Global Shape with Morphogen Gradients and Motile Polarized Cells

Till Steiner, Jens Trommler, Martin Brenn, Yaochu Jin, Bernhard Sendhoff

2009

Preprint:

This is an accepted article published in Proceedings of the 2009 IEEE Congress on Evolutionary Computation, Trondheim. The final authenticated version is available online at: https://doi.org/[DOI not available]

Global Shape with Morphogen Gradients and Motile Polarized Cells

Till Steiner, Jens Trommler, Martin Brenn, Yaochu Jin, Bernhard Sendhoff

***Abstract*— A new cellular model for evolving stable, lightweight structures is presented in this paper. The focus lies in enhancing the ability of the cellular system to create complex 3D shapes with non self-similar regions. Compared to our previous work [17], the model proposed in this paper is composed of polarized cells that have directionally differential force functions for cell adhesion and thus are able to follow morphogen gradients (chemotaxis). We investigate the evolution of global information in form of evolving morphogen gradients that are created prior to development, which serve to guide cellular and shape differentiation.**

Our analysis shows that for a set of Pareto-optimal solutions of lightweight stable structures, no unique gradient can be evolved. Nevertheless, it is revealed that neighboring individuals in the genotype space are also neighbored in the gradient space. By contrast, neighborhood in the fitness space is not maintained in the genotype space. These results suggest that a hierarchical genetic formulation might be better than a 'common predefined spatial pattern' in form of a predefined gradient. In addition, our analysis also implies that some well-known properties in direct-coding evolutionary algorithms may be lost in developmental mappings.

I. Introduction

Biological multicellular development creates an organism starting from one cell, the fertilized egg, also referred to as the zygote. During the various phases of development, cells change their fate and a coordinated growth takes place which results in the formation of tissues and organs which eventually make up the complete organism. Cellular specialization and the coordination of development is based on cell-cell signaling and chemical gradients. One important developmental step, which is virtually always the initial stage of growth, is body axis determination. This stage yields polarity of the embryo [8] represented by chemical gradients, which enable differential gene expression at different positions inside the growing organism. These so called morphogen gradients or maternal effects are usually established prior to organism growth and kept intact for a specified period of developmental time.

Gradients have a great influence on development. Both the developmental process and the morphogen gradient are under the genetic control of the DNA belonging to the embryo or the mother individual. Therefore, evolution must have found a way to coordinate both, the developmental process and the gradient such that an optimal growth of the organism can take place.

Till Steiner, Yaochu Jin, and Bernhard Sendhoff are with the Honda Research Institute Europe, 63073 Offenbach, Germany. Email: {till.steiner; yaochu.jin; bernhard.sendhoff}@honda-ri.de.

Jens Trommler and Martin Brenn are with the Control Theory and Robotics Lab, Technische Universität Darmstadt, Karolinenplatz 5, 64289 Darmstadt, Germany.

In artificial development, different aspects of natural development for evolutionary optimization purposes are simulated. A thorough overview is presented in [10]. Predefined signals in terms of morphogen gradients have been used in many evolutionary development approaches as guiding signals (see e.g. [3], [6]). However, these signals were to our knowledge always manually chosen and inserted as a starting condition for the development of every individual throughout evolution. Thus, the question remains, how a good choice for this predefined signal can be determined. Of course, this depends on the given problem, e.g. if the solution is known to be symmetric, a symmetric gradient would clearly be beneficial. Unfortunately, such prior knowledge is usually not given. Also, it is not clear, whether there is a unique best signal for a given problem, especially if two or more conflicting objectives are present.

These questions are the basis for the research work presented in this paper. In the following, we want to investigate the coevolution of predefined signals and the DNA as the blueprint for growth. The objective of the evolutionary run is the design of lightweight, stable structures, i.e. a multiobjective optimization problem in 3D design space. 3D development has become more popular in the artificial development community recently, see e.g., [1], [4], [6], [9], [11], [12], [14], [17], [19].

Our paper is structured as follows: Firstly, we will give a short introduction to predefined signals or morphogen gradients in biology, and their role in development. This motivates the simulation of predefined gradients in artificial developmental systems. Then, we will describe our model and the evolutionary scheme we use to investigate predefined signals in simulated evolutionary development. A separate section is devoted to the way we measure the distance between two genotypes, since this is a nontrivial task in our case. Experiments and results are presented, and based on the findings given in the analysis section, we close with a conclusion.

II. Globally predefined signals in biological development

A coarse 'global' patterning precedes development of multicellular organisms in biology [8]. This patterning is defined by several chemicals, which spatially differ in concentration in the embryo, and thus form gradients. Such gradients are known to specify the main body axes in early development, (i.e. head-tail and front-back) and serve for further segmentation of the embryo.

Gradients can arise autonomously, triggered by an event such as the sperm entry into the egg cell during fertilization (which e.g. is found in sea urchin development [8]), or be

created by cells of the body of the mother individual (e.g. nurse cells, which surround the egg cell in Drosophila fruit flies). Such gradients are therefore termed maternal effects.

The simulation of these maternal effects in artificial development is mainly inspired by the knowledge gained from Drosophila development: Cytoplasmic polarity yields a head-tail orientation and facilitates further patterning of the embryo. Subsequent subdivisions of the initial chemically distinguishable head, center, and tail regions result in segments, each defined by an increased concentration of one or several specific proteins, regulating differential gene expression in that area.

