A Morphogenetic Evolutionary System: Phylogenesis of the POEtic Circuit

Daniel Roggen, Dario Floreano, and Claudio Mattiussi

Autonomous Systems Laboratory, Institute of Systems Engineering, EPFL, Lausanne, Switzerland
http://asl.epfl.ch
name.surname@epfl.ch

Abstract. This paper describes a new evolutionary mechanism developed specifically for cellular circuits. Called morphogenetic system, it is inspired by the mechanisms of gene expression and cell differentiation found in living organisms. It will be used as the phylogenetic (evolutionary) mechanism in the POEtic project. The POEtic project will deliver an electronic circuit, called the POEtic circuit, with the capability to evolve (Phylogenesis), self-repair and grow (Ontogenesis) and learn (Epigenesis). The morphogenetic system is applied to the generation of patterns and to the evolution of spiking neural networks, with experiments of pattern recognition and obstacle avoidance with robots. Experimental results show that the morphogenetic system outperforms a direct genetic coding in several experiments.

1 Introduction

The POEtic circuit is a multi-cellular electronic circuit composed of a regular 2D array of cells (i.e. functional units), which has capabilities to evolve (Phylogenesis), self-repair and grow (Ontogenesis) and learn (Epigenesis). The POEtic circuit is reprogrammable and its functionality comes from its configuration bits or genotype, which will be evolved using the phylogenetic mechanism described herein. To implement self-repair and growth, each cell may contain the genome of the whole circuit. In addition, each cell has virtual input/output connections to sensors and actuators. The POEtic circuit will be used in a wide range of applications, including autonomous mobile robotics. A more detailed description of the project is given in [10].

This paper is the first accounting of the morphogenetic evolutionary system which will be implemented in the POEtic circuit. This system is designed to be computationally simple (hardware implementation) and suited to multi-cellular circuits like POEtic. The morphogenetic system relies on a simple signalling mechanism and expression rules to encode the functionality of a cellular system which can be a neural network or any other cellular electronic circuit (e.g. the BioWall [9]). Its development is motivated by the dynamical reconfiguration needs of the POEtic circuit, which cannot be easily handled by direct genetic codings with one-to-one mappings between the genotype and the phenotype [11].

In this paper we describe the morphogenetic system and compare it with a direct genetic encoding in experiments of pattern coverage and in the evolution of spiking neural networks for pattern recognition and obstacle avoidance using real miniature robots. Results show that the morphogenetic system outperforms direct genetic encoding by finding individuals of better fitness in less generations. Analysis suggests that the morphogenetic system generates more individuals of higher fitness than the direct genetic encoding, thus easing the evolutionary process.

2 Morphogenetic Coding

Introduction The morphogenetic system is motivated by the requirement of the POEtic circuit to have a phylogenetic mechanism which is suited to its cellular structure and its capacity to dynamically reconfigure, e.g. when errors are detected, when the environment changes or when the circuit is expanded with new cells. Direct genetic encodings are not suited for that kind of architecture because the number of elements in the system must be known in advance and cannot change throughout the life of the system.

The morphogenetic system assigns a functionality to each cell of the circuit from a set of predefined functionalities (something akin to skin, muscle, neuron cells, etc. in living organisms). The process works in two phases: first a signalling phase then an expression phase. The signalling phase relies on the ability of the cellular circuit to exchange signals among adjacent cells to implement a diffusion process. The second phase, expression, finds the functionality to be expressed in each cell by matching the signal intensities in each cell with a corresponding functionality stored in an expression table. The genetic code contains the position of diffusing cells (diffusers) and the content of the expression table, which are both evolved using a genetic algorithm. The morphogenetic system is inspired by the mechanisms of gene expression and cell differentiation of living organism [2], notably by the fact that concentrations of proteins and inter-cellular chemical signalling regulate the functionality of cells. Other works related to this morphogenetic systems include L-Systems [7], use of developmental processes [3, 6], and also approximate representations [1] seeking to compress the genotype.

