



**HAL**  
open science

# A Multi-Cellular Developmental System in Continuous Space using Cell Migration

Nicolas Bredeche

► **To cite this version:**

Nicolas Bredeche. A Multi-Cellular Developmental System in Continuous Space using Cell Migration. Simulation of Adaptive Behavior, Jul 2008, Osaka, Japan. inria-00288037

**HAL Id: inria-00288037**

**<https://inria.hal.science/inria-00288037>**

Submitted on 13 Jun 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Multi-Cellular Developmental System in Continuous Space using Cell Migration

Nicolas Bredeche<sup>1</sup>

TAO / LRI ; Univ Paris-Sud, CNRS, INRIA Futurs  
Université de Paris-Sud - LRI, Bat. 490 - F-91405 Orsay Cedex, France  
bredeche@lri.fr ; <http://www.lri.fr/~bredeche>

**Abstract.** This paper introduces a novel multi-cellular developmental system where cells are placed in a continuous space. Cells communicate by diffusing and perceiving substances in the environment and are able to migrate around following affinities with substance gradients. The optimization process is performed using Echo State neural networks on the problem of minimizing tile size variations in the context of a tiling problem. Experimental results show that problem complexity only impacts the number of substances used, rather than the number of cells, which implies some sort of scalability with regards to the size of the phenotype. Symmetry breaking and robustness are addressed by adding noise as an intrinsic property of the model. A (positive) side effect is that the resulting model produces very robust solutions with efficient self-healing behavior in the presence of perturbations never met before.

## 1 Introduction

Evolutionary Design deals with the optimization of structures such as objects, buildings or robot morphologies (to name a few). A key problem in this context is the search space: it is now widely accepted that indirect encoding approaches[16, 10, 2] (ie. representing a construction plan) may yield to better results in some cases than direct encoding (ie. representing the construction itself). However, these approaches failed to address scalability issues since the size of the construction plan still grows with the size of the construction itself. Recent works on developmental systems have taken a step further by optimizing the very developmental process rather than the construction plan. This new approach is often referred to as multi-cellular Artificial Ontogeny[4, 17] and addresses both scalability and robustness issues.

A multi-cellular developmental system is defined as a dynamical system with a group of interacting cells. These cells communicate by emitting and receiving substances in the environment (e.g. concentration of chemicals) and may perform a given number of operations depending on substances concentration. The most basic operation is cell differentiation, where a cell appearance may change to a given state (e.g. color, size or shape of a cell). At each step of development, all cells are updated synchronously until a halting criterion is met (e.g. maximum number of development steps, stability measure, etc.). When development stops,

the state of all cells is considered as the resulting phenotype. The important point is that all cells share the same duplicated controller, which results in different output values depending on each cell current context.

Some few works have addressed evolutionary design problems using a multi-cellular approach, ranging from evolution of neural networks [11] to artificial creatures morphologies[4] and gene regulatory networks[3]. However, these setups make it difficult to study the internal dynamics of such systems and most recent works have focused on simpler problem. The basic setup for studying such developmental systems, proposed by [13], is to consider a matrix of pixels (ie. an image) where each pixel contains one cell. Each of these cells may emit and receive substance concentration to neighboring cells, usually in either a von neumann or moore neighborhood fashion. Evaluation of a developmental model is usually performed by optimizing the cell controller so that the resulting phenotype matches a given image pattern.

This paper proposes an extension of multi-cellular developmental systems in the context of a continuous environment. In this scope, development is considered both through time and space: cells communicate with one another by diffusing substances in the environment and are either attracted or repulsed along the gradient of the diffused substances. Each cell perceives substance concentration at its position in a totalistic fashion so that the search space is dependant only on the number of substances in the environment rather than the number of cells. As a result, a simple yet large phenotype may be build with very few substances. Another key issue of our model is that symmetry breaking and robustness towards noise are both addressed by featuring noise as an intrinsic property of the environment. Such a model makes possible to address problem where cell positioning is necessary such as group behaviors in swarm robotics, mobile sensor optimal positioning, light sources positioning in an architectural building, etc.

