

# Evolving heterochrony for cellular differentiation using vector field embryogeny

Till Steiner  
Honda Research Institute  
Europe GmbH  
Carl-Legien-Str. 30  
63073 Offenbach / Main,  
Germany  
till.steiner@honda-ri.de

Yaochu Jin  
Department of Computing,  
University of Surrey  
Guildford, Surrey, GU2 7XH  
yaochu.jin@surrey.ac.uk

Bernhard Sendhoff  
Honda Research Institute  
Europe GmbH  
Carl-Legien-Str. 30  
63073 Offenbach / Main,  
Germany  
bs@honda-ri.de

## ABSTRACT

Heterochrony is the evolutionary change of rates and timing of developmental processes. Many artificial evolutionary development systems have the ability to produce heterochrony implicitly. However, few systems have the ability to employ heterochronic change as direct evolutionary means, i.e. to directly change heterochronic features of the developmental program.

In this paper, we present and investigate a system that can directly modify heterochronic features of its basic dynamics. We show how heterochrony can be included in a novel developmental framework called vector field embryogeny, and investigate the benefit of direct evolution of heterochronic change for four different cellular differentiation tasks.

## Categories and Subject Descriptors

I.6.3 [Computing Methodologies]: Simulation and Modeling

## General Terms

Algorithms

## Keywords

artificial development, development, dynamic system, embryogeny, heterochrony, phase space, vector field

## 1. INTRODUCTION

Artificial development (artificial embryogeny) is a field of science based on abstracting biological developmental processes to create artifacts with biological complexity and advanced features such as robustness, self-repair, adaptivity, and evolvability in an artificial evolution context. Also, the investigation of developmental principles in a computer based system can enhance and facilitate research in Systems

Biology in that simulations are easy to perform and repeat in silico.

Artificial development is a relatively young field of research. As one of the first contributions to the field, Kitano [12] described a method to efficiently encode connection matrices for neural network evolution<sup>1</sup>. Probably because the scientific community in this area is mainly composed of computer scientists and engineers, artificial development has since then been applied mainly to problem domains such as shape and structure optimization (e.g. [19, 14, 8, 24, 27]), evolvable hardware (e.g. [28, 11]), controller design, and co-evolution of morphology and control of autonomous agents (e.g. [6, 4, 7, 20]). These works are part of a steadily increasing number of contributions to the field, of which good overviews can be found in [23] and [15].

A typical characteristic of artificial development systems is that they are based on a growth process, i.e. they create an indirect mapping from an encoding (genetic representation), to a design (phenotypic representation), which is usually achieved by a translation process, iterated over several discrete time steps. More or less inspired by biological development, individuals grow into their phenotypes by concerted cellular division, starting from a single cell.

Here, we want to investigate the temporal aspect of this process. In biology, both material acquisition and cellular communication for the interpretation of the information stored on the DNA necessitate that development spans over a certain time in the life of an organism. However, additionally, biologists have found that the developmental process uses time as an additional dimension, which can be exploited by evolution: Changes in the rates and timing of developmental processes are available for evolutionary change. This temporal plasticity is termed heterochrony [10]: different modular developmental processes, i.e. processes that have a defined onset and/or offset in time, can be shifted, skewed and combined differentially along the developmental time dimension.

In the field of artificial development, Stanley [23] defines heterochrony as one of the five dimensions to describe and compare artificial embryogeny systems, much in accordance with traditional classification methods in biology. However,

<sup>1</sup>The results of the paper have been disputed in [22]: The authors found that Kitano's encoding scheme is only advantageous for some defined starting conditions. Nevertheless, the indirect developmental characteristics of Kitano's paper remains a starting point for artificial development.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO2010 Portland, Oregon

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

these methods are also disputed by biologists: e.g., Klingenberg criticizes in his review paper on heterochrony [13] that in a biological context the concept is only understood as classification method and a label for observations, rather than as a feature which contributes to evolutionary change and species adaptation.

Few research papers in the field of artificial development present explicit evolution of time dependency of embryogenic processes. A typical example of existing work is presented by Federici and Downing [9]. The authors use *embryonic stages*, a way to complexify designs over evolutionary time. The strategy consists of creating a duplication of an evolved chromosome, which takes over the control of development after a defined number of developmental time-steps (after an embryonic stage), and is mutated with different rates as compared to the original copy. This yields a development in which embryonic states are temporarily distinct, each controlled by its own chromosome. However, this cannot be seen as heterochrony, since rate, onset or offset of developmental processes are not evolved.

Implicitly, all developmental models using a graph structure (i.e. a gene regulatory network) for the temporal control of development can exhibit heterochronic features. For example, increasing the production rate of an activating substance or lowering a threshold results in an earlier activation of a certain gene. Since such heterochronic effects depend on the basic parameters of the graph, it seems impossible to create an evolutionary graph-based development scheme, which does not employ heterochronic effects. Therefore, the evolutionary role of heterochrony is hard to assess in a graph-based developmental model.

In this contribution, we demonstrate the use of direct heterochronic change of developmental characteristics in an artificial evolution framework, and compare it to a system that does not employ heterochrony. For this purpose, we employ vector field embryogeny [25], an abstract model of development, which describes development on a level of system phase spaces. This framework allows us to conveniently change temporal characteristics of dynamics, and thus to study the effect of such changes in an evolutionary context.