Description The morphogenetic system relies on a set of predefined functionalities which is called a family of functions. The family of functions must offer a rich enough repertoire of functionalities to realize the desired circuit. The family which is used when evolving neural networks on the POEtic circuit is composed of spiking neurons with different connectivity patterns, and either excitatory or inhibitory characteristics (fig. 1). Spiking neurons are used because they display rich non-linear dynamics and because they are well suited for implementation in digital circuits [5].

The cellular circuit allows for signals to be exchanged between adjacent cells. A signal is a simple numerical value (the signal intensity) that the cell owns, and that adjacent cells are able to read (the intensity of signal s in cell i is

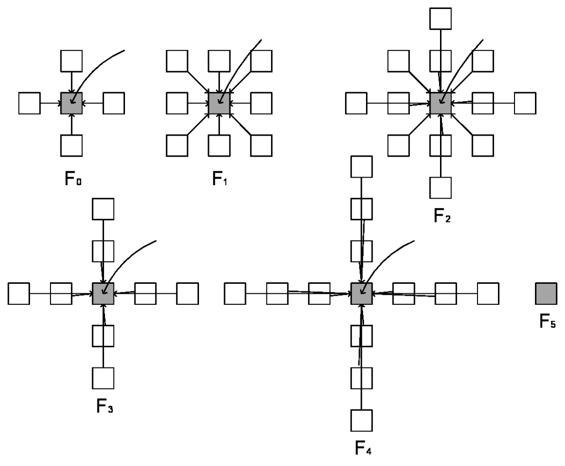


Fig. 1. In the current implementation, a family composed of 12 functions is used when evolving neural networks. The functions are spiking neurons with different connectivities (the 6 types of connectivity shown in the figure) and with either excitatory or inhibitory characteristics. Each cell of the multi-cellular circuit implements a single neuron, shown in gray. It receives inputs from neighboring neurons (outlined), which are implemented in neighbouring cells. Each neuron has an extra external input (curved arrow). Neuron F_5 is equivalent to a void cell. At the boundary of the cellular array the connectivity is truncated (no periodic boundary condition).

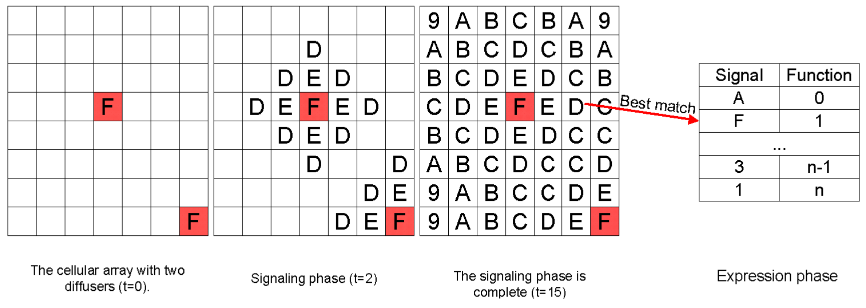


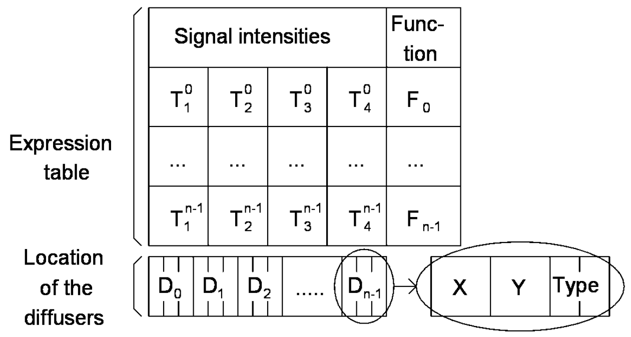
Fig. 2. The three arrays on the left are snapshots of the signalling phase with one type of signal and two diffusers (gray cells) at the start of the signalling phase (left array), after two time steps (middle) and when the signalling is complete (right). The number inside the cells indicates the intensity of the signal in hexadecimal. The expression table used in the expression phase is shown on the right. In this example the signal D matches the second entry of the table with signal F (smallest Hamming distance), thus expressing function F_1.

noted by C_s^i). Special cells, called *diffusers*, own a signal of maximum intensity. The signal intensity in the neighbouring cells then decreases linearly with the Manhattan distance to the diffuser. In the current implementation four type of signals (each represented by a 4-bit number) are used. They are diffused independently, without interaction among them. Figure 2 shows an example of the signalling phase in the case of a single type of signal, with two diffusers placed in the cellular circuit.