Patterning gradients can be observed not only in the fruit fly, but also in virtually all other species. Several reasons exist for simulating these gradients: Firstly, they can provide cells with global positioning information. Since cells usually only react to local information, this facilitates global patterning of the embryo. Secondly, gradients can yield directional information. Cells can move relative to the concentration gradient of chemicals, e.g. following the gradient to the position of a local maximum. This process is termed chemotaxis, and can lead to cell sorting and organ separation. Thirdly, the chemicals trigger the initial stage of development, and therefore determine its starting point, i.e. they have temporal influence on the process.

As briefly mentioned above, we see the hierarchical formation of gradients in biology, usually starting with few monotonic gradients along one future body axis (e.g. head-tail), and subsequent subdivisions of this gradient into segments from coarse to fine. For our simulations, we choose to define the 'coarseness' of the gradient information beforehand and keep the same level throughout development, although it would certainly be worthwhile to simulate a gradual fine-graining while development progresses.

III. The model

In this section, we will describe the simulation model. It is an extension of a model we described earlier [18] and contains two new features in the cellular representation: polarization and chemotaxis. The section gives details on the cellular model, the genetic model and the evolutionary model. We also explain the mechanism used to simulate evolution of chemical gradients and give the fitness function used during evolutionary runs.

A. The cellular model

The cellular model presented here is derived from the sphere cell model presented in [18]. In that model, a cell is represented by a sphere with a constant radius and an arbitrary position in space. Cells interact physically by nonlinear forces, i.e. depending on their distance, they adhere to each other, and if they overlap, they repel each other. Adhesion between cells can be differential, which may lead to cell sorting.

The resulting outer surface of the shape of a cell aggregation always resembles a solid block, since this is the minimal energy state for a heap of cells that adhere to each other, independent of the strength of the adhesion. To enable the model to create complex, globally differentiated shapes, we add polarization to the cells.

B. Polarized cells

The inspiration for this extension to our model partly comes from the way complex shapes are created by molecules due to differentially charged regions, and partly from observing cellular adhesion in biology, where cells form sheets rather than heaps.

We implement polarized cells as spherical cells with an arbitrary number of charges attached to an arbitrary position on their surface. During our simulations, for computational reasons, we restrict the number of charges per cell to 2, attached to the equator of the cell. The magnitude of the charges can be any real valued number between 0 and 1, while one of the two charges is always treated as positive charge and the other, multiplied by -1, as negative charge. Charges with equal sign repel, charges with different sign adhere, according to the force function

$$F = c_1 \cdot c_2 \cdot 4e^{\frac{-d^2}{\sigma^2}}. \quad (1)$$

Here, c_1 and c_2 are the respective charges, d is the distance between these charges and σ is a constant calculated such that the adhesion force F for maximum charge difference (i.e. both charges have magnitude one and opposite sign) and maximum distance (d =FR) equals to 0.01, and at smallest distance equals to 4. FR is a value which is genetically determined, and can take values between $2r$ and $10r$, where r is the cell radius. FR represents the region in which neighboring cells exert forces on each other.

Genetically encoded values are: the magnitude of the charges, the positions of the charges on the equator of the cell, and FR for every cell, respectively.

C. Chemotaxis

For the simulation of chemotaxis, we determine a direction and a magnitude of a force that is applied to cell centers, in addition to the force arising from cell-cell interaction. The direction of this force is calculated by the local gradient at the position of the cell. Since chemicals are distributed on a fixed, equally spaced grid, and cells can have a position independent of that grid, we calculate a local approximation of the gradient in the following way: First, we take the 8 nodes of the diffusion grid that are closest to the cell. They represent a cube in which the cell lies. Then we calculate a first order approximation of the derivative on all edges of that cube, which gives us four derivatives for each direction. Then, we linearly interpolate the respective approximations for x-, y- and z-directions to the actual position of the cell, such that we receive an approximated gradient.

The magnitude of the force is genetically encoded and is between 0 and 0.5, so that the maximum chemotaxis force is much smaller (factor 8) than the maximum possible cell adhesion/repulsion force due to polarization.

D. The GRN model

We use the same GRN model as described in [18]. Therefore, we will only briefly outline it here. During development of an individual, each cell contains the same genetic information, a copy of the virtual DNA (vDNA) of the organism. The vDNA is a vector of genetic units, which can have one of the following two types: regulatory units (RUs) and structural units (SUs). A gene is a collection of a number of RUs and SUs adjacent to each other. Gene boundaries are given by the vDNA-positions that have a SU to their left and a RU to their right (see Figure 1).

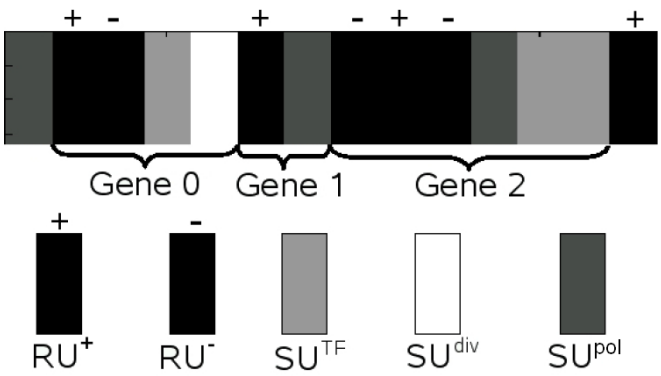

Fig. 1. An illustrative vDNA with three genes, each consisting of one or more structural subunits (SUs) and regulatory subunits (RUs). Two different kinds of RUs exist: inhibitor (RU^-) and activator (RU^+). A SU coding for the production of a transcription factor (TF) is denoted by SU^{TF}, a SU coding for a division by SU^{div} and a polarity modification SU by SU^{pol}.

