was introduced in detail with a description of the design process, using relevant literature from biology and EC as a guide. Several mechanisms and design constraints were embodied as part of the ADS during the design process. But the logic behind these decisions was merely an interpretation of the biological development and the artificial development in EC. Although the literature on multicellular ADS is rich with various models and impressive demonstrations on the capabilities of ADSs, the understanding of the importance of mechanisms and parameters is still poor.

Consequently, a detailed investigation into some of the poorly understood mechanisms and design constraints followed the introduction of the ADS. These investigations were undertaken with the aim of optimizing the presented ADS as well as providing a better understanding of some of the developmental mechanisms. 2D patterns were used as the experimental problems. Such 2D patterns are easy to understand, implement, and make a distinction between different organisational ordering. Using problems of different orders proved to be important throughout the experiments, as patterns with different types of orders usually ended up providing different results. This was an important lesson in choosing the problem-set carefully for empirical investigations on the properties of a system. An example of this was demonstrated by the experiments on the effects of constant diffusion; one of the patterns strongly favoured constant diffusion, three of the patterns produced acceptable results with it, and two of the patterns produced poor results. These results as well as the ones observed from the "chemical-gene interaction parameters", highlighted that most of the time the optimal parameters, and well performing mechanisms are highly problem dependent. The best mechanism/parameter value was found to be the most flexible one that provided the best overall performance, but not necessarily the best performance for each problem.

In almost all the experiments it was observed that an ADS is more evolvable when evolution has more control over it. The ADS was the most evolvable:

- When its parameters were included in genotype being evolved.
- With a diffusion cell signalling mechanism, where the diffusion rate could be adjusted by the ADS; indirectly giving control to evolution.
- With a contact cell signalling mechanism, where the ADS controlled the flow of chemicals; again indirectly giving control to evolution.

A large number of experiments (of over 40 sets of 50 runs) were done with six different patterns. Each run took approximately an hour of computational time on the single core of a 2.83GHz Intel Q9550 CPU based unix system. This totalled up to approximately 12000 hours of CPU time for

all the investigations presented in this paper. The experiments presented in this paper were run on a Unix based cluster PC that was built of 10 Intel Q9550 CPUs and 80GB of memory.

The conclusions obtained here apply on the evolvability of a micro-model developmental system to achieve multicellular organization. Hence they provide valuable information on the effects of several important mechanisms in cellular organization. The investigations provided here should inform researchers about the relative importance of developmental mechanisms and parameter choices. The data and conclusions provided in this work can be used for the design and optimization of a developmental system, or used for future investigations probing other problem domains. Albeit focused, the results obtained still provide a valid direction on which parameters and mechanisms have large effects on the evolvability of a micro-model developmental system. Understanding the capabilities of the mechanisms that accompany ADSs is an important step towards harnessing the full potential of ADSs for designing and optimizing engineering problems. With the advancing technology and a better understanding of what is needed from evolutionary computation, ADS has the potential to be used in the evolutionary design of real life systems.

ACKNOWLEDGEMENT

The authors would like to thank James Walker for his assistance with the use of the unix cluster. This work is part of a project that is funded by EPSRC - EP/E028381/1.

REFERENCES

[1] K. Fleischer and A. H. Barr, "A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis," in *Third Artificial Life Workshop*, Santa Fe, New Mexico, USA, June 1993, pp. 389–416.

[2] F. Dellaert and R. Beer, *Toward an Evolvable Model of Development for Autonomous Agent Synthesis*. MIT Press Cambridge, 1994.

[3] A. Devert, N. Bredeche, and M. Schoenauer, "Robust multi-cellular developmental design," in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2007, pp. 982–989.

[4] P. C. Haddow and J. Hoye, "Achieving a simple development model for 3d shapes: Are chemicals necessary?" in *Proc. of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO)*. New York, NY, USA: ACM, 2007, pp. 1013–1020.

[5] P. Haddow and J. Hoye, "Investigating the effect of regulatory decisions in a development model," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May 2009, pp. 293–300.

[6] K. O. Stanley and R. Miikkulainen, "A taxonomy for artificial embryogeny," *Artif. Life*, vol. 9, no. 2, pp. 93–130, 2003.

[7] T. Steiner, Y. Jin, and B. Sendhoff, "A cellular model for the evolutionary development of lightweight material with an inner structure," in *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2008, pp. 851–858.

[8] N. Flann, H. Jing, M. Bansal, V. Patel, and G. Podgorski, "Biological development of cell patterns : Characterizing the space of cell chemistry genetic regulatory networks," in *8th European conference, ECAL*. Springer Berlin / Heidelberg, 2005.