The expression phase assigns a function to each cell by matching the signal intensities inside that cell with the entries of an expression table T (fig. 2) stored in the genetic code. Each entry of the table contains the intensities of the four signals and the function to express in case of match. The intensity of signal s in the entry j of the table is noted by T_s^j. A cell i is said to match an entry j of the expression table when the distance $d = \sum_{s=1}^{4} DOp(C_s^i, T_s^j)$ is minimum. The distance operator DOp is the Hamming distance.

The genetic code contains two parts. The first part is the expression table T and the second is the location of the diffusers (fig. 3). The expression table contains 16 bits for each entry (4 signals coded on 4 bits each). Each entry corresponds to a predefined function. The functions are not encoded and evolved in these experiments. The location of the diffuser is stored as a pair of X,Y coordinates, plus two bits indicating the type of the diffuser (i.e. $2^2 = 4$ types of diffuser).

A genetic algorithm is used to evolve the morphogenetic coding. In all of the experiments presented here the population is composed of 50 individuals, selection consists of rank selection of the 15 best individuals, the mutation rate is 1%, one-point crossover rate is 20% and elitism is used by copying the 5 best individuals without modifications in the new generation.



Expression table
Signal intensities
Function
T_1^0 T_2^0 T_3^0 T_4^0 F_0
... ...
T_1^{n-1} T_2^{n-1} T_3^{n-1} T_4^{n-1} F_{n-1}
Location of the diffusers
D_0 D_1 D_2 ..... D_{n-1}
X
Y
Type

Fig. 3. The genetic code contains two parts. The first is the expression table T, here with n entries. Each entry is 16 bits long. The second part contains the location of the diffusers and their type. The number of bits for the X and Y coordinates depends on the size of the network. The type of the diffuser is encoded on 2 bits.

3 Pattern coverage experiment

This experiment is aimed at testing whether the morphogenetic system can generate circuits with diversified structures (without any specific functionality). It consists of covering an array of 8x8 cells with a specific binary pattern. Four different patterns are tested (fig. 4). A family of two cell functions (black or white cell) is used. The morphogenetic system uses two entries in the expression table and 16 diffusers. Preliminary tests with numbers of diffusers in the range of 2 to 64 have shown that a too low or a too high number of diffusers hinder the performance of the morphogenetic system with some patterns. Values in the range of 5 to 20 showed quite similar performance. As a good compromise, 16 diffusers have been chosen. The size of the morphogenetic coding is 160 bits: 2 entries in the expression table * 16 bits + 16 diffusers * 8 bits (6 bits to store the coordinates and 2 bits to store the type of the diffuser). Also, a direct genetic encoding is used for comparison, which consists of a binary string with 1 bit per cell (indicating whether the cell is black or white), resulting in a code of 64 bits.

The figure 4 shows the evolution of the maximum normalized fitness, which is proportional to the number of cells covered correctly, for the *mixed1* and *mixed2* patterns, averaged over 50 runs. The evolution of the fitness for the *uniform* and *checkerboard* patterns is very similar to the *mixed1* pattern. In the first few generations, the maximum fitness increases notably faster with the morphogenetic system than with the direct coding with all four patterns. With the *uniform*, *checkerboard* and *mixed1* pattern the morphogenetic system covers the patterns almost three times faster than the direct coding in terms of generation number (about 10 generations instead of 30). The morphogenetic system cannot fully cover the *mixed2* pattern, but it comes very close. This seems to indicate that some type of regularity is necessary for the morphogenetic system to perform well, or that the system has some biases toward some types of pattern. Further investigation are necessary to clarify this point. However, in the context of the POEtic circuit this may not be an issue, as the epigenetic (learning) mechanism may deal with structures at a finer granularity level, while the phylogenetic mechanism deals with structures at a coarser level. Further experiments may also be performed to assess how the morphogenetic system scales with larger arrays.