In section 2, a model for multi-cellular developmental system in the continuous space is described. Cell migration, symmetry breaking and controller issues are also addressed. Section 3 presents an implementation of the model and describes the benchmark problem used for evaluation. Section 4 gives the experimental settings as well as results regarding both optimization performance and self healing capabilities. Then, section 5 provides a comparaisn between our model and multi-cellular developmental models from the litterature and establish some links with some application problems such as mobile sensor deployment and diffusion of a swarm of robots

## 2 Development in a continuous space

Our model for multi-cellular development addresses the problem of development of a group of moving cells in a continuous environment. In this setup, cells are placed in a continuous environment and are free to move around (ie. cell migration). Cell communication is performed through emitting and perceiving substance concentrations from the neighborhood. Cell substance emission is performed through radial diffusion. Substance concentration decreases depending on

the distance to the emitting cell. As a result, a cell neighborhood is defined in a given radius and neighboring cells influences depend on distance to the target cell. Substance concentrations is thus perceived in a totalistic fashion, ie. the number of inputs for a cell depends on the number of substances only.

## 2.1 Cell migration

Cell migration refers indirectly to a coordinate system. In this model, cell migration is performed depending on available substances in the environment, limited to the cell perception radius. A cell controller determines current *affinities* with substance concentrations in the environment. Affinities range between  $-1.0$  (ie. repulsed) and  $+1.0$  (ie. attracted), if affinity is close to  $0.0$ , the cell maintains its current position. By considering each substance affinities with substance actual concentrations nearby the cell at hand, it is possible to compute a migration vector (direction and (bounded) force) for each cell. Given  $N$  substance concentrations perceived by a cell, the cell migration vector  $\vec{\Delta}_{c_0}$  for one specific cell  $c_0$  at time  $t$  is computed as follow:

$$\vec{\Delta}_{c_0} = f_{max}(\sum_{i=1}^{N_s} (aff_{c_0}^i * \sum_{k=1}^{N_c} (\overrightarrow{(x_{c_k}^{N_d} - x_{c_0}^{N_d})}) * f_{\theta_{c_0}^{rec}(i)}(\theta_{c_k}^{emit}(i), d(x_{c_0}^{N_d}, x_{c_k}^{N_d}))))$$

Roughly, this means that the migration vector for a given cell is computed at each time step from the cell affinities with substances produced by neighboring cells within a predefined radius, given that perceived substance concentrations depend both on distance and neighboring cells production. Notations are as follow:  $c_0$  is the target cell,  $\Delta_{c_0}$  is the localization update for the current iteration,  $N_d$  is the number of space dimensions,  $N_s$  is the number of substances used for communication,  $N_c$  is the number of cells in the (possibly limited) neighborhood of  $c_0$ ,  $x_{c_k}^{N_d}$  is the coordinate of cell  $c_k$  in the  $N_d$ -dimensional space, and  $d(x_{c_1}^{N_d}, x_{c_2}^{N_d})$  is the euclidian distance between cell 1 and cell 2.  $aff_{c_0}^i$  state affinity of cell  $c_0$  with substance  $i$  and  $f_{max}$  is a function to limit the norm of the resulting migration vector such as, for example, a simple maximum norm threshold.  $\theta_{c_k}^{emit}(i)$  is the concentration value of substance  $i$  produced by cell  $c_k$  and  $f_{\theta_{c_0}^{rec}(i)}$  is the corresponding concentration value *perceived* by cell  $c_0$ . In this setup,  $\vec{\Delta}_{c_0}$  determines the instant speed to update the position of cell  $c_0$ <sup>1</sup>. Step 2 of figure 1 illustrates the cell migration operation in a 2D world.

A key feature of our model is that the coordinate system is generated by the developing system itself as there is no reference to any external coordinate system - for example: given three cells and three substances, these cells may form an equilateral triangle by each emitting one specific substance (through radial diffusion) and migrating so that perceived concentrations of the two other substances is equal. As a consequence, this triangle of cells result in a non-ambiguous two dimensional space where each point in the 2D space is defined by a unique triplet of substance concentrations.

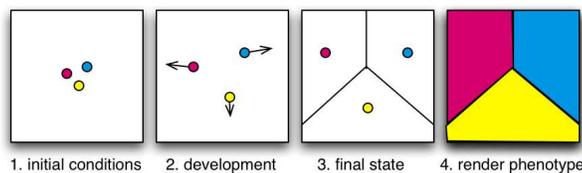
However, two (or more) cells may be located at the exact same position. In this case, cells potentially receive exactly the same inputs and may produce

<sup>1</sup> However it should be noted that  $\vec{\Delta}_{c_0}$  could be used to determine acceleration, for example.

exactly the same outputs thereafter. This is what happens if all cells start at the same initial position: at each step, all cells behave in a mimetic fashion. In order to avoid this, it is necessary to introduce a random perturbation during cell migration, ie. each cell position is updated at each step according to its migration affinities and migration noise.