[9] T. Kuyucu, M. Trefzer, J. F. Miller, and A. M. Tyrrell, "On the properties of artificial development and its use in evolvable hardware," in *IEEE Symposium Series on Computational Intelligence - IEEE SSCI 2009*, 2009, pp. 108–115.

[10] L. Wolpert, R. Beddington, T. M. Jessell, P. Lawrence, E. M. Meyerowitz, and J. Smith, *Principles of Development*. Oxford University Press, 2002.

[11] J. T. Bonner, "The origins of multicellularity," *Integrative Biology: Issues, News, and Reviews*, vol. 1, no. 1, pp. 27–36, 1998.

[12] A. Chavoya, *Foundations of Computational, Intelligence Volume 1*, ser. Studies in Computational Intelligence. Springer Berlin / Heidelberg, 2009, vol. 201/2009, ch. Artificial Development, pp. 185–215.

[13] P. Eggenberger, "Evolving morphologies of simulated 3d organisms based on differential gene expression," in *Proceedings of 4th European Conference on Artificial Life*, 1997, pp. 205–213.

[14] S. L. Harding, J. F. Miller, and W. Banzhaf, "Self-modifying cartesian genetic programming," in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2007, pp. 1021–1028.

[15] S. Kumar and P. Bentley, Eds., *On Growth, Form and Computers*. Elsevier Academic Press, 2003.

[16] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines*, vol. 8, no. 2, pp. 131–162, 2007.

[17] G. Tufte, "The discrete dynamics of developmental systems," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May 2009, pp. 2209–2216.

[18] J. F. Miller, "Evolving a self-repairing, self-regulating, french flag organism," in *Genetic and Evolutionary Computation GECCO*. Springer Berlin / Heidelberg, 2004, pp. 129–139.

[19] D. Roggen, "Multi-cellular reconfigurable circuits: Evolution morphogenesis and learning," Ph.D. dissertation, EPFL, 2005.

[20] W. Banzhaf, G. Beslon, S. Christensen, J. A. Foster, F. Kepes, V. Lefort, J. F. Miller, M. Radman, and J. J. Ramsden, "Guidelines: From artificial evolution to computational evolution: a research agenda," *Nature Reviews Genetics*, vol. 7, pp. 729–735, 2006.

[21] P. C. Haddow and G. Tufte, "Bridging the genotype-phenotype mapping for digital fpgas," in *EH '01: Proceedings of the The 3rd NASA/DoD Workshop on Evolvable Hardware*. Washington, DC, USA: IEEE Computer Society, 2001, p. 109.

[22] T. Kalganova, "Bidirectional incremental evolution in extrinsic evolvable hardware," in *The Second NASA/DoD workshop on Evolvable Hardware*, J. Lohn, A. Stoica, and D. Keymeulen, Eds. Palo Alto, California: IEEE Computer Society, 13-15 Jul. 2000, pp. 65–74.

[23] J. R. Koza, *Genetic programming II: automatic discovery of reusable programs*. Cambridge, MA, USA: MIT Press, 1994.

[24] M. Murakawa, S. Yoshizawa, I. Kajitani, T. Furuya, M. Iwata, and T. Higuchi, "Hardware evolution at function level," in *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, 1996, pp. 62–71.

[25] J. Torresen, "A divide-and-conquer approach to evolvable hardware," in *ICES '98: Proceedings of the Second International Conference on Evolvable Systems*. London, UK: Springer-Verlag, 1998, pp. 57–65.

[26] V. K. Vassilev and J. F. Miller, "Scalability problems of digital circuit evolution: Evolvability and efficient designs," in *EH '00: Proceedings of the 2nd NASA/DoD workshop on Evolvable Hardware*. Washington, DC, USA: IEEE Computer Society, 2000, pp. 55–64.

[27] J. Walker and J. Miller, "Evolution and acquisition of modules in cartesian genetic programming," in *EuroGp*, vol. 3003/2004. Springer Berlin / Heidelberg, 2004, pp. 187–197.

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

[29] J. F. Miller and P. Thomson, "Cartesian genetic programming," in *Genetic Programming, Proceedings of EuroGP'2000*. Springer-Verlag, 2000, pp. 121–132.

[30] P. Bentley and S. Kumar, "Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem," in *Proc. of the Genetic and Evolutionary Computation Conf.* Orlando, Florida, USA: Morgan Kaufmann, 1999, pp. 35–43.

[31] T. G. W. Gordon, "Exploiting development to enhance the scalability of hardware evolution," Ph.D. dissertation, University College London, July 2005.