4 Spiking neuron model

The spiking neuron model used in the following experiments is a discrete-time, integrate-and-fire model with leakage and a refractory period. Each neuron has weighted inputs (+2 or -2 depending on whether the presynaptic neuron is excitatory or inhibitory) from connected neurons. Each cell can express one neuron type, from a family of 12 (6 connectivity patterns times 2 sign types) shown in fig 1. It has one more connection from an external input, e.g. to connect from a sensor, with fixed weight of +10. The neuron integrates the incoming spikes in the membrane potential, according to the weights of the connections. Once the

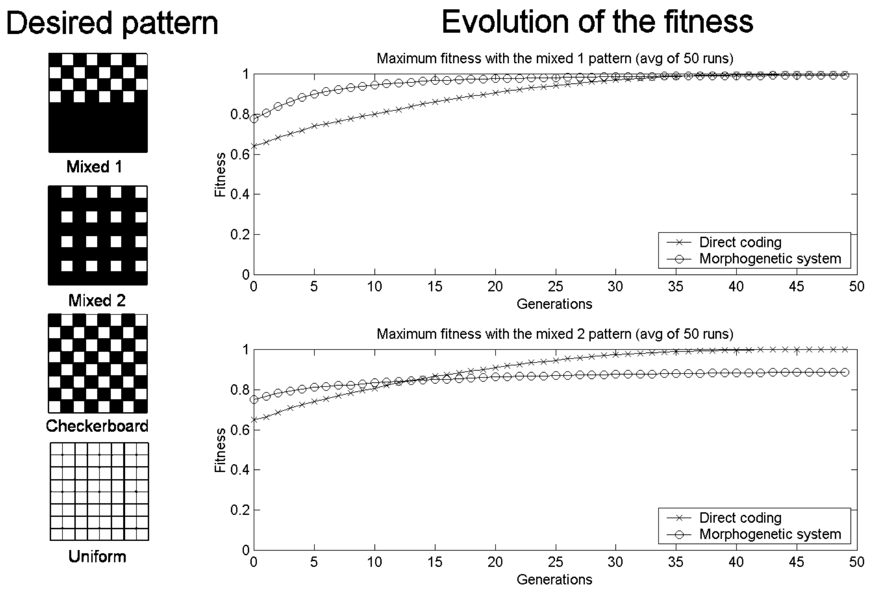


Fig. 4. The pattern coverage experiment consists in covering an array of 8x8 cells with a binary pattern. The left column shows the four desired pattern (*mixed1*, *mixed2*, *checkerboard* and *uniform*). The right column shows the evolution of the maximum fitness for the *mixed1* and *mixed2* patterns.

membrane potential reaches a threshold (fixed to 4), the neuron fires (emits a spike), resets its membrane potential to 0 and enters a refractory period where it does not integrate incoming spikes for one time step. After the integration phase and if the neuron has not fired, leakage is applied by decrementing membrane potential by 1 (or incrementing it if the potential is below 0), so that the potential tends to 0.

5 Pattern recognition experiment

A circuit of 8x8 cells (fig. 5) was evolved to recognize characters (any other pattern could be used) using two training sets: one set contains corrupted versions of two characters, the other set contains random patterns. The circuit must indicate whether the current pattern is one of both characters or not. The fitness of the network is evaluated by presenting successively all the patterns of a training set, and it is equal to the number of times it correctly classifies the input pattern. Fig. 5 shows the training set for the recognition of characters A and C (noted as A+C). The upper line shows the subset of patterns to recognize. The second line contains random patterns that must be rejected. The maximum fitness achievable is 20 (20 patterns in the training set). The experiments have been performed with four different training sets, for the recognition of characters A+B, A+C, A+D and A+E.