Because substance concentrations are perceived in a totalistic fashion, a cell may have  $N_s$  inputs (perceived substance concentrations) and  $2 * N_s$  outputs (diffused substance concentrations + substance affinities for migration), with  $N_s$  the number of substances in the environment. If cell differentiation is enabled (which is not the case in the scope of this paper), additional outputs may be required (e.g. cell color/size/shape state).

### 3 Optimization of tiling problems



*Fig. 1:* Example of development of an embryo to produce a voronoi diagram. Starting with an initial condition where cells are placed in an environment (step 1), cells migrate by communicating with each other (step 2) so as to reach a stable state (step 3). At this point, the resulting embryo may be rendered as a phenotype which can be evaluated (step 4 - an example of voronoi diagram rendering).

The objective function is defined as follow: given  $N_c$  cells in an environment, the goal is to position the cell in the environment such as each cell determines a voronoi site. The phenotype is then evaluated as the voronoi diagram built from these voronoi sites and the goal is to minimize the standard deviation  $\sigma$  of the average of the resulting voronoi regions. More formally, the objective function is the following:

$$fitness(x) = \sqrt{\frac{1}{N_c} * \sum_i^{N_c} (surface_i - \overline{surface})^2}$$

With the following notations:  $x$  is the genotype to be optimized;  $N_c$  is the number of cells (ie. voronoi sites);  $surface_i$  the surface covered by the  $i^{th}$  voronoi region and  $\overline{surface}$  the average surface of all voronoi regions. This benchmark problem is referred to as the "tiling problem" since the goal is to tile space with possibly repeating patterns, or at least with patterns of the same size (if not of the same shape). Figure 1 shows an example of development from the initial starting point which results in a voronoi diagram where cell positions act as voronoi sites. Moreover, the environment is considered as a non-bounded 2D world (ie. a toroidal world) in order to avoid interfering with cell migrations. This

makes it possible to take into account the fact that the very coordinate system of our model implies that cell positions are dependant only on one another (i.e. not related to the image translation/orientation as it is perceived after rendering).

Evaluation is performed as follow: development starts from an initial state where  $N_c$  cells are placed in the center of the image. As stated in the previous section, cells move away from one another because of migration noise, then each cell is able to behave in a different fashion as it is experiencing a different environmental context. Development stops when a termination criterion is met. In this particular setup, development stops after a maximum number of iterations, as proposed in other works[13, 14, 9]. Then, the resulting phenotype is evaluated by rendering the corresponding Voronoi diagram and computing the aforementioned fitness value. Voronoi sites are located in a continuous environment and Voronoi regions are rendered (and evaluated) in a discrete environment. Both migration and rendering assumes that the environment is a toroidal world.

In practical, voronoi regions are rendered in a  $32x32$  or  $128x128$  RGB image (depending on the number of cells) using cell coordinates in the environment. At each iteration  $it$ , cell position update ( $\delta$ ) results from the combination of random noise perturbation and cell migration vector as follows:

$$\vec{\delta}_{it}^{c_0} = \frac{max_{it}-it}{max_{it}} * \vec{perturbation} + \vec{\Delta}_{it}^{c_0}$$

Cell migration maximum speed ( $\vec{\Delta}_{it}^{c_0}$ ) is bounded and scaled so that a cell may not move faster than one pixel per iteration. Cell random noise perturbation ( $\vec{perturbation}$ ) starts with a maximum bounded at 1.0 pixel-equivalent size and decreases linearly over time so as to reach zero influence when development stops ( $max_{it}$  is the number of development iterations).

## 4 Experiments

In order to evaluate our model, a set of experiments with a different number of cells have been conducted. The number of cells is either 3 or 16 cells (corresponding to tiling the environment with 3 or 16 voronoi regions) with only one substance for communication. The perception radius for each cell is set to a little more than twice ( $= 2.15$ ) the radius of the optimal surface voronoi regions should cover. It also ensures that each cell is still able to communicate with its neighbors. Finally, this means that some cells may possibly not be able to directly communicate with one another depending on their respective position.

The experiments presented hereafter evaluate the relevance (or not) of an additional internal substance which is different to each cell and is set with a unique read-only value. This makes it possible to number each cell in a unique fashion, so that the controller has access to a cell identification number.