The paper is structured as follows: we will firstly describe the biological concepts of allometry and heterochrony, and show the relationship between both. Then, we will present vector field embryogeny, an abstract framework for artificial development and explain how we can introduce explicit evolution of time dependency. Subsequently, we present our investigations into heterochrony for artificial development. Finally, we will conclude with a discussion of the results.

## 2. TIMING IN DEVELOPMENT ALLOMETRY AND HETEROCHRONY

Biologists use the terms heterochrony and allometry when describing temporal characteristics of developmental processes [13]. Historically, the two terms are distinguished, even though they basically describe similar process features: on the one hand, heterochrony is describing the evolutionary change in rates and timing of developmental processes. This means, an explicit view on the time axis of development is employed. On the other hand, allometry refers to time implicitly, describing covariation of size and shape of features of an embryo.

A typical example of allometric change in species is given

by the elongated central toe that forms a horse hoof, resulting from a higher growth rate as compared to the lateral toes. Heterochronic change can e.g. be observed in the development of the vertebrate eye: the initiation of eye development is earlier in lizards than in mammals. The lizard eyes are thus proportionally larger [10].

The close relationship between heterochrony and allometry becomes visible in these two examples: both describe a change in relative size of a body part, on the one hand with a focus on the size, on the other hand with a focus on the growth rate. Since a variation in the size of a body part necessarily results from a changed growth rate or changed relative timing, allometry is closely linked to heterochrony and vice versa. Thus, in this contribution, we will use the term heterochrony to refer to evolutionary adaptation of temporal characteristics of developmental processes.

Interestingly, in biology, heterochrony seems to be under direct evolutionary influence. For example, genes of *C. elegans* have been identified which control heterochrony directly [2]: mutations of these single specific genes result in a change of relative timing of cell differentiation. Furthermore, these mutations have also been observed among natural species. Thus, observing direct evolution of heterochrony in an artificial system might also allow for new insights into biological evolution.

In the following section, we describe the abstract simulation framework that we use in our simulations.

## 3. VECTOR FIELD EMBRYOGENY A NOVEL ABSTRACTION OF DEVELOPMENT

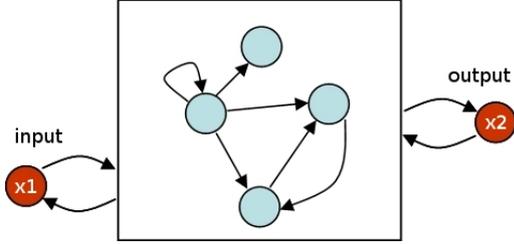
In this contribution, we employ an artificial development framework called vector field embryogeny as basis for our research. The following description of the framework is an excerpt version of the detailed information given in [25].

The dynamics of biological development are controlled by gene regulatory networks (GRNs). The dynamic behavior of these GRNs seems to account for the flexibility and robustness of biological organisms. Therefore, the most common approach to realize an artificial developmental system with such features is to model the interplay between a number of genes to create regulatory networks. The natural representation of these networks is a directed graph. Each node of such a graph represents a state variable of the system, and the links indicate modes of interaction between nodes with connection weights and more or less complex activation functions. Standard approaches toward evolving these networks for artificial development are based on evolving both, structure and weights of the networks (e.g. [26]).

In this contribution, we employ an alternative approach: vector field embryogeny shifts evolutionary focus from the structure and weights of a network to the dynamics that such a network would create, i.e., to its system phase space. Vector field embryogeny enables mutation operators to directly create and shape a system phase space (direct manipulation), instead of doing so indirectly via graph manipulation. This allows a more causal relationship between mutation and resulting changes in system dynamics. Another advantage is the possibility to include direct evolution of heterochrony into such a framework (see below). Vector field embryogeny is inspired by a method known as vector field editing [5] and will be described in the following.

### 3.1 Direct manipulation of the phase space

In computer graphics, the vector field editing method is used for creating texture alignments and extracting analytical information about given graphical representations of vector fields [5, 29, 18]. The following considerations are presented using a two-dimensional version of the system for clarity and visualization purposes. Note that the method is extendable to D dimensions by applying the respective D-dimensional geometrical operations.



**Figure 1:** An arbitrary GRN with two genes of interest:  $x_1$  and  $x_2$ . Here,  $x_1$  and  $x_2$  are the input and the output node of the network.

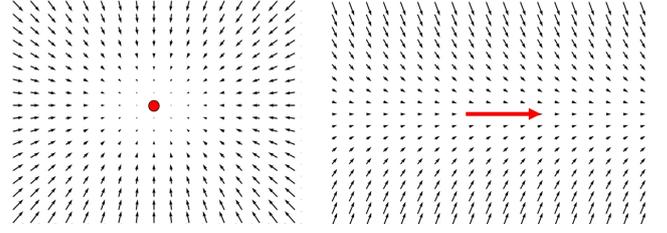
Consider an arbitrary simulated GRN inside a cell, with two genes of interest (Figure 1). We denote the state (i.e., activation level) of these two genes by  $x_1$  and  $x_2$  respectively, and together as the vector  $\mathbf{X} = (x_1, x_2)$ . The temporal behavior of any deterministic simulation of a regulatory network containing these two genes can now be described with respect to  $\mathbf{X}$  by the differential equation  $d\mathbf{X}/dt = \mathbf{F}(\mathbf{X}, \boldsymbol{\lambda}, t)$ , where  $\mathbf{F}$  is a vector field and  $\boldsymbol{\lambda}$  is a vector of parameters. In this paper, we will focus on isolated cells in constant environmental conditions, such that  $\mathbf{F} = \mathbf{F}(\mathbf{X}, \boldsymbol{\lambda})$ . Hence,  $\mathbf{F}$  describes a time independent, two dimensional vector for each system state  $\mathbf{X}$ , which represents the direction and magnitude of change in time, whenever the system reaches the state  $\mathbf{X}$ .  $\mathbf{F}(\mathbf{X}, \boldsymbol{\lambda})$  is a vector field, which is referred to as the phase space of the dynamic system [16].