Figure 5 shows the array of cells in the circuit. The input pattern is applied on a subset of cells through the external input of the cells. Each cell receives one pixel of the pattern: if the pixel is black, then it receives a spike every two time steps, otherwise it receives no spike. The network is run for 100 time steps with the input applied to it, after which the activity of the output neuron on the right of the network is read. The activity level (number of spikes) of that neuron indicates whether the pattern is recognized or not (threshold=50% of maximum spike number).

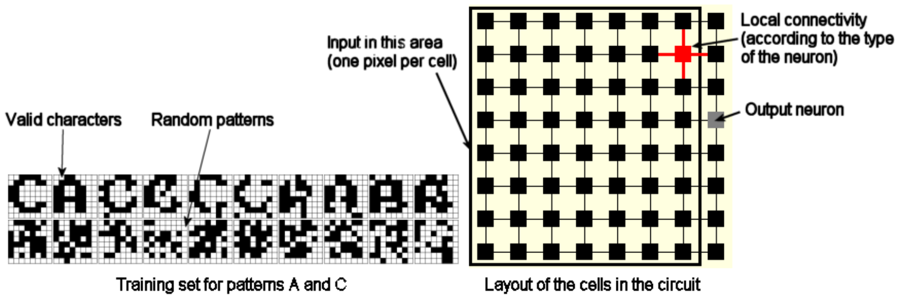

Valid characters
Random patterns
Training set for patterns A and C
Input in this area (one pixel per cell)
Local connectivity (according to the type of the neuron)
Output neuron
Layout of the cells in the circuit

Fig. 5. Left: training set for the recognition of the patterns A and C. It is composed of 20 patterns in two subsets of 10 patterns. The upper line contains the patterns to recognize which are the letters A and C. The second line contains the random patterns that the network must reject. Right: layout of the cells in the circuit. The input pattern is applied on an array of 7x8 cells on the left, with each cell receiving the input from one pixel of the pattern. The output of a single cell at the right column is used to indicate whether the character has been recognized or not.

A morphogenetic system with 16 diffusers and 12 entries in the expression table (one for each type of neuron) is compared to the direct coding. The size of the morphogenetic coding is 320 bits: 12 entries in the expression table * 16 bits + 16 diffusers * 8 bits (6 bits for the coordinates and 2 bits for the type of the diffuser). The size of the direct coding is 256 bits (12 type of neurons, thus 4 bits/cells * 64 cells).

The circuit has been evolved one hundred times for each of the four training sets (fig. 6). The morphogenetic system outperforms the direct coding, both when comparing the maximum fitness reached at a given generation and when comparing the number of runs that have reached the maximum fitness after 50 generations. Table 1 reports the number of runs (on the 100 runs performed) where the maximum fitness of 20 is reached. Averaged over the four training sets, runs finding a maximum fitness using the morphogenetic coding are more than twice as frequent than when using the direct coding.

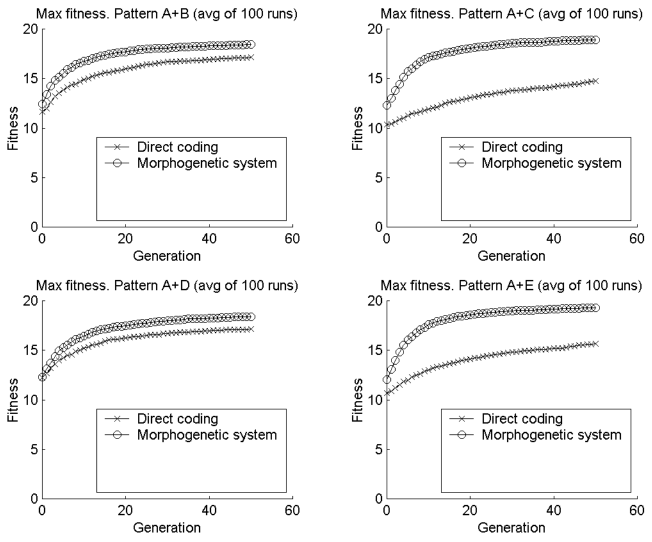


Fig. 6. Evolution of maximum fitness for pattern recognition with the four training sets.

