Evolving the Ability of Limited Growth and Self-Repair for Artificial Embryos

Felix Streichert, Christian Spieth, Holger Ulmer, and Andreas Zell

Center for Bioinformatics Tübingen (ZBIT), University of Tübingen,
Sand 1, 72074 Tübingen, Germany,
streiche@informatik.uni-tuebingen.de,
http://www-ra.informatik.uni-tuebingen.de/welcome.html

Abstract. In this paper we address the problem of limited growth and the difficulty of self-repair in the field of Artificial Embryology. We implemented a topological simulation of multiple cells which is continuous, structure-oriented, with a dynamically connected network of growing cells and endogenous communication between cells. The cell behavior is simulated based on models of gene regulatory networks like Random Boolean Networks and S-systems. Evolutionary Algorithms are used to evolve and optimize the parameters of the models of gene regulatory networks. We compare the performance of Random Boolean Networks and S-systems when optimized by Evolutionary Algorithms on the problem of limited growth and two implementations of cell death and signaling cell death on the problem of self-repair.

1 INTRODUCTION

The research field of Artificial Embryo logy (AE) is rather new and only few papers have been published on this topic. Most researchers regard AE as a simulation environment for early developmental processes, to test or to develop biologically plausible theories of pattern formation. Others apply the generative grammatical encodings of AE together with Evolutionary Algorithms (EA) to shape and structure optimization problems like growing neural network topologies or Evolutionary Engineering (EE). For both topics the generative grammatical encoding of AE offers the advantage of a better scaling behavior over direct encoding schemes.

The first application for AE is to be used for multi-cellular simulation environments in the field of theoretical biology. One example for such a multi-cellular simulation environment is the *Cell Programming Language* (CPL) by Agarwal [1]. Here the cell behavior can be programmed with a number of high level instructions like move(direction) or differentiateto(tissuetype). Another AE environment was introduced by Fleischer [13] [12], which uses cells with attributes like position, shape, and concentrations of biochemicals. The behavior of a cell is based on programmable differential equations computing the concentration of biochemicals. Another example for cell behavior based on differential equations is the *Cellerator* by Shapiro and Mjolsness [22].

Regarding the two applications of AE as coding scheme for growing neural nets and general shapes in EE, growing neural nets seems to be most popular and there are many publications in this research area [3] [7] [10]. EE on the other hand seems to lack a suitable representation scheme for generals shapes. There are only few successful examples of EE in the literature based on specialized EA representations, for instance, Funes and Pollak [14] were able to evolve Lego®bridges and cranes using a tree based EA representation. The general problem of shape representation in EE was discussed by Schoenauer on a topological optimum design problem [21]. He compares several parametric representations but in his conclusions he indicates that generative grammatical encodings could yield major advantages for EE. Actually Hornby and Pollak describe the advantages of generative grammatical encodings as they occur in AE by comparing a parametric, context-free L-System, generating LOGO style GP code, and a standard non-generative GP to build a model of a desk [16].
In this paper we address two fundamental problems for AE, first the problem of limited growth and second the problem of self-repair for artificial embryos, because without the ability of limited growth AE becomes useless as dynamic simulation environment in theoretical biology and also for EE. We solve these problems not by crafting suitable rules, but by optimizing the cell behavior, i.e. the parameters of the cell behavior, with an EA. We compare the performance and the properties of two models of cell behavior on the problem of limited growth and two different cell models on the problem of self-repair.
The next section gives an overview over some of the related work in the field of AE. In section 3 we describe our implementation of a simulator for a multicellular environment and the EA implementation. Results on the problems of limited growth and self-repair are discussed in section 4. Finally, conclusions on the achieved results and an outlook is given in section 5.