Vector field editing relies on creating and changing vector fields by superposition and adaptation of basic field elements  $\mathbf{E}_i(\mathbf{X}, \boldsymbol{\lambda}_i)$ . The vector field for any system state  $\mathbf{X}$  is then given by the superposition of these elements:

$$\mathbf{F}(\mathbf{X}, \boldsymbol{\lambda}) = \sum_i \mathbf{E}_i(\mathbf{X}, \boldsymbol{\lambda}_i). \quad (1)$$

Typical elements are proposed in [5] and [29] and can be grouped into singular elements and regular elements. Singular elements are those which create a singularity in the vector field (i.e., a source or a sink) while regular elements do not contain a singularity in their description, and thus generally change the vector field without creating singularities. Two examples are depicted in Figure 2.

In our framework, we adopt the regular element formulation given in [5] and use a simplified version of singular elements. The regular element we use is called attachment element. It creates a flow of surrounding system states toward an attachment line at the center of the phase space. The mathematical formulation to create such an element, where the attachment line is oriented along an arbitrary an-



**Figure 2:** Two basic field elements are employed in the simulations: a singular element is depicted on the left panel, and a regular element (attachment element) is depicted on the right panel. Point and arrow mark the center and center line of the elements respectively.

gle  $\theta \in [0, 2\pi]$  is given by

$$\mathbf{A}(x_1, x_2) = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} - cP(x_1, x_2) \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix}. \quad (2)$$

Here,  $P(x_1, x_2) = -\sin \theta(x_1 - u_1) + \cos \theta(x_2 - u_2)$  and  $c$  is a parameter describing the speed with which the flow is attracted to the line and  $\mathbf{U} = (u_1, u_2)$  is the center position of the element. Note that for negative  $c$ , system states will diverge from the line instead of converging to it. To spatially limit the element's influence for superposition, this attachment element is multiplied by a Gaussian kernel  $B(x_1, x_2)$  of width  $2\sigma$  and center  $\mathbf{U}$ :  $B(x_1, x_2) = e^{-((x_1 - u_1)^2 + (x_2 - u_2)^2)/2\sigma}$ . Therefore, the complete formulation of the attachment element is given by

$$\mathbf{V}^R(x_1, x_2) = B(x_1, x_2) \cdot \mathbf{A}(x_1, x_2). \quad (3)$$

We create a singular element by applying

$$\mathbf{V}^S(x_1, x_2) = \begin{cases} (\mathbf{U} - \mathbf{X})/\sigma & \text{if } r < \sigma \\ (2/r - 1/\sigma) \cdot (\mathbf{U} - \mathbf{X}) & \text{if } \sigma \leq r < 2\sigma. \end{cases} \quad (4)$$

The variable  $r := \|\mathbf{X} - \mathbf{U}\|_2$  describes the distance of the system state  $\mathbf{X}$  to the center  $\mathbf{U}$  of the singular element. The width of the element is denoted by  $2\sigma$ . Formulation (4) is a coarse piecewise linear approximation of  $\mathbf{V}(x_1, x_2) = B(x_1, x_2) \cdot (\mathbf{U} - \mathbf{X})$ . We use it, since it is more efficient in computer simulations.

A superposition of  $\eta$  field elements, each weighted by a factor  $\alpha_i$ , yields an arbitrarily complex vector field, which can be interpreted as system phase space:

$$\mathbf{F}(x_1, x_2) = \sum_{i=1}^{\eta} \alpha_i \mathbf{V}_i(x_1, x_2), \quad (5)$$

where  $\alpha_i \mathbf{V}_i(x_1, x_2)$  corresponds to  $\mathbf{E}_i(\mathbf{X}, \boldsymbol{\lambda}_i)$  in Equation (1), with  $\boldsymbol{\lambda}$  consisting of all  $\mathbf{U}_i, \sigma_i, \alpha_i$  of all field elements, and additionally  $\theta_i$  and  $c_i$  of the regular elements. Thus, the vector field described in Equation (5) constitutes the right hand side of the differential equation

$$\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}, \boldsymbol{\lambda}), \quad (6)$$

which is integrated from  $t = 0$  to  $t = t_{\max}$  to yield a trajectory of the dynamic system. Evolving such a system consists

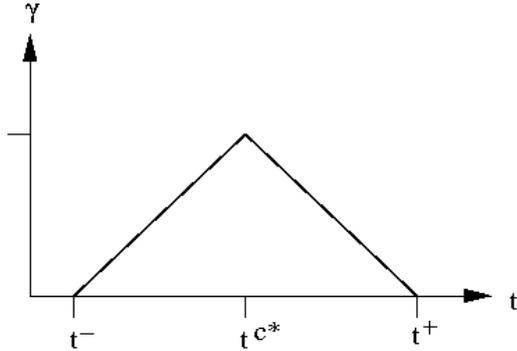
of encoding  $\lambda$  in a chromosome and employing standard evolutionary methods for the search, such as Evolution Strategies (ES) [21].