[32] P. C. Haddow, G. Tufte, and P. van Remortel, "Shrinking the genotype: L-systems for EHW?" in *ICES*, 2001, pp. 128–139.

[33] J. F. Miller and P. Thomson, "A developmental method for growing graphs and circuits," in *Evolvable Systems: From Biology to Hardware, 5th International Conference*, 2003, pp. 93–104.

[34] V. Braitenberg, "Brain size and number of neurons: An exercise in synthetic neuroanatomy," *Journal of Computational Neuroscience*, vol. 10, no. 1, pp. 71–77, 2001.

[35] J.-M. Claverie, "What if there are only 30,000 human genes?" *Science*, vol. 291, no. 5507, pp. 1255–1257, 2001.

[36] S. Kauffman, *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford University Press, 1996.

[37] T. Kuyucu, M. Trefzer, J. F. Miller, and A. M. Tyrrell, "A scalable solution to n-bit parity via articial development," in *5th International Conference on Ph.D. Research in Microelectronics & Electronics*, 2009.

[38] S. Harding, J. Miller, and W. Banzhaf, "Developments in cartesian genetic programming: Self-modifying cgp," *Genetic Programming and Evolvable Machines*, vol. 11, pp. 397–439, 2010.

[39] O. Leyser and S. Day, *Mechanisms in Plant Development*. Blackwell, 2003.

[40] H. R. Bode, "Head regeneration in hydra," *Developmental dynamics*, vol. 226, pp. 225–236, 2003.

[41] P. K. Lala, *Self-checking and fault-tolerant digital design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[42] G. M. Edelman and J. A. Gally, "Degeneracy and complexity in biological systems." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 24, pp. 13 763–13 768, November 2001. [Online]. Available: http://dx.doi.org/10.1073/pnas.231499798

[43] A. C. Pearce, Y. A. Senis, D. D. Billadeau, M. Turner, S. P. Watson, and E. Vigorito, "Vav1 and vav3 have critical but redundant roles in mediating platelet activation by collagen," *Biological Chemistry*, vol. 279, pp. 53 955–53 962, 2004.

[44] P. Bentley, "Investigations into graceful degradation of evolutionary developmental software," *Natural Computing: an international journal*, vol. 4, no. 4, pp. 417–437, 2005.

[45] T. Reil, "Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny," in *ECAL '99: Proceedings of the 5th European Conference on Advances in Artificial Life*. London, UK: Springer-Verlag, 1999, pp. 457–466.

[46] D. Federici, "Using embryonic stages to increase the evolvability of development," in *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*. Springer Verlag, 2004.

[47] H. Liu, J. F. Miller, and A. M. Tyrrell, "Intrinsic evolvable hardware implementation of a robust biological development model for digital systems," in *EH '05: Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 87–92.

[48] A. Lindenmayer, "Mathematical models for cellular interactions in development. i. filaments with one-sided inputs." *Journal of Theoretical Biology*, pp. 280–299, 1968.

[49] H. Kitano, "Building complex systems using developmental process: An engineering approach," in *ICES '98: Proceedings of the Second International Conference on Evolvable Systems*. London, UK: Springer-Verlag, 1998, pp. 218–229.

[50] E. J. Boers and H. Kuiper, "Biological metaphors and the design of modular artificial neural networks," Master's thesis, Leiden University, 1992.

[51] G. S. Hornby and J. B. Pollack, "The advantages of generative grammatical encodings for physical design," in *In Congress on Evolutionary Computation*. IEEE Press, 2001, pp. 600–607.

[52] K. Sims, "Evolving 3d morphology and behavior by competition," *Artif. Life*, vol. 1, no. 4, pp. 353–372, 1994.

[53] F. Gruau, "Neural network synthesis using cellular encoding and the genetic algorithm." Ph.D. dissertation, Ecole Normale Supirieure de Lyon, France, 1994.

[54] N. Jakobi, "Harnessing morphogenesis," in *International Conference on Information Processing in Cells and Tissues*, 1995, pp. 29–41.

[55] H. Kitano, "A simple model of neurogenesis and cell differentiation based on evolutionary large-scale chaos," *Artif. Life*, vol. 2, no. 1, pp. 79–99, 1995.

[56] S. Kumar and P. J. Bentley, "Biologically plausible evolutionary development," in *In Proc. of ICES 03, the 5th International Conference on Evolvable Systems: From Biology to Hardware*, 2003, pp. 57–68.

[57] G. Tempesti, D. Mange, E. Petraglio, A. Stauffer, and Y. Thoma, "Developmental processes in silicon: An engineering perspective," in *EH '03: Proceedings of the 2003 NASA/DoD Conference on Evolvable Hardware*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 255–264.