2 RELATED WORK

In one of the earliest work by Hugo de Garis [5] the cell behavior was similar to neighborhood interaction rules of cellular automata and determined if a cell is to remain inactive, to die or to split as long as a maximum number of cell division was not exceeded. By using Genetic Algorithms (GA) he was able to grow simple convex shapes but failed on non-convex shapes. Therefore de Garis extended the cell model with sensors for gradients of biochemicals and added iteration counters [4]. Also the cell behavior was enlarged by 'operons' imitating the dynamics of gene regulatory networks, that allowed rules to be switched on or off. With additional sources of biochemicals placed in the environment he was able to grow simple non-convex shapes.
Frank Dellaert examined a biologically defensible model of development and evolved the shape and behavior of simple agents using a GA [6]. To emulate the behavior of gene regulatory networks he used Random Boolean Networks (RBN) and took the binary states of RBN's to detect cell differentiation. Delleart was able to evolve differentiated agents with sensors and actuators, but to

do so he used asymmetrical division for the very first division to establish the anterior/posterior orientation of the organism and he gave cells adjacent to the horizontal midline of the organism a special input signal. Also only 64 cells were allowed to be generated, then the simulation stopped.
Peter Eggenberger also used the concept of gene regulation to control the behavior of developing cells [9] [11]. The cell behavior was based on structural genes, turned on or off by regulating genes depending on a local concentration of biochemicals. Activated structural genes produced biochemicals, cell adhesion molecules, receptors for biochemicals and caused cell division and death. The activation states of genes were used for cell differentiation. Although Eggenberger used EA to optimize shapes, the results he presented relied on additionally crafted exogenous sources of biochemicals that gave the necessary positional information to the organism.
A topological model was introduced by Duvdevani-Bar and Segel [8]. The behavior of the cells was based on the reaction-diffusion model devised by Gierer and Meinhardt [15]. They crafted examples for animal/vegetal region differentiation and examples for cell migration and neural differentiation in the visual system of Drosophila.
Only Fleischer and Barr discussed the topic of limited growth in [12]. They gave examples how to craft rules that produce limited growth by:

- using a threshold of a biochemical concentration to disable cell division.
- exhausting a limited, not regenerating factor necessary for cell division.

All other papers did not explicitly address the problem of limited growth, neither did they mention, if the organism were able to maintain the cell differentiation over a extended period of time. And although most researches were able to produce more or less fancy shapes, we want to concentrate on the most basic problems of AE first and then will start to evolve shapes and structure with EA.

3 ARTIFICIAL EMBRYOLOGY

To discuss our multi-cellular simulation environment we will distinguish between cell behavior, cell mechanics and cell model that merges both. The cell behavior computes the upcoming cell states given by concentrations of n biochemicals $\boldsymbol{B} \in [0;1]^n$. The cell model translates the cell state into cell actions like cell differentiation, growth (mitosis), cell death, etc.. The cell mechanics simulates the effect of cell actions on the neighborhood topology and spatial distribution of cells in the organism.
Using the classification scheme suggested by Prusinkiewicz [20] our cell model is continuous structure-oriented, synchronous, but time discrete, with a network topology and dynamic neighborhood relations and the communication between cells is based on lineage and endogenous interaction.
To simulate the development of an organism we start with a single cell c_0 with $\boldsymbol{B}_{t=0} = [0]^n$, compare fg. 1 at $t = 0$. Each discrete time step we compute the cell model for each cell c_j of the organism. The cell model determines if cell actions take place and then calls cell behavior and cell mechanics successively.