### 3.2 Heterochrony in vector field editing

Heterochronic changes are changes in the timing of developmental processes. In our interpretation, developmental processes are characterized by the influence of basic field elements on the trajectory of a system state. Thus, we interpret heterochrony as allowing evolution to choose a time frame during development, in which such an element has influence. Outside of this time frame, the element is without influence. Therefore, we propose that heterochrony can be introduced into the vector field embryogeny framework by adding a temporal dimension to each basic field element as follows: Firstly, two new variables  $t_i^c \in [0, 1]$  and  $t_i^d \in [0, 1]$  are encoded in the genome for the  $i$ -th field element. These values are scaled to fit simulation time:  $t_i^{c*} = t_i^c \cdot t_{\max}$ , and  $t_i^{d*} = t_i^d \cdot \frac{t_{\max}}{5}$ . Here,  $t_{\max}$  is the maximum simulation time.  $t_i^{c*}$  specifies the temporal center of the field element, while  $t_i^{d*}$  denotes the length of the time frame of the field element. Note that we set the maximum temporal width of an element to one fifth of the simulation time, such that a temporal separation of basic field elements is facilitated. For convenience, we define  $t^+ = t_i^{c*} + \frac{t_i^{d*}}{2}$  and  $t^- = t_i^{c*} - \frac{t_i^{d*}}{2}$ . The values  $t_i^{c*}$  and  $t_i^{d*}$  are used to create a time dependent basic field element  $\mathbf{E}_i(t) = \gamma_i(t) \cdot \mathbf{E}_i(\mathbf{X}, \lambda_i)$ , that replaces  $\mathbf{E}_i(\mathbf{X}, \lambda_i)$  in Equation (1).  $\gamma_i(t)$  is a piecewise linear function, which has its maximum value of 1 at  $t = t_i^{c*}$  (see Figure 3):

$$\gamma_i(t) = \begin{cases} \frac{2(t-t_i^{c*})}{t_i^{d*}} + 1 & \text{if } t^- < t \leq t_i^{c*} \\ \frac{2(t_i^{c*}-t)}{t_i^{d*}} + 1 & \text{if } t_i^{c*} < t \leq t^+ \\ 0 & \text{if } t \leq t^- \text{ or } t > t^+ \end{cases} \quad (7)$$

This formulation allows evolution to tune both, onset and temporal length of basic field elements explicitly. We choose



**Figure 3: The scaling factor  $\gamma$  that introduces heterochrony into the vector field embryogeny framework, has the presented piecewise linear temporal characteristics.**

the piecewise linear characteristics to ensure that small mutations of  $t_i^{c*}$  and  $t_i^{d*}$  result in smooth changes of the temporal characteristics of the phase space. Experiments with a simple on-off switch in time of basic field elements seemed detrimental for evolvability (results not shown). A more

natural activation curve than the triangular one depicted in Figure 3, such as a logistic curve of growth and decay would be straight forward to implement which would of course be less simple and less computational efficient. For employing the framework in a more biological context, the formulation should be critically assessed.

## 4. EXPERIMENTAL SETUP

For our experiments, we set up a phase space model in three dimensions,  $x$ ,  $y$  and  $z$ , constrained to the interval  $[0, 1]$  in each dimension. Thus,  $\mathbf{X} = (x, y, z)$  in Equation (1). This would correspond to a GRN where the state of three genes is observable during developmental time.

We then perform the following steps:

1. Determine the initial states of these three variables for development (in a biological context, initial values may result from environmental signals or a maternal gradient).
2. Create a three-dimensional phase space, i.e., choose  $\mathbf{V}_i(x, y, z)$ .
3. Use an evolution strategy to mutate the parameters  $\lambda$  and thereby change the vector field representation of the phase space  $\mathbf{F}(\mathbf{X}, \lambda)$ .

4. For individual  $j$ , use the differential equation

$$d\mathbf{X}^j/dt = \mathbf{F}^j(\mathbf{X}^j, \lambda^j)$$

to create time courses of the corresponding three variables to control its development.

5. Use the evolution strategy to select fitter individuals for reproduction and repeat steps 1 to 5 until a stop criterion is met.