RUs and SUs are themselves vectors of double precision values. The first value of a SU encodes the respective SU-type. Possible types are: cell division, production of an inter/intra-cellular signal (so called transcription factor (TF)), change of chemotactic behavior, or change of cell polarization. Depending on the SU-type, the remaining values of the SU code for specific features, such as division angles for division in 3D space, magnitudes of charges, or diffusion and decay constants for diffusing signals, etc.

TFs are simulated chemicals that are either produced by the cells or already present in the environment. Their distribution is allocated on a 3D grid. If a cell produces a TF, it retrieves a genetically encoded 'signal-type', a diffusion rate, a decay rate and a release magnitude from the respective SU. Also, a factor in the SU encodes the share of TF which is kept inside the cell and the share that is released into the environment. These chemical signals, their concentration and type at different positions and in different cells, regulate the expression of genes in the cells. Gene activity for every gene in every cell is calculated using the RUs of the gene, with the mechanism outlined below. Note that depending on the activity level of a gene coding for TF production, the amount of released TF is varied. This creates regulatory networks with intra- and inter-cellular signaling.

A RU can read chemical signals, either from the environment at the position of the respective cell, or from within the cell. Depending on a threshold for the concentration of the TF, the 'type-value' of the TF, and an associated 'type-value' inside the RU, an affinity between the TF and the RU can be calculated. The larger the affinity, the stronger the activation power of the respective TF on the RU. An overall activity of a gene is calculated taking all RUs of the gene and all TFs in the environment into account. This activity is then used to regulate the expression of all SUs inside the gene.

E. The evolutionary model

Similar to our previous work [17], we use a modified real valued NSGA-II algorithm [5] with two objectives. We do not use crossover and mutate individuals in an Evolution Strategy-fashion with strategy parameter adaptation [15]. Also, we add gene duplication and transposition: with a probability of 0.1, a random number of consecutive RUs and SUs are either cut out (transposition) or copied (duplication) and then inserted into the vDNA at a new position. Population size for all experiments is 100 and evolutions are run for 300 generations.

F. The gradient evolution scheme

We want to investigate the evolution of pre-defined chemical gradients, in combination with the evolution of the vDNA and the growth process. To achieve this, we define several positions inside the cube shaped calculation area, on which the center of different Gauss-shaped gradients can be allocated. These positions are the center of the cube, the centers of all faces and the corners of the cube, totaling a number of 15.

Each Gauss gradient has a constant variance of $\sigma^2 = 0.5 \cdot s$ with s being the side length of the calculation area. Gaussians are scaled by $\sigma\sqrt{2\pi}$ to reach a maximum possible concentration of 1 at their center points. Additionally, the i-th gradient is scaled by a respective value h_i encoded in an additional chromosome of length 15 in each individual. Finally, the gradients are superposed and inserted into the calculation area as one predefined TF (see the first panel in Figure 7 for an illustrative example of a resulting distribution). Equation (2) gives the concentration $c(\vec{x})$ at every point $\vec{x}$ in the calculation area:

$$c(\vec{x}) = \sum_i h_i \cdot e^{\left(-\frac{(\vec{x}-\vec{\mu}_i)^2}{2\sigma^2}\right)}, \qquad (2)$$

where μ_i is the center of the i-th Gaussian.

The predefined TF is used for chemotaxis, and can also be read by the cells for gene activation. Other TFs that may be released by cells are not used for chemotaxis, and only serve for intercellular communication and gene activity calculation.

G. Fitness function

Fitness calculation is based on two objectives: the minimization of the overall weight of the grown structure, and the maximization of the stability against a load which is applied to the top of the structure after development. When cellular development has finished, the resulting cellular design is converted into a voxel-structure and a top plate is added (see lower right panel in Figure 5). The overall weight of

the structure is then calculated by summation of the number of voxels that build up this resulting design. To calculate stability, the displacements resulting from application of a constantly distributed force to each voxel of the top plate are calculated using the Z88 FE-solver[1]. The maximal displacement describes the 'weakest' point of the design and thus, minimization of this displacement represents the second objective. There are several methods designed specifically to solve this kind of problem. Two such methods are the Solid Isotropic Material with Penalization method (SIMP) and the Evolutionary Structural Optimization method (ESO) [2].

IV. Gene-Neighborhood

To compare different individuals on the genotype level in a generation and throughout evolution, it is necessary to be able to measure how "close" one genotype is to an arbitrary other one. This is, in some cases, straight forward: e.g., in the case of an evolution using a direct genotype-phenotype-mapping with constant chromosome sizes, one can calculate the Euclidean distance between the chromosomes.

Problems arise when individuals have complex representations, especially for individuals whose genomes have different lengths. Then, no clear assignment of elements in one chromosome to elements in the other chromosome is possible. Further complications arise when duplication and transposition are possible mutation mechanisms, because they can be the reason for some major realignments of the chromosome.