	8x8 network	
Training set	Direct	Morph.
A+B	18	27
A+C	8	34
A+D	19	20
A+E	14	54
Total (max is 400):	59	135

Table 1. Number of runs, out of 100 performed for each training set, reaching the maximum fitness.

6 Mobile Robot Controller

A spiking circuit is evolved as a controller for a Khepera miniature autonomous robot. The objective is perform obstacle avoidance using the sensory informations coming from the proximity sensors of the robot (fig. 7). Four sensory groups of two cells are connected to the infrared sensors. The activity of two other cells are used to set the speeds of the wheels.

The robot has a sensory-motor period of 100ms. During that period, the neurons on the circuit are updated 20 times and, according to the distance to the obstacles, either 0, 1 (the "low" activity), or both input neurons ("low" and "high" activity) of each sensory group receive a spike train of period 2 (one spike every two time steps). At the end of the sensory-motor period, the speeds of the wheels are updated and the proximity sensors are read to compute spike trains for the next sensory-motor cycle. Some amount of noise is introduced in the spike trains. The cells ML and MR control the speed of the wheels in a way which is inversely proportional to their activity. This allows the robot to move forward when no obstacles are sensed and thus when there is potentially no activity in the network. A minimum activity of the neuron sets the speed of the wheel to +80 mm/s. A maximum activity sets the speed to -80 mm/s. The speed of the wheels scales linearly in between.

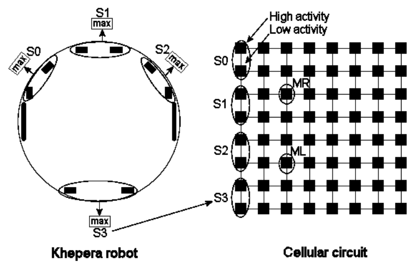


Fig. 7. The Khepera robot (left) and the neural controller (right). The Khepera has 8 proximity sensors. They are grouped by two, taking value of the most active sensors, to have 4 sensory inputs S1 to S4. The circuit receives S1 to S4 as sensory inputs. Each input is coded on two neurons. The cells ML and MR control the speed of the wheels according to their activity.

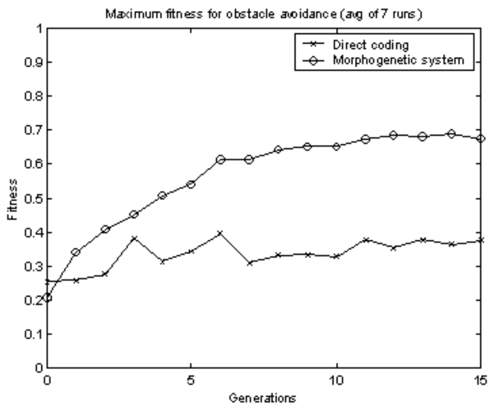


Fig. 8. Evolution of the maximum fitness in the obstacle avoidance experiment (average of 7 runs on a physical robot).

The spiking neuron model used here is the same as used in the pattern recognition experiment (section 4), with the addition of random variations in the threshold value to avoid locked oscillations during the 100 ms sensory-motor period. For each neuron and at each time step the threshold value is incremented or decremented by 1 with a probability of 5%.

The fitness of the robot is measured on two tests of 30 seconds in a rectangular arena (40x65 cm). It is the average of the fitness computed at each sensory-motor step using the following equation [4]: $f = \overline{v} \cdot (1 - \Delta v) \cdot (1 - p)$, where $\overline{v}$ is the average speed of the two motors, Δv is the absolute value of the difference of speed of the motors, and p is the activity of the most active sensor ($\overline{v}$, Δv and p are in the range [0;1]). The three parts of this function aim to 1) maximize the speed of the robot, 2) minimize the rotation and 3) maximize the distance to the obstacles.