To investigate cellular differentiation, the system state  $\mathbf{X}$  is interpreted as the expression level of three genes in a certain cell of an individual. The cells that belong to the same individual share the same phase space, but have different initializations of  $x$  and  $y$ . We define  $z$  to correspond to the cell type and initialize it at  $z_0 = 0.5$  for all cells, representing a non-differentiated state. The cell's environmental information is encoded in  $x$  and  $y$  and can be interpreted as maternal factors, similar to those found in the early *Drosophila* embryo [17]. Cells do not divide or interact; note however that both mechanisms would be possible to include in the framework (see [25]). For visualization, cells are positioned on a 2D lattice, where the coordinate of a cell is chosen according to its initial state of the genes  $x$  and  $y$ . Figures 4 and 5 and their captions explain the concept. Two different spatial resolutions are used for experiments:  $2 \times 2$  and  $4 \times 4$  cells. Therefore,  $x \in \{0, 1\}$  and  $y \in \{0, 1\}$ , or  $x \in \{0, 0.33, 0.67, 1\}$  and  $y \in \{0, 0.33, 0.67, 1\}$  for the respective experiments. The phase space of an individual is evolved by changing the key parameters of a fixed number of field elements. These key parameters for singular elements are  $\mathbf{U} = (u_1, u_2, u_3)$ ,  $\alpha$ , and  $\sigma$ .  $\mathbf{U}$  represents the position of the element in 3D space,  $\alpha$  is its strength and  $\sigma$  its width (see field element description above). For an attachment element, three additional parameters are encoded:  $\theta$ ,  $\phi$ , and  $c$ .  $\theta$  and  $\phi$  are the two angles describing the direction of the element in 3D space, and  $c$  is the relative speed of attachment (see Equation (2)). The resulting system equations are solved for each cell by a

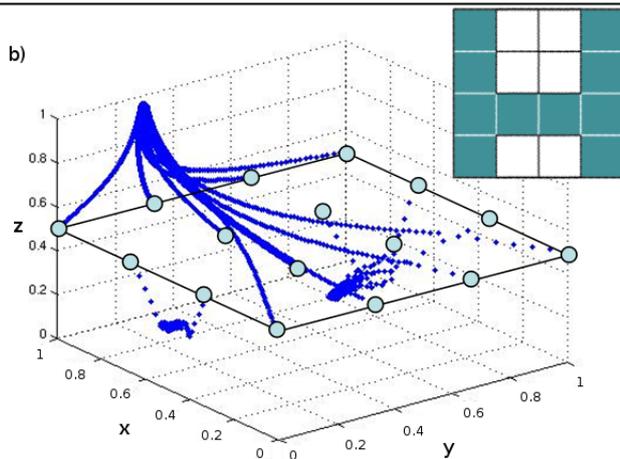
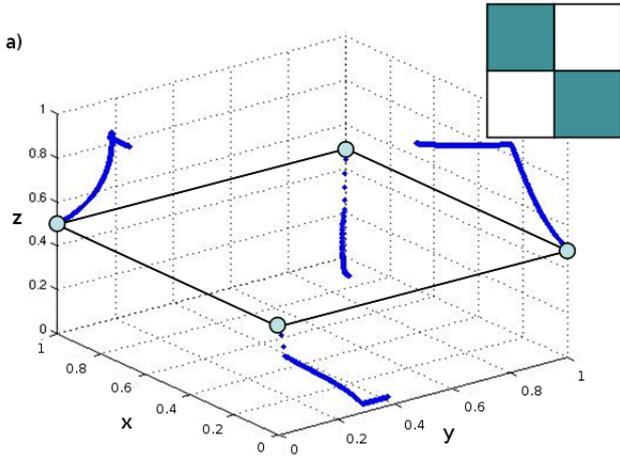


Figure 4: Two phase space plots for the experimental setups are depicted. The state of a cell is represented by a vector  $(x, y, z)$ .  $z$  denotes the cell type. Initially, all cells are undifferentiated (i.e.  $z = 0.5$ , starting points are denoted by circles). The phase space simulation yields time courses for the system states of the cells. After simulation the resulting cell types are encoded in color (dark represents a high value, bright represents a value close to 0) and projected onto the  $x$  and  $y$  position of the respective starting point (lattice representation, upper right corner of the plots). a) The target lattice and a solution to the 'xor' experiment with four cells. b) The target lattice and a solution to the 'h' experiment, employing 16 cells.

4th order Runge-Kutta method. The maximum simulation time is set to  $t_{\max} = 500s$ , with a step width of  $0.25s$  and 8 sub-iterations per step. We expect system states to have reached a stable state before the simulation time reaches 500 seconds. However, if this is not the case, solutions are not penalized. Simulation is terminated when either the maximum time  $t_{\max}$  is exceeded or when the system state does not vary more than  $\epsilon = 10^{-12}$  in two consecutive steps. A standard evolution strategy [21] is employed, with population sizes of 15 and 100 for parent and offspring population

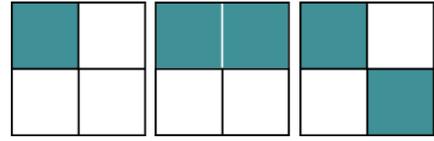


Figure 5: The three target lattices for the four cell experiments. From left to right: 'onepoint', 'half', and 'xor'.

respectively, with a single strategy parameter with step size adaptation. A more sophisticated evolution strategy could be applied ([3] gives a comprehensive overview), however, the standard version is very robust and its performance is sufficient for our purpose. The initial strategy parameter is chosen to be  $\sigma_{init} = 0.1$ . The fitness  $F$  is calculated by taking the squared distance between the cell types of the  $n$  cells belonging to an individual after development, and a given target matrix  $\rho$ :  $F = \sum_i (z_i - \rho_i)^2$ . Therefore, the task is a minimization task, and optimal fitness is reached if  $F = 0$ . The matrix  $\rho$  is chosen according to the 2D lattice representations of the targets in Figures 4 and 5: it contains a 1 where the lattice has a dark square, and a 0 where the lattice contains a white square. Thus, if the lattice contains a dark square at position  $(x_a, y_a)$ , a system initialization of  $\mathbf{X} = (x_a, y_a, 0.5)$  should optimally result in a trajectory leading to a system state  $\mathbf{X} = (*, *, 1.0)$ . Twenty evolutionary runs are performed per setup.