An echo state network with a reservoir size of 10 (with a connection density of 0.2 and damping factor set to 0.8) is used as controller<sup>2</sup>. The state-of-the-art

<sup>2</sup> ESN can be defined as a discrete-time recurrent neural network based on the reservoir computing framework known to perform very well in the context of temporal data. The reader may refer to [12] for a full description of ESN and to [7] for application in control problem related to multi-cellular developmental systems.

CMA Evolution Strategy algorithm with restart feature[1] is used to optimize the set of neural network weights. The genotype size depends on the number of inputs and outputs as well as the size and connectivity of the hidden layer and is set here to 20 (ESN with or without cell identification - while number of inputs varied, only weights from the reservoir to the output nodes are considered for optimization). In preliminary experiments, classic multi-layered perceptron (MLP) was also considered (results not shown here). MLP used 6 hidden nodes so that the number of dimensions of the optimization problem remains the same as with ESN. Results were shown to be comparable in the end of the evolution process with that obtained with ESN (which implies that temporal capability is not crucial), but convergence tends to be slower than with ESN.

The number of iterations for the development process of one genotype is set to 256 (note that less than 32 iterations is enough for one cell to travel across the whole environment) and noise perturbation is set as stated before. In order to compute an accurate performance, evaluation is averaged from 2 tries so as to smooth the effect of random perturbation, which is particularly useful in the first steps of optimization where genotypes either fail to feature robustness wrt. developmental perturbations or suffer from the discretization process during rendering<sup>3</sup>. All experiments in the next section are limited to 1000 evaluations (approx. 60 generations). CMA-ES automatically selects population size (initial population size is 12) and default restart parameters are used.

#### 4.1 Result, Robustness towards perturbations and scaling issues

All results are shown in table 2. All figures show results of the best individuals from 11 independent runs (median best, best of best, worst of best and 25% and 75% quartiles for each generation) - the y-axis features the fitness values and the x-axis features the number of evaluations (rather than the number of generations). Columns: either 3 or 16 cells problems ; Rows: without or with Cell Identifier. Random development figures are given as a naive baseline, and are always quickly outperformed. From these results, two main considerations emerge: (1) a cell unique identifier does not provide any advantage, which is no surprise in this context since the cell population may be homogeneous and there is no clear gain as to identify each cell explicitly ; (2) Problem complexity seems not to be related with the number of cells as the 3 cells problem even show slightly slower convergence, which may be explained by the fact that each cell must cover a wider region (implying more freedom of migration).

In order to evaluate robustness towards noise, best individuals from all the runs were put to the test by applying two kind of perturbations: development is started with either (1) all cells positioned at the same location, as used

---

<sup>3</sup> Fitness values for the same phenotype may vary slightly since Voronoi diagram rendering leads to slightly different phenotypes depending on cell positions because of the discretization process that results in a 32x32 or 128x128 image. Evaluation for the 16 cells problem is performed in 128x128 so as to provide comparable fitness figures with that of the 3 cells problem.

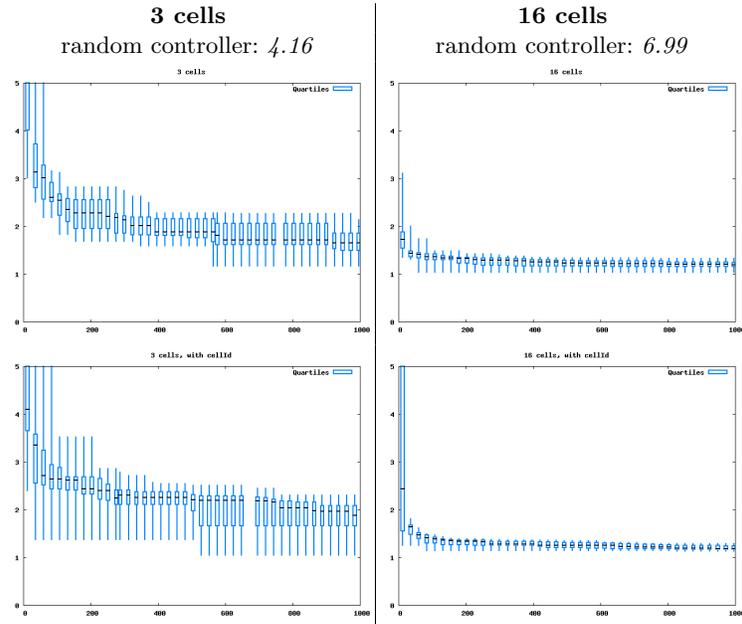
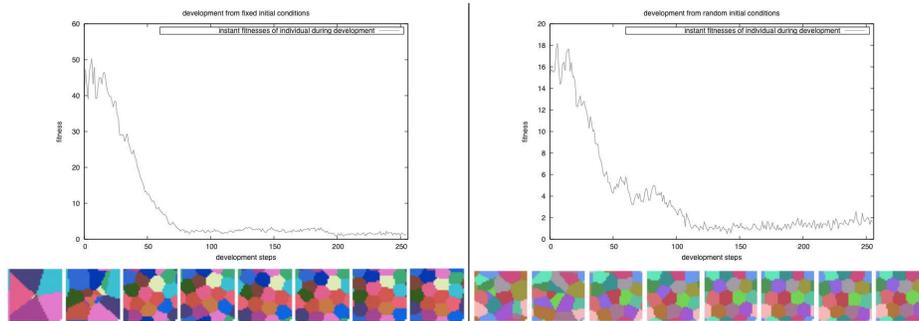