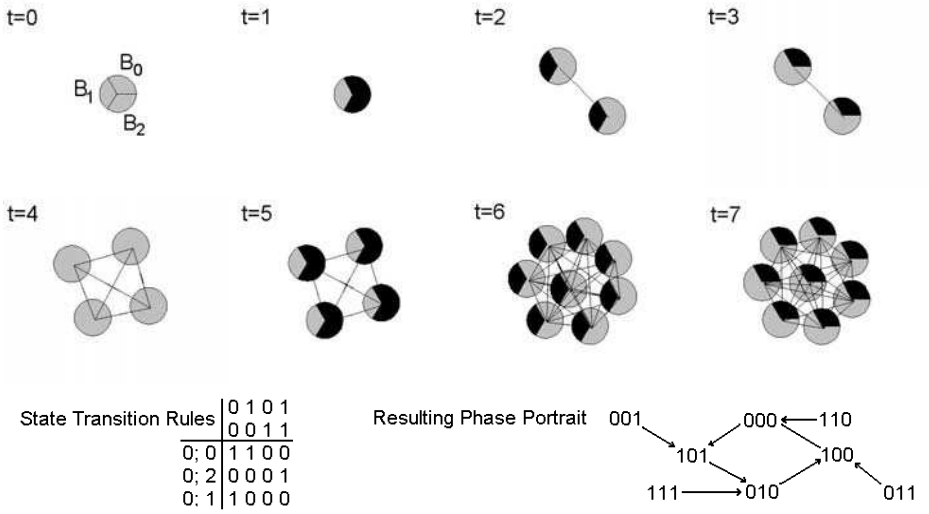

Fig. 1. Simulated organism with RBN ($n = 3; k = 2$), $\boldsymbol{T} = [0.5]^3$, $\boldsymbol{R_{prod}} = [1]^3$, $\boldsymbol{R_{dgr}} = [0]^3$, $\boldsymbol{D} = [0]^3$ and $\boldsymbol{B_{t=0}} = [0]^3$. The cells state equals the state of RBN.

3.1 CELL MODEL

All possible actions performed by cell c_j are based on the current state of the cell's biochemicals $\boldsymbol{B}$ or on external causes, as for example in the case of cell death or diffusion. Currently only few cell actions are possible:
Cell growth (mitosis): In the case of $B_0 = 1$ cell mitosis can happen if enough space is available. To prevent proliferation a cell is only allowed to split, if no neighboring cell is closer than $\frac{t_{dist}}{2}$. During mitosis the daughter cell inherits all properties of c_j, this is called communication through lineage. For both cells the values of $\boldsymbol{B}$ are halved, assuming that both cells instantaneously reach the original volume without generating additional biochemicals. Compare fg. 1 in this example mitosis occurs at $t = 1$, $t = 3$, $t = 5$ etc..
Cell death: Currently death can only be caused by external evens, see chap. 4.2. When dying the cell removes all connections to neighboring cells and is then removed from the environment. In an alternative implementation the dying cell injects an additional biochemical into neighboring cells indicating it's death.
Diffusion of biochemicals: Endogenous interaction between cells is simulated by exchanging biochemicals $\boldsymbol{B}$ between neighboring cells depending on a concentration gradient and the diffusion rate $\boldsymbol{D}$. In fg. 1 diffusion is disabled, $\boldsymbol{D} = [0]^3$.

3.2 CELL BEHAVIOR

The behavior of a cell is to resemble a gene regulatory network of n genes to compute the concentration of biochemicals $\boldsymbol{B_{t+1}} = f(\boldsymbol{B_t})$. There are a number of alternatives to model gene regulatory networks: Random Boolean Networks

(RBNs) [18], Qualitative Network models [2], Weight Matrices [24], Dynamic Bayesian Networks [19], S-Systems [17], general differential equations [15] and many more.
We decided to compare RBNs and S-systems since they are the most different and are both well documented and examined.

Random Boolean Networks (RBN) are one example for a model of gene regulatory networks [18] [25]. They consist of a state vector of n booleans, each representing the state (on/off) of a single gene, in our cell model each one is associated with a biochemical B_i. The state transition rule S_i for each state is defined by n boolean functions with k biochemicals as input I_i. An example for state transition rules and the resulting phase portrait, describing the succession to states, for a RBN is given in fig. 1.
Since this behavior can't be considered realistic we extended the binary RBN to a real-valued RBN. Here the boolean input states for the RBN are calculated from B and each one is set *true* if the associated biochemical B_i exceeds a given threshold T_i. If the subsequent boolean state is *true*, the activated gene produces biochemicals proportional to the production rate $\boldsymbol{R}_{prod}$ or the deactivated gene degrades the biochemical proportional to the degradation rate $\boldsymbol{R}_{dgr}$.