For the heterochrony experiments, we encode the additional parameters  $t_i^c$  and  $t_i^d$  for the  $i$ -th element in the chromosome of an individual (see section 3.2). For all targets, we compare the performance of the setup using heterochrony with a reference setup that does not employ explicit heterochronic changes, denoted as baseline setup.

## 5. RESULTS

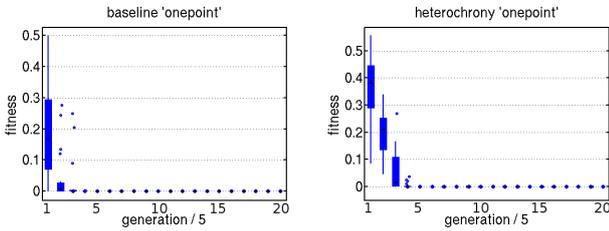
### 5.1 Four cells targets

Figure 5 gives the target lattices for the first differentiation experiment. In our previous work, the same targets have been used to evaluate optimal number and composition of basic field elements [25]. It was found that optimally, two regular elements and two singular elements are combined to fulfill this task. We therefore employed this setup for our comparisons.

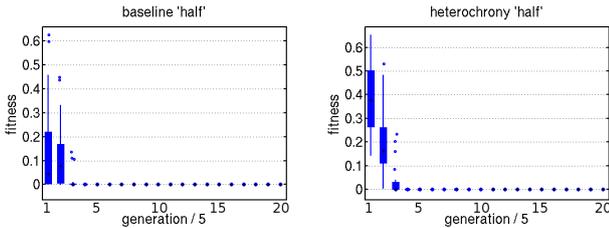
Figures 6 – 8 show the fitness distributions over evolutionary generations for the best individuals of 20 runs for each of the three experiments, employing both baseline setup (left panels) and heterochrony setup (right panels). It is clearly visible that the 'onepoint' and 'half' experiments converge quickly to an optimal solution in all runs, with no significant differences in the evolutions, except that initial populations of the heterochrony runs generally seem to have a worse fitness. Nevertheless, convergence to optimal fitness takes place for all runs in few generations.

While most of the 'xor' experiments also converge to the optimum in 100 generations, the problem is more complex to solve for vector field embryogeny. Nevertheless, vector field embryogeny seems suited for the task. It is visible that for this problem, the initial populations of the baseline and heterochrony setup have a similar fitness distribution. However,

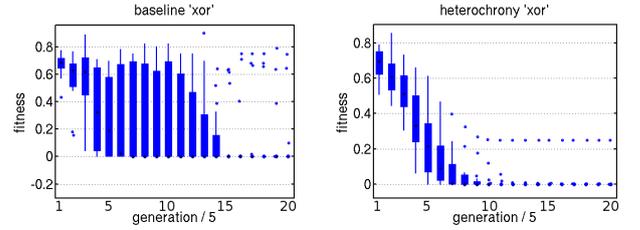
the heterochrony setup has a smoother transition in fewer generations to optimal fitness. Note that this comparison is not entirely fair: the heterochrony setup has two additional values encoded per basic field element, thus a chromosome size that is larger by 8 elements. To account for this, we repeated the baseline experiment for the 'xor' task with an increased number of basic field elements: The left panel of Figure 9 depicts the fitness distribution for an experiment employing 3 regular and two singular elements, i.e. the same number of free parameters as compared to the 2 regular + 2 singular element heterochrony setup. Interestingly, the performance decreases drastically. This could be related to an overfitting phenomenon. To complete the investigation, the right panel of Figure 9 shows the performance of an experiment employing 2 regular and 3 singular elements. It is visible that no performance increase results from the additional parameters for the evolution of the vector field. Thus, the heterochrony setup does not decrease performance of the vector field embryogeny approach for simple problems, and increases performance for the slightly more complex 'xor' problem.



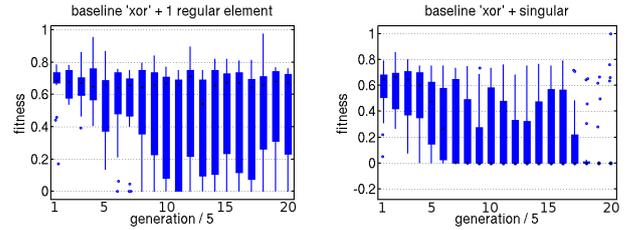
**Figure 6:** The plots show the fitness distribution of the best individuals of 20 evolutionary runs over 100 generations for the 'onepoint' experiment (generations in steps of five). The left panel depicts the result for the baseline experiment, while the right panel shows the performance of the heterochrony experiment. Boxes in the box plot denote the 25th to 75th percentile, dots are outliers. Whiskers extend to ca. 99.3% of the data.



**Figure 7:** The plots show the fitness distribution of the best individuals of 20 evolutionary runs over 100 generations for the 'half' experiment (generations in steps of five). The left panel depicts the result for the baseline experiment, while the right panel shows the performance of the heterochrony experiment.