[58] S. Wolfram, *A New Kind of Science*. Champaign, IL, USA: Wolfram Media, 2002.

[59] J. F. Miller, "Evolving developmental programs for adaptation, morphogenesis, and self-repair," in *7th European Conference on Artificial Life*. Springer LNAI, 2003, pp. 256–265.

[60] R. Dawkins, *On Growth, Form and Computers*. Elsevier Academic Press, 2003, ch. The evolution of Evolvability, pp. 239–255.

[61] E. Davidson, *The Regulatory Genome: Gene Regulatory Networks In Development And Evolution*, 1st ed. ACADEMIC PRESS, 2006.

[62] S. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, March 1969. [Online]. Available: http://dx.doi.org/10.1016/0022-5193(69)90015-0

[63] T. Steiner, L. Schramm, Y. Jin, and B. Sendhoff, "Emergence of feedback in articial gene regulatory networks," in *IEEE Congress on Evolutionary Computation (CEC 2007)*. IEEE, 2007, pp. 867–874.

[64] E. Stomeo, C. Lambert, N. Lipnitsakya, and Y. Yatskevich, "On evolution of relatively large combinational logic circuits," in *EH '05: Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 59–66.

[65] J. A. Walker, J. F. Miller, and R. Cavill, "A multi-chromosome approach to standard and embedded cartesian genetic programming," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2006, pp. 903–910.

[66] B. Arad and A. El-Amawy, "Robust fault tolerant training of feedforward neural networks," in *Circuits and Systems, 1994., Proceedings of the 37th Midwest Symposium on*, Lafayette, LA, USA, 1994, pp. 539–544.

[67] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, 2002.

[68] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, January 2002. [Online]. Available: http://dx.doi.org/10.1089/10665270252833208

[69] F. Fagotto and B. M. Gumbiner, "Cell contact-dependent signalling," *Developmental Biology*, vol. 180, pp. 445–454, 1996.

[70] T. Berleth and T. Sachs, "Plant morphogenesis: long-distance coordination and local patterning," *Current Opinion in Plant Biology*, vol. 4, pp. 57–62, 2001.

[71] G. Tufte and P. Haddow, "Identification of functionality during development on a virtual sblock fpga," in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, vol. 1, Dec. 2003, pp. 731–738 Vol.1.

[72] S. Zhan, J. F. Miller, and A. M. Tyrrell, "A developmental gene regulation network for constructing electronic circuits," in *8th International Conference on Evolvable Systems: From Biology to Hardware*. Springer-Verlag, 2008, pp. 177–188.

[73] ——, "An evolutionary system using development and artificial genetic regulatory networks for electronic circuit design," *BioSystems*, vol. 93, no. 3, pp. 176–192, December 2009.

[74] M. Trefzer, T. Kuyucu, J. Miller, and A. Tyrrel, "Evolution and analysis of a robot controller based on a gene regulatory network," in *The 9th International Conference on Evolvable Systems: From Biology to Hardware*, 2010.

[75] M. A. Trefzer, T. Kuyucu, J. F. Miller, and A. M. Tyrrel, "Image compression of natural images using artificial gene regulatory networks," in *GECCO'10*, 2010.

[76] A. M. Tyrrell and H. Sun, "A honeycomb development architecture for robust fault-tolerant design," in *AHS '06: Proceedings of the first NASA/ESA conference on Adaptive Hardware and Systems*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 281–287.

[77] I. Harvey and A. Thompson, "Through the labyrinth evolution finds a way: A silicon ridge," in *Proc. 1st Int.Conf.on Evolvable Systems (ICES'96)*, ser. LNCS, vol. 1259. Springer-Verlag, 1997, pp. 406–422.

[78] V. Vassilev and J. Miller, "The advantages of landscape neutrality in digital circuit evolution," in *Proceedings of the 3rd International Conference on Evolvable Systems: From Biology to Hardware*. Springer, 2000, pp. 252–26.

[79] O. Kramer, *Self-Adaptive Heuristics for Evolutionary Computation*. Springer, 2008, vol. 147.

[80] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *American Statistician*, vol. 32, pp. 12–16, 1978.

[81] A. Vargha and H. D. Delaney, "A critique and improvement of the cl common language effect size statistics of mcgraw and wong," *Journal of Educational and Behavioral Statistics*, vol. 25, no. 2, pp. 101–132, January 2000. [Online]. Available: http://dx.doi.org/10.3102/10769986025002101

[82] Y. Jin and J. Trommler, "A fitness-independent evolvability measure for evolutionary developmental systems," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, 2010, pp. 69–76.