Seven runs were performed to compare the same morphogenetic system and direct coding as those used in the pattern recognition experiments. The morphogenetic system performs better than the direct coding (fig. 8): it clearly achieves a higher maximum fitness than the direct coding. Moreover, after 15 generations, only three runs managed to find individuals displaying obstacle avoidance behavior with direct coding, whereas with the morphogenetic system individuals were found displaying this behavior in all seven runs. It is thought that the difference in performance may come from some characteristics that individuals generated with the morphogenetic system possess but which are harder to get with the direct coding. A preliminary analysis is discussed below.

7 Analysis

The better performance of the morphogenetic system over the direct genetic encoding may be due to differences in characteristics of the fitness landscape. Ruggedness is often linked to the difficulty of search when genetic algorithms are used [8]. To investigate the level of ruggedness, random walks using the mutation operator have been performed in the pattern recognition experiment. For 600 individuals of maximum fitness, 5 random walks of 10 steps (10 consecutive applications of the mutation operator) have been performed on circuits evolved using the morphogenetic system and the direct genetic encoding. The fitness averaged on the 3000 random walks is plotted against the number of application of the mutation operator in fig 9. The fitness drops faster with the morphogenetic system when moving away from a point of maximum fitness, which may imply a more rugged fitness landscape. The better performance of the morphogenetic system thus cannot be explained by having a smoother fitness landscape.

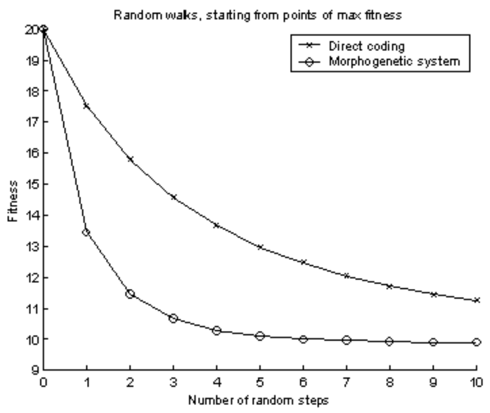


Fig. 9. Comparison of the random walks. The fitness drops faster with the morphogenetic system when going farther away from points of maximum fitness. This seems to indicate that the fitness landscape is more rugged with the morphogenetic system.

	Direct		Morph.	
Training set	Max	Std dev	Max	Std dev
A+B	11.7	0.77	11.92	0.88
A+C	10.38	0.74	12.14	1.08
A+D	12.22	0.64	12.1	0.85
A+E	10.58	0.95	12.28	1.03

Table 2. Maximum fitness and standard deviation of the fitness in a sample of 5000 randomly generated individuals.

We also measured the fitness values of 5000 randomly generated individuals using the morphogenetic system and the direct genetic encoding. The statistics of table 2 for the pattern recognition task show that the morphogenetic system tends to display higher values and higher standard deviation of the fitness distribution. Although the differences are small and may not be significative, the higher fitness variability may explain why the morphogenetic system can generate individuals of the same fitness faster than direct genetic encoding. The reason why morphogenetic systems also generate individuals with higher fitness is still not clear. We speculate that the difference can be explained by structural properties of the circuits generated by the two genetic encodings. For example, in the case of the robotic experiments, we noticed that the morphogenetic system generated very easily large patches of interconnected excitatory neurons that link sensors to motor neurons, causing a reversal of wheel speeds when the robot approaches an obstacle. Instead, direct genetic encodings take more time to set all the redundant cells to suitable values. However, these data are only preliminary require further statistical investigations.