**Figure 8:** The plots show the fitness distribution of the best individuals of 20 evolutionary runs over 100 generations for the 'xor' experiment (generations in steps of five). The left panel depicts the result for the baseline experiment, while the right panel shows the performance of the heterochrony experiment.



**Figure 9:** The plots show the fitness distribution of the best individuals of 20 evolutionary runs over 100 generations for the 'half' experiment (generations in steps of five). The baseline setup is employed for both panels. The left panel depicts performance of the experiment employing three regular and two singular elements, the right panel shows fitness distributions of the experiment employing two regular and three singular elements.

## 5.2 16 cells target

We decided to compare the heterochrony setup to the baseline setup on the more complex 'h'-task, depicted in Figure 4b). To account for the increased complexity of the problem, we increased the number of basic field elements in the heterochrony setup to 4 regular + 4 singular elements. Each experiment consists of 40 runs of 200 generations. For a fair comparison, we equipped the baseline setup with 5 regular + 5 singular elements. Note that this baseline setup outperforms a 4 regular + 4 singular elements setup (results not shown). Figures 10 and 11 show the fitness distributions of the experiments. In both approaches, 10 runs converge to the optimum fitness. The heterochrony experiment has a smaller extent of the box denoting the 25th to 75th percentile, but more runs do not converge to a fitness value below 2, as compared to the convergence of all runs in Figure 10. Figure 12 shows an enlarged box plot of the distributions of the best individuals after 200 generations. Triangular markers give the 95% confidence comparison intervals, which do not overlap. Thus, it is visible that the median of the heterochrony setup is lower than the median of the baseline setup, for the given task.

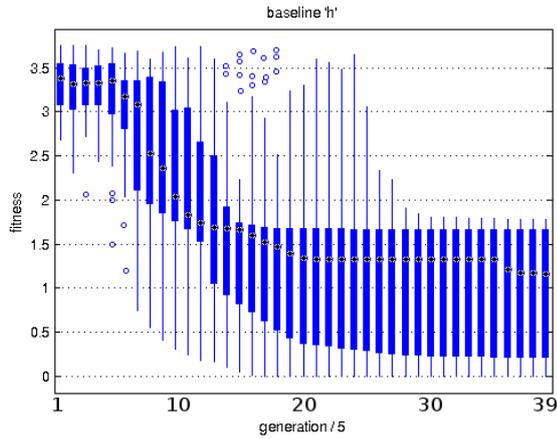


Figure 10: The fitness distribution over 199 generations of 40 runs of the 16 cells baseline experiment toward the 'h' target. For box-plot attributes, see caption of Figure 6.

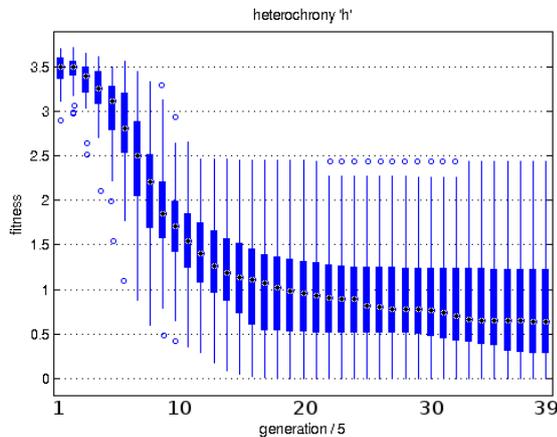


Figure 11: The fitness distribution over 199 generations of 40 runs of the 16 cells heterochrony experiment toward the 'h' target. For box-plot attributes, see caption of Figure 6.

## 6. DISCUSSION

The explicit use of the temporal dimension (heterochrony) has been observed in evolutionary biology and been used to describe phylogenetic differences. Particularly, [2] discovered genes in *C. elegans*, which have direct influence on heterochronic change in biological development.

Artificial development systems implicitly allow heterochrony to occur, but do not allow a comparison between non heterochronic and heterochronic development. Here, we have used an abstract system called vector field embryogeny, which allows us to switch between non-heterochronic and heterochronic development. Our results suggest that for the given problems, using heterochrony improves evolutionary success of nontrivial cellular differentiation tasks. Also, us-

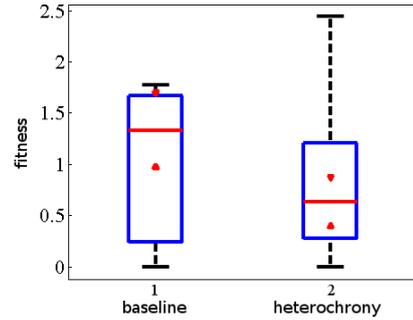


Figure 12: Comparison of the baseline and heterochrony setup. Depicted are the distributions of fitness of the best individuals of 40 runs after 200 evolutionary generations. Boxes denote the 25th to 75th percentile. Whiskers extend to ca. 99.3% of the data. Triangular markers give the comparison intervals (see text).

ing heterochrony can allow to avoid overfitting, as seen in the 'xor' experiment. Therefore, we believe it is worthwhile to consider investigations of the temporal dimension of artificial developmental systems in general, and in particular to exploit differential combination of modular control aspects of development along the time axis. Similar to vector field embryogeny, which allows for heterochronic as well as non heterochronic development, it would be interesting to find a way to more directly influence heterochrony in graph-based genetic representations, such as artificial gene regulatory networks. We think that this can be achieved by carefully tuning and investigating structural aspects of graphs, and may eventually result in finding motifs [1] that influence the timescale of graph dynamics explicitly. Apart from heterochrony, we plan to investigate the links between vector field embryogeny and graph representations in general. We believe that an approach combining an abstract, phase space centered view with a more detailed, graph centered view of gene regulatory networks could yield a better insight into the evolution of networks in biology.