We decided that for our case, it is possible to use a measure based on a sequence alignment algorithm known from bioinformatics. We use a modified Needleman-Wunsch-Algorithm [13] and a dynamic programming approach for its calculation. The algorithm tries to find a way to transform one chromosome into an arbitrarily sized other chromosome. At the same time, it measures the "effort" that is necessary to do so, in terms of mutation-, copy- and transposition-operations. This effort can then be taken as a measure for how "close" the one chromosome is to the other.

We will now outline the algorithm by means of an illustrative example. Imagine we want to transform the chromosome ABC into the chromosome ABABC. First, we need to define all distances between single elements. The element distance d we use in this example is for simplicity based on the position of the characters in the alphabet, e.g. d(A,B)=1, d(A,C)=2. Second, we create an alignment matrix R in Figure 2(b), where every column is labeled with one element of the first chromosome, and every row is labeled according to the element in the target chromosome. A first row, initialized with zeros is added. The entries in this matrix, once computed, will allow us to find the best alignment. The elements of the matrix are calculated row by row in the following way: For each element in row i, column j, perform the following minimization, looking at row $i-1$:

$$\min_k(d_{i,j} + r_{i-1,k} + V_k), \tag{3}$$

[1]http://www.z88.org

where $k \in [1..N]$, and N is the size of the first chromosome. $r_{i-1,k}$ is the entry in row $i-1$ and column k of matrix R, and V_k represents a gap penalty, being $V_k = 0$ for $k = j-1$ and $V_k = p$ else. The choice of p is done manually, and will be discussed later. In this example, we set it to 0.5. V_k and d are depicted exemplary as small numbers in Figure 2(a).

(a)

R	A	B	C
	0	0	0
A	0	1	2
B	1,5	0	

+0 +0,0 +0 +0,5 +0 +0,5

(b)

R	A	B	C
	0	0	0
A	0	1	2
B	1,5	0	1,5
A	0,5	1,5	2,5
B	2,0	0,5	2,0
C	3,0	2,0	0,5

Fig. 2. (a) An illustrative example for the computation of an element of the back-trace matrix: Elements that make up the new entry are the minimum of the values of the previous row plus the distance between the actual row-element and the actual column element, plus the gap penalty where it applies. The minimum for the depicted case is emphasized by a bold arrow. (b) The resulting back-trace matrix for the example. The circle denotes the starting point for back-tracing, and the arrows give the alignment of the two vectors (ABC) and (ABABC).

We insert the found minimum into the respective position (i, j) in R and create a pointer from that position to the cell $(i-1, k)$ where the value was taken from (denoted by a bold arrow in Figure 2(a)). Doing the same for the whole row, we end up with a value and a pointer in each cell.

This scheme is continued until the whole alignment matrix is filled. The best alignment can be found by starting in the cell containing the lowest value in the last row (denoted by the circle in Figure 2(b)) and following its pointer to the penultimate row and from there onwards until the first row is reached (indicated by arrows in the same Figure). The visited positions in the matrix give the alignment. We calculate the effort E for the alignment of two chromosomes Chr_1 and Chr_2 by

$$E(Chr_1, Chr_2) = \frac{\sum d*}{M} + \sum V_k*, \tag{4}$$

where M is the size of the target chromosome, $d*$ are the element distances d on the way and V_k* are the gap penalties on the way. E taken from the alignment of chromosome 1 ($Chr1$) with chromosome 2 ($Chr2$) may differ from E taken from the alignment of $Chr2$ with $Chr1$. We therefore define the distance D between the two chromosomes as $D(Chr_1, Chr_2) = E(Chr_1, Chr_2) + E(Chr_2, Chr_1)$.

The characters ABC in the example represent gene parts, i.e. RUs and SUs. The distance d between two RUs or two SUs is given by $\frac{||v_1 - v_2||_2}{N}$ where v_1 and v_2 are the element vectors respectively, and N is the length of the vectors, $||\cdot||_2$ denotes the 2-norm. The distance between a RU and a SU is set to 10000, such that this alignment will never be chosen.

Note that the resulting alignment depends on the penalty value p. If the value is very low, the algorithm tries to match

elements which are close in element-wise distance d and uses many transposition steps for this. If the penalty value is high, the algorithm matches values with an eventually high element-wise distance to receive long local alignments which avoids duplications or transpositions.

The choice of the gap penalty is in itself a multi-objective optimization problem, because depending on the choice of this parameter, an alignment with more or less transpositions will be created. Since there is no "one correct alignment" we need to choose one gap penalty and keep it as reference for the analysis. The way we choose this penalty will be described in the analysis section.

V. Experiments and Results

Our simulation comprises two different experiments: The first is an evolutionary run with a predefined, manually chosen gradient. The spatial function of the concentration c is given in Equation (5). Note that the calculation area extends from -8 to 8 in each spatial direction.

$$c(\vec{x}) = 1 - e^{\left(-\frac{(\vec{x}-\vec{\mu})^2}{2\sigma^2}\right)}, \quad \mu = (0, 0, -8). \tag{5}$$

σ^2 is chosen similar to Equation (2). The choice of this gradient is inspired by the shapes that typically result from the SIMP-Algorithm, where 'table-legs' are created at the corners and centers of side-boundaries, Figure 3(a). Figure 3(b) gives a typical solution from the ESO algorithm. Using the gradient in Equation (5), even without evolutionary optimization we can generate a table-like structure 3(d). A simple chemotaxis along the manually defined gradient and continued cell division seems to be sufficient to create the four-legged table. For comparison, we also depict a typical solution from the previous model on which this work is based [17] in Figure 3(c). There, only dark cells are used as structure, while white cells represent spacers and are removed prior to fitness calculation. By comparison, Figure 3(e) and (f) show an evolved design using this manually chosen gradient, and the voxelization of the resulting design.