S-systems (*synergistic* and *saturable* systems) have been suggested by Irvine and Savageau [17]. A S-system is given by a set of nonlinear differential equations:

$$\frac{dB_i(t)}{dt} = \alpha_i \prod_{j=1}^{n} B_j(t)^{\mathcal{G}_{i,j}} - \beta_i \prod_{j=1}^{n} B_j(t)^{\mathcal{H}_{i,j}} \tag{1}$$

For each simulation time step $\Delta t = 1$ equation (1) is integrated using a Runge-Kutta algorithm with fixed step size of $t_{step} = 0.02$. To prevent the S-system running into a fixed state with $\boldsymbol{B} = [0]^n$ we demand $\boldsymbol{B}_i \geq 0.0001 \; \forall i$.

3.3 CELL MECHANICS

Our cell mechanics are based on a 'winged vertex' structure which is given by the cell's position (vertex) and the cell's neighborhood relations (wings). The cell mechanics updates both, to achieve equally spaced cells with a smooth neighborhood topology in two dimensional environments we used $r_{cell} = 1$ as cell radius, $t_{dist} = 2 \cdot r_{cell}$ as target distance between cells and $t_{neigh} = 7$ as target number of neighborhood relations.
Regarding the neighborhood relations each cell c_j adds the t_{neigh} most closest cells as neighbors. An alternative method to calculate the neighborhood topology are Voronoi diagrams.
To update the position of c_j, we use the neighborhood topology to simulate the forces of each neighbor exerted on c_j. This is simulated repeatedly to reach a state of equilibrium. To introduce an element of chance, brownian motion is added to the cells position. An example distribution of cells can be seen in fig.

1. A broken symmetry in this example can occur due to limited space available for cell mitosis or due to asymmetrical diffusion of biochemicals because of local differences in neighborhood topology.

Although the example in fig. 1 connotes reproducibility, the brownian motion of cells has a major impact on our results. Compare fig 2 for four examples of the same real-valued RBN based cell behavior. Only in three simulations the organism developed the upper left convexity. This can be explained by taking into account the discrete RBN behavior. Only slight changes in the neighborhood topology can cause variations in the concentration of biochemicals. When using boolean discretization these variations can lead to major changes in cell behavior and finally, the shape of the organism.

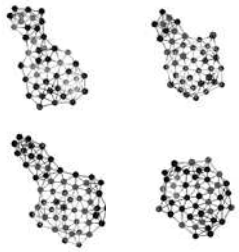

Fig. 2. Example organisms

4 RESULTS

We use an EA-hybrid to optimize the parameters of the RBN and the S-system to evolve suitable cell behavior for organisms with the ability of limited growth and self-repair formulated as maximization problem. The EA-hybrid allows us to mix different EA genotypes and mutation/crossover operators on the level of individuals. We use an $(\mu = 50, \lambda = 100)$ EA population strategy, tournament selection with $t_{group} = 8$ and a mutation and crossover probability of $p_{mut} = p_{cross} = 0.7$ for each selected EA operator. The GA mutation alters one bit per mutation and uses one-point crossover on the GA genotype. On the ES genotype we use global ES mutation and discrete recombination.
To evaluate the fitness each individual (organism) is simulated for $t_{max} = 100$ time steps. For each experiment we perform 10 independent optimizing runs and the behavior of the best found organism of each run is averaged over at least 10 simulations.

4.1 LIMITED GROWTH

We define the EA fitness function for the problem of limited growth as the inverse of minimizing the quadratic error, weighted with t to prefer fast growing organisms:

$$\Phi(x_i) = \frac{1}{1 + \sum_{t=0}^{t_{max}} \left(\frac{t*(\Theta - ||O(x_i)_t||)^2}{t_{max}} \right)} \tag{2}$$

where $\Theta = 30$ gives the desired size of the resulting organism and $||O(x_i)_t||$ is the size of the organism at time t based on the cell behavior suggested by the EA individual x_i.