8 Conclusions

We have presented a simple morphogenetic system and compared it to a direct genotype-to-phenotype mapping. The morphogenetic system uses a simple signalling and expression mechanism, which is inspired by the mechanisms of gene expression and cell differentiation of living beings. Several experiments were performed and the morphogenetic system outperformed a direct coding, both when comparing the maximum fitness reached at a given generation and when comparing the number of runs reaching the maximum fitness. The morphogenetic system, designed to be simple (hardware implementation), and its good performance make it well suited for use in the POEtic circuit. Further experiments need to be done to see how the morphogenetic system scales to networks of larger size. Also, further analysis are needed to explain the better performance of the morphogenetic system. As the mechanism is inspired upon biology, it may be that knowledge from that field will help to shed some light on this topic.

However, the real strength of the morphogenetic system is yet to be realized. It resides in its capacity to handle run-time dynamically reconfigurable circuits, e.g. by adding or removing diffusers, or by expressing the functionality of newly inserted cells in the circuit. Those aspects must still be explored, however they may pave the way toward dynamically reconfigurable electronic circuits that reorganize when a sensor fails or when new sensors or actuators are added to the system. In the future we also plan to extend the system by allowing environmental interactions while the circuit evolves, as well as including the evolution of functions that each cell can take, which, for the sake of simplicity, here have been limited to a predefined set.

9 Acknowledgments

The authors thank Jesper Blynel, Eduardo Sanchez, Gianluca Tempesti and Yann Thoma for their ideas and comments. This project is funded by the Future and Emerging Technologies programme (IST-FET) of the European Community, under grant

IST-2000-28027 (POETIC). The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication. The Swiss participants to this project are funded by the Swiss government grant 00.0529-1.

References

1. S. Boshy and E. Ruppin. Small is beautiful: Near minimal evolutionary neurocontrolllers obtained with self-organizing compressed encoding. In B. Hallam, D. Floreano, J. Hallam, G. Hayes, and J.-A. Meyer, editors, *Proceedings of the Seventh International Conference on Simulation of Adaptive Behaviour*, pages 345–346, Cambridge, MA, 2002. MIT Press-Bradford Books.
2. E. Coen. *The art of genes*. Oxford University Press, New York, 1999.
3. P. Eggenberger. Cell interactions as a control tool of developmental processes for evolutionary robotics. In P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson, editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 440–448, Cambridge, MA, 1996. MIT Press-Bradford Books.
4. D. Floreano and F. Mondada. Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, editors, *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 421–430, Cambridge, MA, 1994. MIT Press-Bradford Books.
5. D. Floreano, N. Schoeni, G. Caprari, and J. Blynel. Evolutionary bits'n'spikes. In *Artificial Life VIII Proceedings*. MIT Press, 2002.
6. F. Gruau. Automatic definition of modular neural networks. *Adaptive Behavior*, 3:151–183, 1994.
7. P. C. Haddow, G. Tufte, and P. van Remortel. Shrinking the Genotype: L-systems for EHW? In Y. Liu, K. Tanaka, M. Iwata, T. Higuchi, and M. Yasunaga, editors, *Evolvable Systems: From Biology to Hardware; Proceedings of the Fourth International Conference on Evolvable Systems (ICES 2001)*, pages 128–139, Berlin, 2001. Springer.
8. T. Smith, P. Husbands, and M. O'Shea. Not measuring evolvability: Initial investigation of an evolutionary robotics search space. In *Congress on Evolutionary Computation 2001*, pages 9–16. IEEE Press, 2001.
9. G. Tempesti, D. Mange, A. Stauffer, and C. Teuscher. The biowall: An electronic tissue for prototyping bio-inspired systems. In A. Stoica, J. Lohn, R. Katz, D. Keymeulen, and R. S. Zebulum, editors, *Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware*, pages 221–230. IEEE Computer Society, Los Alamitos, CA, 2002.
10. A. M. Tyrrell, E. Sanchez, D. Floreano, G. Tempesti, D. Mange, J.-M. Moreno, J. Rosenberg, and A. Villa. POEtic Tissue: An Integrated Architecture for Bio-Inspired Hardware. In *Evolvable Systems: From Biology to Hardware; Proceedings of the Fifth International Conference on Evolvable Systems (ICES 2003)*, Berlin, 2003. Springer.
11. X. Yao. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 4:203–222, 1993.