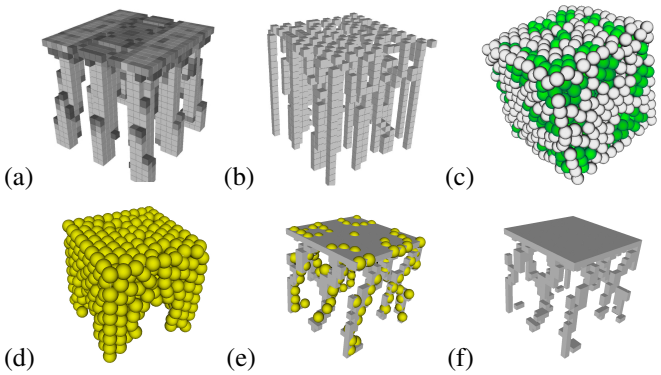

Fig. 3. (a) A typical solution using the SIMP algorithm. (b) A typical solution using the ESO algorithm. (c) A typical solution created by the old model, presented in [17], where white cells are spacer cells and removed for FE-calculation. (d) An un-evolved, hand-coded genome yields this solution with motile polarized cells and a predefined gradient. (e) An evolved solution with motile polarized cells using a predefined gradient, and (f) the resulting voxel grid.

The second experiment uses a superposition of Gaussian distributions as described in section III-F. For all developments, 4 cells are initially placed at the four upper corners of the calculation area. Evolution starts with a random genome of 5 SUs and 5 RUs, and the evolution runs for 300 generations. The gradient is created in the calculation domain before the development of the individual starts, and does not change during the developmental time.

Figure 4 shows the Pareto-optimal solutions at generation 500 achieved by five different models, where "old model" means the results obtained by a simplified cellular model used in [17]. It is clearly visible that the new model has a greater ability to reach better results on the Pareto-front, especially solutions with less than 1300 voxels are reachable. Also, a wider variety of designs emerge, which can be seen in Figure 6, giving sample phenotypes.

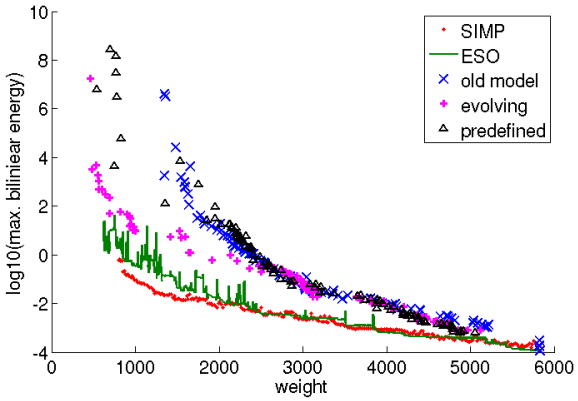

Fig. 4. The populations of typical evolutionary runs after 300 generations. Depicted are individuals of the old model, the model using a fixed, predefined gradient, the model using evolving gradients, and for reference solutions from the SIMP- and ESO-algorithms. Note: the whole generation is plotted, not only individuals on the Pareto-front. Also note that for computational reasons, the cell number of the polarized sphere-cell model was restricted to 500 cells, such that a maximum of 5100 voxels will not be exceeded.

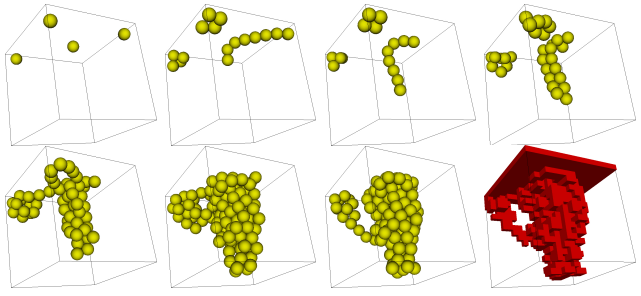

Fig. 5. Excerpts from a sample growth of an individual and the resulting voxel grid. Note the added top plate. Sequence starts at upper left and ends at lower right.

Comparing individuals with evolved and predefined gradient, Figure 6(a) and 6(b), we can see that in the latter case, phenotypes are all built up using a common principle: Cells use the clue given by the predefined gradient and move towards the outer area of the cube. The center remains

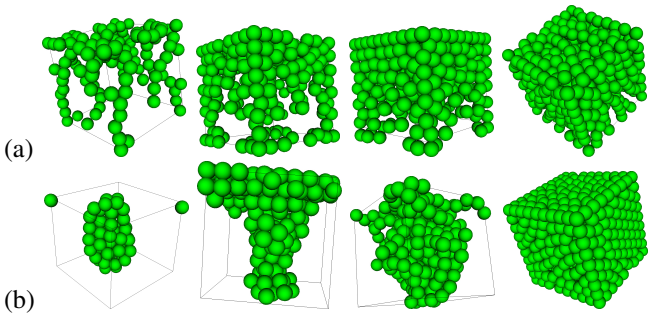

Fig. 6. (a) Individuals taken from generation 300 from a fixed, predefined gradient evolution run. Number of voxels for the individuals (from left to right): 755, 1361, 2027, 2866. (b) Individuals taken from generation 300 of an evolving gradient run. Number of voxels (from left to right): 758, 1420, 1665, 2331.

sparsely occupied by cells. Designs only differ by the number of cellular divisions and resulting 'compactness' of the design.