Fig. 2: Summary of results (see text for details).

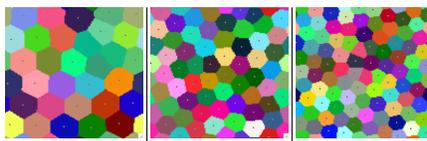
during optimization (ie. intrinsic random perturbation may lead to a different outcome); or (2) cells are randomly placed in the environment. In both case, the resulting phenotype is evaluated and fitness values are compared to the predicted fitness values recorded during evolution. Development in case 1 shows robustness toward noise for all runs, which is no surprise since this setup is very close to what was used during optimization (which confirms that individuals are able to cope with the noisy fitness function). Case 2 leads to similar results - which is more difficult since robustness wrt. random starting positions was not considered during optimization. In this latter case, all individuals showed great robustness and converged towards phenotypes that displayed the awaited fitness, both in the 3 and 16 cells setup, whatever the starting initial cell positions. Figure 4 shows an example monitoring the development from an optimized genotype for the 16 cells problem. The instant fitness value at each of the development step is traced in both case and the development course for the two experimental conditions can be compared: in case 2, convergence is not as straight-forward as in case 1 at first, but the cell population displays an efficient self-healing behavior as it quickly recovers from the initial localization perturbation and end up with the same fitness value as expected. In both case, the best fitness value is also reached before the end of development.

Lastly, the scaling problem was considered. Figure 4 shows the final stage of development of the best individual from the 16 cells problem but in the context of 32, 64 and then 96 cells. Indeed, optimized controllers produced behaviors



*Fig. 3:* Example of robustness towards development for the 16 cells problem (best individual). Left: development with centered voronoi sites (ie. all cells start from the same position). Right: development from random starting positions (all cells start from a different position). the y-axis gives the instant fitness value (translation/rotation invariant) of the phenotype and the x-axis gives the number of development steps so far for this individual. Voronoi region colors are randomly assigned and are used only for visualization.

very close to a simple repulse behavior (where each cell flies away from substance concentration) combined with a threshold limit so as to maintain a stable distance between cells, which is very simple yet efficient solution for this problem and scales up easily. It is interesting to note that resulting phenotype closely resemble a regular tiling, which is the optimal solution whenever it is possible (i.e. when all regions have either 3,4 or 6 borders, which may not be the case for the problems at hand).



*Fig. 4:* Scaling up to 32, 64 and 96 cells (best controller from the 16 cells setup).

## 5 Discussion and Conclusions

This paper introduces a new model for multi-cellular artificial embryogeny in a continuous space for optimization. The proposed model relies on cell migration based on substance affinities rather than explicit cell neighborhood and noise so as to break possible symmetries. Results showed that problem difficulty in our model is related to the number of substances rather than the number of cells, which is a key feature with regards to scalability since the size of the

cell population is independent from problem complexity. It was also shown to produce a robust and, to some extent, scalable behavior at least for the problem at hand.