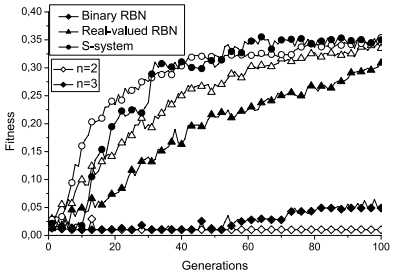

Fig. 3. Avg. of best solutions

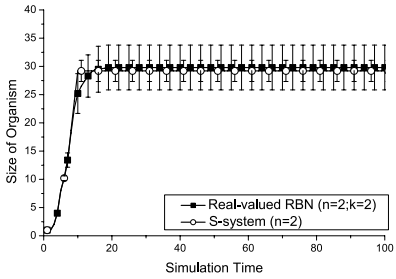

Fig. 4. Avg. size of best samples

We examine the performance of RBNs and S-system with either two or three genes on this problem. First we compare the binary and the real-valued RBN on this problem optimizing the state transition rules for each state with a GA and threshold values $\boldsymbol{T} \in [0;1]^n$, the production rate $\boldsymbol{R}_{prod} \in [0;1]^n$, the degradation rate $\boldsymbol{R}_{dgr} \in [0;1]^n$ and the diffusion rate $\boldsymbol{D} \in [0;0.5]^n$ for each biochemical with an ES if necessary. Fig. 3 shows that the binary RBN fails to evolve the ability of limited growth, but that the real-valued RBN solves this problem even with only two genes available. Regarding the S-system we achieved even better results optimizing $\boldsymbol{\alpha} \in [0;10]^n$, $\boldsymbol{\beta} \in [0;10]^n$, the matrices $\mathcal{G} \in [-3;+3]^{n \times n}$ and $\mathcal{H} \in [-3;+3]^{n \times n}$ and again the diffusion rate $\boldsymbol{D} \in [0;0.5]^n$ and a GA optimized bit-mask for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathcal{G}$ and $\mathcal{H}$ to allow structure skeletalizing [23]. Fig. 4 gives the behavior of sample solutions averaged over ten simulations the corresponding parameters for the examples are given in tab. 1. Both solutions converge to the target size, but in case of the real-valued RBN with a high standard deviation.

4.2 SELF-REPAIR

Next we want the EA to evolve limited growth with the additional ability of self-repair, so that if partially destroyed or wounded, the organism regrows to the former size. To prevent the EA to exploit loopholes, we use two test cases to evaluate the fitness. First the organism is tested whether it is able to grow to a limited size and remain stable, otherwise continuous or time dependent growth could feign the ability of self-repair. Then an additional simulation run is done as a second test case where the organism is wounded at $t = 60$ by killing a group of cells. For both cases the fitness (equ. 2) is used but with two sums under the fraction bar, one for each test case. If the organism fails on the first test case, the second test case is omitted and a penalty is added to the fitness.

Only RBNs with $(n = 3; k = 2)$ are used on this problem and we compare the standard cell model with the alternative version (compare sec. 3.3). It is shown in fig. 5 that organisms with the additional indicator for wounds perform better than the organisms without. Fig. 6 shows the averaged size of organisms on the first test case without wounding. Here the organisms without the additional