The evolving gradient simulations show a wider range of different designs and growth processes. Both chemotaxis and polarized adhesion can be observed during the growth process of several individuals. Heavy solutions are created by similar design processes: Cells possess equal charges, such that they repel each other. Due to the restricted calculation area, a sponge-like distribution occurs. Most light solutions are built up from a central stem, which branches towards the top of the calculation area. Figure 5 gives snapshots of a growth sequence for such a design.

The quality of designs using evolving gradients, especially for design weights smaller than 2500 voxels, is clearly closer to the reference curves created by the SIMP and ESO algorithms as compared to the old model. Figure 7 gives two more interesting phenotypes from the evolving gradients run.

To understand the role of predefined gradients for evolution, we performed several analysis of the evolutionary run with evolving gradients, which we present in the following section.

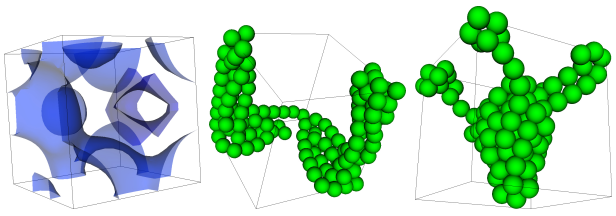

Fig. 7. Equipotential surfaces of a sample gradient from the gradient evolution run is depicted. Also, two interesting designs that emerged during evolution in the evolving gradient run.

VI. Analysis

We look at the evolved TF-gradients along the Pareto-front. At first glance, gradients of the individuals look quite different from each other. What is of interest is, (a) is there a common 'best' gradient evolving, and (b) whether and how are genetic neighborhood and gradient similarity related? Also, we want to investigate whether individuals neighboring on the Pareto-front possess more similar gradient information compared to individuals with a larger distance on the Pareto-front.

Figure 8 shows the standard deviations for each of the 15 h-values (compare Equation 2) of the individuals in the respective generation. Clearly, no common distribution is evolving, rather a large spread can be seen. Therefore, we direct our attention towards the neighborhoods, both in the genotype space as well as on the Pareto-front.

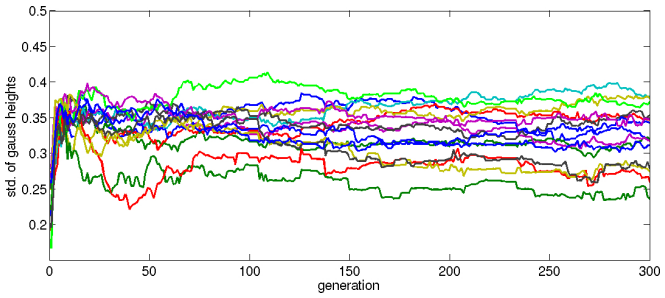

Fig. 8. Each line represents the evolution of the standard deviation in each generation of h_i, the i-th Gaussian height encoded in the genome of the individuals.

Figure 9(a) gives first information about the neighborhood for the population of generation 300: we distribute solutions on the circle according to their total number of voxels, i.e. their weight. Note that this sorting gives a good representation of neighborhood in fitness-space since the Pareto-curve in the fitness-plot (Figure 4) is almost monotonic. Thus, the plots in Figure 9 would look similar using the the maximum bilinear energy instead of the weight. We calculate the Euclidean distance between the h-values in the respective chromosomes. This gives us a measure for the difference in predefined Gaussians for the individuals. For each individual, we look for the nearest neighbor, i.e. the individual with the smallest distance in h-values. We then draw a line between these points to indicate the closest neighbor, and scale line width by the inverse distance. Note that if individual A has individual B as its closest neighbor, this does not necessarily indicate that the closest neighbor of individual B is individual A. Therefore, some individuals have more than one line attached to them. Minimal and maximal values of these minimal distances are given in Figure 10. If a local neighborhood rule would apply here, only points are neighbors on the circle should be connected. This is not the case. The only region where this can partly be observed is the region around 3000 voxels, where individuals are highly clustered in fitness-space. Interestingly, feasible solutions in the first generation originated from this area. We also investigated the same plot throughout evolution and did not find a local neighborhood rule to be applicable in any generation. For comparison, we now look at the same plot for the chromosomes controlling growth, using the previously described gene neighborhood. The first consideration lies in choosing a suitable gap penalty p. We performed our analysis for several gap penalties between 0.0 and 1.1, representing