Other models have already addressed development of patterns as benchmark problem for artificial embryogeny. While the main feature of our model is that we consider development in a continuous environment with cell migration, it shares some similarities with other models. Firstly, Gordon and Bentley’s model[9] also relies on a totalistic approach to compute information perceived by a specific cell as neighboring cell states are summed. However, this implies breaking symmetry by introducing some bias as initial starting condition. This differs from our model because symmetry breaking is automatically handled through the combination of additional random positional noise and cell relative positioning through selected affinities. The model presented here is able to automatically build from scratch a coordinate system. A direct advantage of such a system is that the number of substances is directly related to the problem complexity, and *not* to the number of cells (ie. scalability wrt. size). Secondly, Miller’s model[13] already considered spatial development so as to limit the size of the cell population wrt. to the task at hand. In this setup, a cell would trigger cell division by growing a new cell on a nearby free site targeted in an explicit coordinate system (in this setup: North, South, East, West). As a consequence, the population of cells would grow spatially even if each cell position is fixed at birth (no cell migration involves). While it is possible to position a cell with a specific state at one precise location, a possibly large population may be needed. Thirdly, Miller[13], Federici[8] as well as our previous work in [6] addressed the issue of self-healing, ie. robustness of development toward noise. In [6], we showed that impressive results could be achieved by adding a strong penalization to individuals that could not reach a stable state<sup>4</sup> *before* the maximum number of development steps allowed. This feature provided very reliable controllers which were able to completely recover even from a 100% noise perturbation. In the current model, stability is also achieved but in a different fashion: positional noise perturbation is intrinsic to the environment and robustness is thus forced from the beginning and results in very robust phenotypes even if perturbation occurs.

The work presented here has also some connections with the problem of dispersion of a swarm of robots (or, more generally, particles). Swarm dispersion is a distributed coverage control problem defined by a clustering of the environment where each agent applies a local strategy based on information from its neighborhood. In this context, both homogeneous and heterogeneous dispersions have been addressed in simulation[5] and with a real-world robot swarm[15]. To some extent, our model can be related to these approaches and is indeed more general in the sense that it relies only on raw information and does not require explicit identification of cells/particles. Of course some fundamental differences still exist since our model does not rely on *physical* cells that are subject to envi-

---

<sup>4</sup> in this previous work, a stable state is reached when all neural network controller internal activity does not change for a given number of iterations.

ronmental dynamics when moving around and that noise is artificially added to the model. However, it is considered as a promising direction for future works.

## References

1. A. Auger and N. Hansen. A restart cma evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1769–1776, 2005.
2. P. Bentley and S. Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In W. B. et al., editor, *GECCO'99*, pages 35–43. Morgan Kaufmann, 1999.
3. J. Bongard. Evolving modular genetic regulatory networks. In D. B. F. et al., editor, *CEC 2002*, pages 1872–1877. IEEE Press, 2002.
4. J. Bongard and R. Pfeifer. Evolving complete agents using artificial ontogeny. In B. Springer-Verlag, editor, *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, pages 237–258, 2003.
5. J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.
6. A. Devert, N. Bredeche, and M. Schoenauer. Robust multi-cellular developmental design. In D. Thierens et al., editor, *GECCO'07*. ACM Press, 2007. Preprint available at <http://hal.inria.fr/inria-00145336/en/>.
7. A. Devert, N. Bredeche, and M. Schoenauer. Unsupervised learning of echo state networks: A case study in artificial embryogeny. In *8th International Conference on Artificial Evolution (Evolution Artificielle - EA'07)*, 2007.
8. D. Federici and T. Ziemke. Why are evolved developing organisms also fault-tolerant? In *SAB'06*, pages 449–460, 2006.
9. T. G. W. Gordon and P. J. Bentley. Bias and scalability in evolutionary development. In *GECCO '05*, pages 83–90. ACM Press, 2005.
10. F. Gruau. Genetic synthesis of modular neural networks. In *ICGA-93*, pages 318–325. Morgan Kaufman, 1993.
11. P. E. Hotz. Evolving morphologies of simulated 3d organisms based on differential gene expression. In P. Husbands and I. Harvey, editors, *ECAL97*. MIT Press, 1997.
12. H. Jaeger. Tutorial on training recurrent neural networks. Technical report, GMD Report 159, Fraunhofer Institute AIS, 2002.
13. J. F. Miller. Evolving a self-repairing, self-regulating, french flag organism. In *GECCO*, pages 129–139, 2004.
14. D. Roggen and D. Federici. Multi-cellular development: Is there scalability and robustness to gain? In X. Y. et al., editor, *PPSN'04*, pages 391–400. LNCS 3242, Springer Verlag, 2004.
15. M. Schwager, J. McLurkin, and D. Rus. Distributed coverage control with sensory feedback for networked robots. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.
16. K. Sims. Evolving virtual creatures. In *SIGGRAPH'94*, pages 15–22. ACM Press, July 1994.
17. K. O. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.