Vector field embryogeny is a novel abstraction of development, such that, apart from heterochrony, additional aspects need to be investigated: cellular communication, as well as methods for complexification could be introduced to the system in many ways. Also, finding a useful classification of basic field elements, such that eventually they can be stored in some sort of element library and used specifically according to the problem that is investigated, is still an open topic.

## 7. ACKNOWLEDGMENTS

We would like to thank the reviewers for valuable comments and suggestions on the manuscript. The work was conducted while Y. Jin was with the Honda Research Institute Europe GmbH.

## 8. REFERENCES

- [1] U. Alon. *An Introduction to Systems Biology - Design Principles of Biological Circuits*. Chapman and Hall/CRC, 2006.
- [2] V. Ambros. Heterochronic genes. In D. Riddle, T. Blumenthal, B. Meyer, and J. Priess, editors, *C. elegans II*. Cold Spring Harbor Laboratory Press, 1997.
- [3] H. Beyer and H. Schwefel. Evolution Strategies – a comprehensive introduction. *Natural Computing*, 1:3–52, 2002.
- [4] J. Bongard and R. Pfeifer. Evolving complete agents using artificial ontogeny. In F. Hara and R. Pfeifer, editors, *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, pages 237–258. Springer, 2003.
- [5] G. Chen, K. Mischaikow, R. Laramée, P. Pilarczyk, and E. Zhang. Vector field editing and periodic orbit extraction using morse decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):1077–2626, 2007.
- [6] D. d’Ambrosio and K. Stanley. Generative encoding for multiagent learning. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2008.
- [7] F. Dellaert and R. Beer. A developmental model for the evolution of complete autonomous agents. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 393–401, 1996.
- [8] A. Devert, N. Bredeche, and M. Schoenauer. Artificial ontogeny for truss structure design. In *Proceedings of the SASOW, Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pages 298–305, 2008.
- [9] D. Federici and K. Downing. Evolution and development of a multicellular organism: Scalability, resilience, and neutral complexification. *Artificial Life*, 12:381–409, 2006.
- [10] S. Gilbert, editor. *Developmental Biology*. Sinauer Associates Inc., 2003.
- [11] P. Haddow, G. Tufte, and P. van Remortel. Shrinking the genotype: L-systems for ehw? In *Proceedings of the ICES 2001*, volume 2210 of *Lecture Notes on Computer Science*, 2001.
- [12] H. Kitano. Designing neural networks using genetic algorithm with graph generation system. *Complex Systems*, 4:461–476, 1990.
- [13] C. Klingenberg. Heterochrony and allometry: the analysis of evolutionary change in ontogeny. *Biol. Rev.*, 73:79–123, 1998.
- [14] T. Kowaliw. *A Good Number of Forms Fairly Beautiful*. PhD thesis, Department of Computer Science, Concordia University, Montreal, Canada, 2007.
- [15] S. Kumar and P. Bentley, editors. *On Growth, Form and Computers*. Elsevier, 2003.
- [16] G. Nicolis. *Introduction to Nonlinear Science*. Cambridge University Press, 1995.
- [17] C. Nüsslein-Volhard and E. Wieschaus. Mutations affecting segment number and polarity in *Drosophila*. *Nature*, 287:795–801, 1980.
- [18] E. Praun, H. Hoppe, M. Webb, and A. Finkelstein. Real-time hatching. In *Proceedings of the ACM SIGGRAPH*, pages 581–586. ACM, 2001.
- [19] J. Rieffel and J. Pollack. Automated assembly as situated development: Using artificial ontogenies to evolve buildable 3-d objects. In *Proceedings of the GECCO’05*, 2005.
- [20] L. Schramm, Y. Jin, and B. Sendhoff. Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. In *Proceedings of the European Conference on Artificial Life*, 2009.
- [21] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley-Interscience, 1995.
- [22] A. Siddiqi and S. Lucas. A comparison of matrix rewriting versus direct encoding for evolving neural networks. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, 1998.
- [23] K. Stanley. A taxonomy for artificial embryogeny. *Artificial Life*, 9:93–130, 2003.
- [24] 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.*, pages 851–858, 2008.
- [25] T. Steiner, Y. Jin, and B. Sendhoff. Vector field embryogeny. *PLoS ONE*, 4(12), 2009.
- [26] T. Steiner, L. Schramm, Y. Jin, and B. Sendhoff. Emergence of feedback in artificial gene regulatory networks. In *Proceedings of the CEC 2007: Congress on Evolutionary Computation.*, pages 867–874, 2007.
- [27] T. Steiner, J. Trommler, M. Brenn, Y. Jin, and B. Sendhoff. Global shape with morphogen gradients and motile polarized cells. In *Proceedings of the 2009 Congress on Evolutionary Computation*, pages 2225–2232, 2009.
- [28] G. Tufte. Gene regulation mechanisms introduced in the evaluation criteria for a hardware cellular development system. In *Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems*, pages 137–144, 2006.
- [29] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics*, 25(4):1294–1326, 2006.