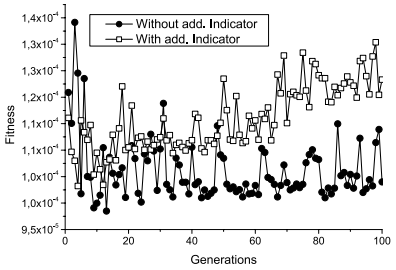

Fig. 5. Averaged best fitness

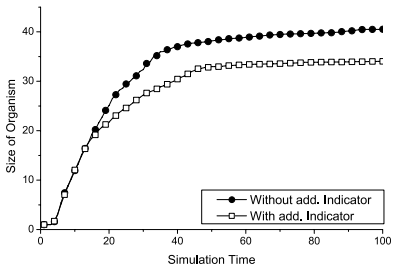

Fig. 6. Av. size on test case 1

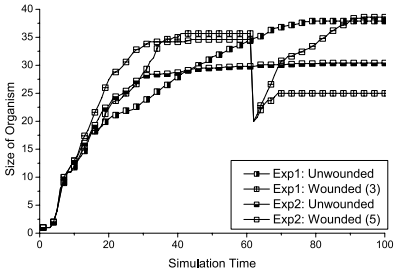

Fig. 7. Sample solutions - indicator

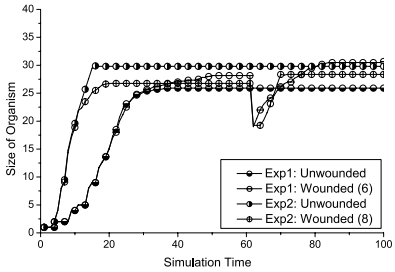

Fig. 8. Sample solutions + indicator

indicator fail to produce limited growth, but tend to use unlimited growth to cope with wounds.
If no additional indicator for wounds was present, only one solution out of 10 runs emerged that reacted to the wound in the second test case in at least half of the ten simulations performed to verify the behavior. If the additional indicator is used, three organisms evolved that reliably reacted to the wound in the second test case, see fig. 7 and fig. 8 for examples. In parenthesis the number of successful simulations of the second test case is given.

5 CONCLUSIONS AND FUTURE RESEARCH

We have shown that Evolutionary Algorithms can be used to optimize models of gene regulatory networks to solve the AE problem of limited growth and astonishingly only two genes seem to be necessary to achieve limited growth. When comparing RBNs and S-systems as models for gene regulatory networks, S-systems produce a more reliable behavior but seem to be harder to optimize for the Evolutionary Algorithm. RBNs on the other hand converge faster to a solution but are more prone to stochastic events due to the discrete nature of the state transition rules.
Regarding the ability of self-repair we were able to show that only an additional

Table 1. Model parameter for given examples

Two genes RBN fg. 4						Two genes S-system fg. 4						
I	S	T	R_{prod}	R_{dgr}	D	α	$\mathcal{G}$		β	$\mathcal{H}$		D
1;1	1010	0.00	1.00	1.00	0.18	6.35	1.06	1.33	0.0	0.0	-0.57	0.33
1;1	1110	0.50	0.32	0.00	0.15	2.31	-2.93	2.85	0.0	2.49	0.00	0.00

Three genes RBN fg. 7 exp. 2						Three genes RBN fg. 8 exp. 2					
I	S	T	R_{prod}	R_{dgr}	D	I	S	T	R_{prod}	R_{dgr}	D
0;1	1001	0.58	0.52	1.97	0.45	0;2	1101	0.97	1.00	0.71	0.04
0;1	0010	0.92	0.77	1.00	0.06	0;3	0101	0.94	0.56	0.59	0.49
1;1	0100	0.21	0.49	0.28	0.32	0;3	1000	0.73	0.98	0.48	0.13

biochemical, indicating the death of neighboring cells, enables the Evolutionary Algorithm to solve this problem more easily and more reliably.
In future experiments we want to apply the S-system to the problem of self-repair. Also we will focus on evolving organisms that differentiate into multiple cell types. Using RBN the cell differentiation can be determined using the attractor cycles of RBN. We also want to extend the cell mechanics to allow cell migration and tissue evagination to evolve more complex shapes.