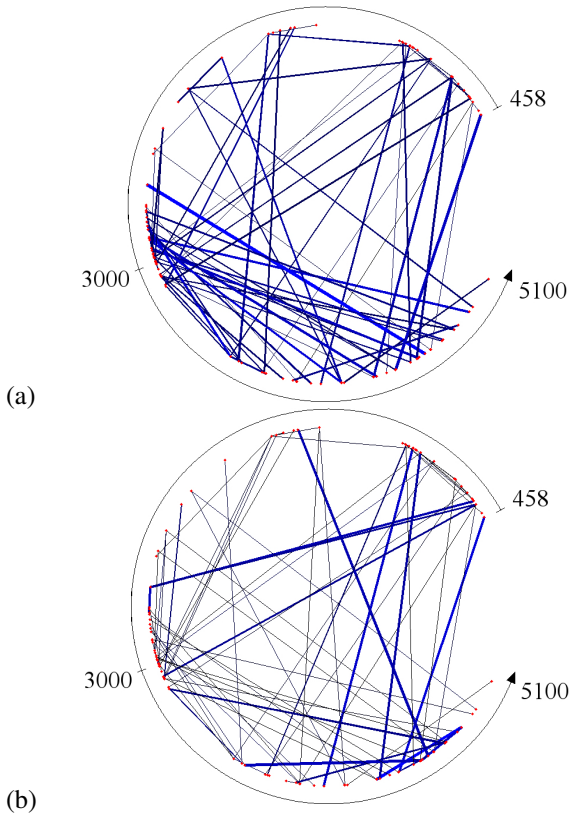

Fig. 9. (a) All individuals from the 300th generation are depicted, where individuals are sorted on the circle by their weight, i.e. by the first objective. Nearest neighbors in terms of evolving gradient are connected with a line. The line width is scaled with the inverse distance of the nearest neighbors. The same plot is depicted in (b) for nearest neighbors in terms of genome-distance.

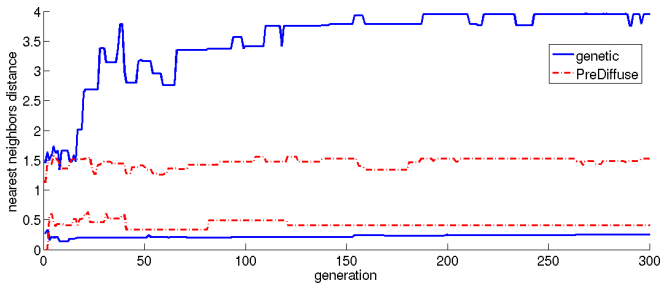

Fig. 10. The minimal and maximal values for nearest neighbors distance for both, the pre-diffused gradient neighborhood, and the genome neighborhood.

the extreme cases of no gap penalty and a gap penalty that is higher than the maximal element wise distance. Interestingly, we found qualitatively similar results for penalty values between 0.1 and 0.4. We also found that the overall distances seem to get smallest for $p = 0.2$ and thus we chose this gap penalty for the investigation.

The resulting neighborhood plot is given in Figure 9(b) for generation 300. On closer inspection, it resembles the neighborhood distribution given in Fig. 9(a) for the evolving gradients, if line widths are not taken into account. This can be done because we are primarily interested in neighborhood relations and not the actual distance between neighboring individuals. To investigate the similarity, we depict the percentage of individuals, in which the nearest genetic neighbor also is the nearest neighbor in terms of evolving gradient in Figure 11 (neighborhood 1). Neighborhood 2 denotes the case where the two nearest neighbors in terms of genes and gradients are taken into account for finding an overlap. For this increased neighborhood, we see that 80 percent of the neighborhoods match, with a slightly decreasing trend as evolution progresses. It seems that the gradients evolve with the genes and vice versa. Thus, 'growth programs' encoded in the genes seems to be fine-tuned to their predefined gradient (and vice versa).

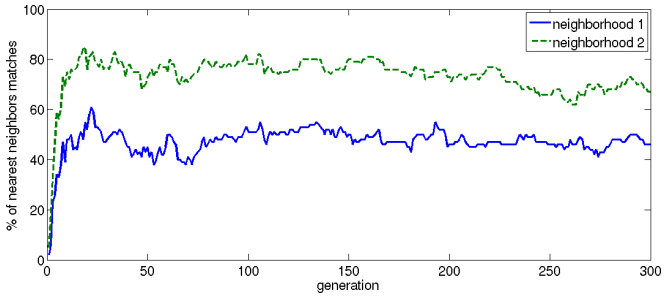

Fig. 11. The percentage of nearest neighbors match between gradient neighborhood and genome neighborhood throughout evolution. The two curves give the result for a neighborhood size of 1 and 2 respectively.

Considering Figure 9(b), we also see that genetic neighbors are not necessarily close in fitness space. We also monitor this fact throughout evolution and present it in Figure 12. Each point in this plot gives the difference in phenotype weight, i.e. the difference in one of the fitness criteria between an individual and its nearest genetic neighbor. Note that due to the monotonic curve of the Pareto-front, we would obtain the same results for the second fitness criterion. It is clearly visible that throughout evolution the distribution of these points does not vary qualitatively. Thus, the evolutionary algorithm has to deal with a complex genotype-phenotype map in every generation, which may prevent it from evolving towards a more 'evolvable' map.

VII. Conclusion

We found that the cellular model using chemotaxis and polarization yields the ability to evolve complex shapes, with globally differentiated parts. Evolution reaches a set of solutions that is better than the simplified model presented in [17] for the given task. Using evolving predefined gradients, phenotypes seem to follow different developmental strategies, although a central development pattern seems to be chosen by most lighter variations.