References

1. Pankaj Agarwal. The cell programming language. *Artificial Life*, 2(1):37–77, 1994.
2. T. Akutsu, S. Miyano, and S. Kuhara. Algorithms for identifying boolean networks and related biological networks based on matrix multiplication and fingerprint function. *Journal of Computational Biology*, 7(3):331-343, 2000.
3. Hugo de Garis. Brain building: The genetic programming of artifcial nervous systems and artifcial embryos. in book 'Progress in Neural Networks, Vol. 4', ed. O.M. Omidvar, Ablex Publ. Corp., 1991.
4. Hugo de Garis. Artifcial embryology - the genetic programming of cellular differentiation. In *Artificial Life III Workshop*, Santa Fe, New Mexico, USA, June 1992.
5. Hugo de Garis. Artifcial embryology: The genetic programming of an artifcial embryo. In Branko Souček, editor, *Dynamic, Genetic, and Chaotic Programming*, pages 373–393. John Wiley, New York, 1992.
6. F. Dellaert. Toward a biologically defensible model of development. Masters Thesis, Case Western Reserve University, Dept. of Computer Engineering and Science, 1995.
7. F. Dellaert and R. Beer. A developmental model for the evolution of complete autonomous agents. In *From Animals to Animats IV, 4th International Conference on Simulation of Adaptive Behavior*, pages 393–401, Cambridge, MA., 1996. The MIT Press/Bradford Books.
8. Sharon Duvdevani-Bar and L. Segel. On topological simulations in developmental biology. *Journal of Theoretical Biology*, 131:33–42, 1994.
9. Peter Eggenberger. Cell interactions as a control tool of developmental processes for evolutionary robotics. In *From Animals to Animats IV, 4th International Conference on Simulation of Adaptive Behavior*, Cambridge, MA., 1996. The MIT Press/Bradford Books.

10. Peter Eggenberger. Creation of neural networks based on developmental and evolutionary principles. In *Proceedings of the International Conference on Artificial Neural Networks*, Lausanne, Switzerland, October 8-10 1997.
11. Peter Eggenberger. Evolving morphologies of simulated 3d organisms based on differential gene expression. In P. Husbands and I. Harvey, editors, *Proceedings of the 4th European Conference on Artificial Life*, Cambridge, MA., 1997. The MIT Press.
12. Kurt Fleischer. Investigations with a multicellular developmental model. In C.G. Langton and K. Shimohara, editors, *Artificial Life V: Proceedings of the 5th International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press, 1996.
13. Kurt Fleischer and Alan H. Barr. A simulation testbed for the study of multicellular development: Multiple mechanisms of morphogenesis. In *Artificial Life III*. Addison-Wesley, 1993.
14. Pablo Funes and Jordan B. Pollack. Computer evolution of buildable objects. In P. Husbands and I. Harvey, editors, *Proceedings of the 4th European Conference on Artificial Life*, pages 358–367, Cambridge, MA, 1997. MIT Press.
15. A. Gierer and M. Meinhardt. A theory of biological pattern formation. Kybernetik 12, 30-39, 1972.
16. Gregory S. Hornby and Jordan B. Pollack. The advantages of generative grammatical encodings for physical design. In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001)*, pages 600–607, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 2001. IEEE Press.
17. D.H. Irving and M.A. Savageau. Efficient solution of nonlinear ordinary differntial equations expressed in s-systems canonical form. *SIAM Journal of Numerical Analysis*, 27:704–735, 1990.
18. S. A. Kauffman. *The Origins of Order*. Oxford University Press, New York, 1993.
19. K. Murphy and S. Mian. Modelling gene expression data using dynamic bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA., 1999.
20. Przemyslaw Prusinkiewicz. Modeling and visualization of biological structures. In *Proceeding of Graphics Interface*, pages 128–137, Toronto, Ontario, 19-21 May 1993.
21. Marc Schoenauer. Shape representations and evolution schemes. In L. J. Fogel, P. J. Angeline, and T. Back, editors, *Proceedings of the 5th Annual Conference on Evolutionary Programming*, pages 121–129, Cambridge, MA, 1996. MIT Press.
22. Bruce E. Shapiro and Eric D. Mjolsness. Developmental simulations with cellerator. In *Proceedings of the 2nd International Conference on Systems Biology*, pages 342–351, 2001.
23. Daisuke Tominaga, Nobuto Kog, and Masahiro Okamoto. Efficient numeral optimization technique based on genetic algorithm for inverse problem. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO2000)*, pages 251–258. Morgan Kaufmann, 2000.
24. D. C. Weaver, C. T. Workman, and G. D. Stromo. Modeling regulatory networks with weight matrices. *Proceedings of the Pacific Symposium on Biocomputing*, 1999.
25. A. Wuensche. Genomic regulation modeled as a network with basins of attraction. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, volume 3, pages 89–102, Singapore, 1998. World Scientific Press.