Evolution results in many genetically different solutions for the problem and a distribution in fitness space. The pre-diffused gradients seem to align with their respective genetic program, such that genetic and gradient neighborhoods overlap, although this effect decreases slightly throughout evolution. Overall, we can say that no 'one best' gradient for the given evolutionary task exists. Rather, it seems that

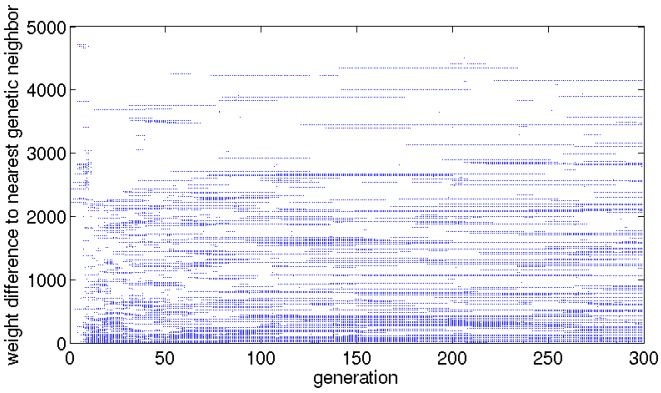

Fig. 12. Each dot represents the design weight difference from one individual to its nearest genetic neighbor. This plot shows that throughout evolution, neighbors in fitness space do not come closer together on the genetic level and vice versa.

individuals use the evolving gradient as a coarse-grained prestructuring part of their own developmental program: The gradient defines regions and coarse cell movements in design space, and tightly coevolves with the developmental programs of the cells which then use this information during growth.

These findings imply that it is not advisable to simply pick a 'best' gradient for a given problem. A manual gradient choice could reduce evolutionary diversity.

Therefore, we suggest that a hierarchical genetic formulation should be used for developmental systems, rather than a complex, problem-specific predefined gradient for all individuals. In the hierarchical formulation, genomes could be separated into parts dedicated to a resolution level, such that the first part creates a coarse structuring of the individual, and the following parts do a subsequent fine-graining of the respective previous level. This seems reasonable, since complexification has been shown to be useful for developmental systems ([7], [16]), although realized in a different way.

Neighborhood on the Pareto curve does not equal neighborhood in genotype space. This fact does not change significantly at any time in evolution. Therefore, an evolutionary method like the NSGA-II, which tries to fill gaps in the Pareto-front by favoring individuals that have a low number of close neighbors, is not adequate.

ACKNOWLEDGMENT

We would like to thank Markus Olhofer, Lars Graening and Lisa Schramm for valuable discussions.

REFERENCES

[1] D. Basanta. *Using genetic algorithms to evolve microstructures*. PhD thesis, University of London, 2005.

[2] M.P. Bendsoe and O. Sigmund. *Topology Optimization - Theory, Methods and Applications.* Springer, 2003.

[3] C.P. Bowers. *Simulating Evolution with a Computational Model of Embryogeny (PhD Thesis).* The University of Birmingham, 2006.

[4] S. Cussat-Blanc, Luga H., and Y. Duthen. From single cell to simple creature morphology and metabolism. In *Artificial Life XI*, pages 134–141, 2008.

[5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms.* Wiley, 2001.

[6] P. Eggenberger. Evolving morphologies of simulated 3d organisms based on differential gene expression. In *Proceedings of the 4th European Conference on Artificial Life*, 1997.

[7] D. Federici. Increasing evolvability for developmental programs. In J. Miller, editor, *GECCO Workshop on Regeneration and Learning in Developmental Systems*, 2004.

[8] S. Gilbert, editor. *Developmental Biology.* Sinauer Associates Inc., 2003.

[9] P. Haddow and J. Hoye. Achieving a simple developmental model for 3d shapes: Are chemicals necessary? In *Proceedings of the GECCO'07*, pages 1013–1020, 2007.

[10] Simon Harding and Wolfgang Banzhaf. Artificial development. In Rolf P. Würtz, editor, *Organic Computing*, pages 201–220. Springer, March 2008.

[11] T. Kowaliw. *A good number of forms fairly beautiful*. PhD thesis, Concordia University, 2007.

[12] S. Kumar. *Investigating computational models of development for the construction of shape and form*. PhD thesis, University College London, 2004.

[13] S.B. Needleman and Wunsch C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[14] D. Roggen, D. aand Federici and D. Floreano. Evolutionary morphogenesis for multi-cellular systems. *Genetic Programming and Evolvable Machines*, 8(1):61–96, 2007.

[15] H.-P. Schwefel. *Evolution and Optimum Search*. John Wiley, 1994.

[16] K.O. Stanley. *Efficient Evolution of Neural Networks through Complexification (PhD Thesis)*. The University of Texas at Austin, 2004.

[17] T. Steiner, Y. Jin, and B. Sendhoff. A cellular model for the evolutionary development of lightweight material with an inner structure. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO)*, pages 851–858, 2008.

[18] T. Steiner, L. Schramm, Y. Jin, and B. Sendhoff. Emergence of feedback in artificial gene regulatory networks. In *Proceedings of the 2007 Congress on Evolutionary Computation (CEC'07)*, pages 867–874, 2007.

[19] O. Yogev and E.K. Antonsson. Growth and development of continuous structures. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1064–1065, New York, NY, USA, 2007. ACM.