Neural Morphogenesis, Synaptic Plasticity, and Evolution

Dario Floreano and Joseba Urzelai
Evolutionary and Adaptive Systems Team
Institute of Robotic Systems
Swiss Federal Institute of Technology, Lausanne (EPFL)
http://dmtwww.epfl.ch/isr/east

Abstract

Morphology plays an important role in the computational properties of neural systems, affecting both their functionality and the way in which this functionality is developed during life. In computer-based models of neural networks, artificial evolution is often used as a method to explore the space of suitable morphologies. In this paper we critically review the most common methods used to evolve neural morphologies and argue that a more effective, and possibly biologically plausible, method consists of genetically encoding rules of synaptic plasticity along with rules of neural morphogenesis. Some preliminary experiments with autonomous robots are described in order to show the feasibility and advantages of the approach.

1 Introduction

Consider a generic network made of nodes and links, such as the internet, an electricity grid, or the brain. Two such networks with the same number of nodes and links, but with different architectures, can display quite different behaviors (Strogatz, 2001).

The role of network architecture, or morphology, becomes even more important when its components are adaptive, such as in biological and artificial neural networks where synaptic connections and node thresholds can change their strength according to the activity of the units they are connected to. In this case the morphology affects both *what* functionality is acquired by the network during the process of synaptic change and *how* the network behaves after the process of change.

Despite the important role of morphology in artificial neural networks, there is a lack of guiding principles for choosing suitable architectures. Most computational models taking into account both synaptic and morphological change can be divided in two types: construction and destruction algorithms. Construction algorithms start with a simple neural architecture and gradually add internal nodes while training additional connection strengths so to maximize the objective function describing the desired network behavior (Grossberg, 1987; Fahlman & Lebiere, 1990, e.g.). Destruction algorithms instead start with large redun-

dant networks and gradually delete connections (and nodes) whose strength remains or goes very close to zero under the pressure of some penalty term in the learning rule (Kramer & Sangiovanni-Vincentelli, 1989, e.g.).

These models preserve general morphological properties, such as type of connectivity and node functionality, and often generate optimized but homogeneous architectures. In other words, they hardly allow the emergence of new structures and modules within the same network. Modularity and structural variation is a property of most biological neural circuits and can be quite useful also in artificial neural networks to prevent cross-talk and other types of interferences among nodes.

More recently, some researchers have attempted to develop network morphologies using evolutionary algorithms whereby architecture and component types are encoded in artificial chromosomes and subjected to the process of selective reproduction, crossover, and random mutation. This methodology has produced morphologies that display both modularity and structural variability. However, the networks evolved so far often do not show better functionality than simple hand-designed neural networks (Siddiqi & Lucas, 1998).

In this paper we describe a method for evolving adaptive neural morphologies that show promising scalability and evolvability. The core of the methodology consists of co-evolving the rules of synaptic plasticity along with the specification of the network morphology. This strategy does not require genetic specification of synaptic strengths because the suitable values are developed by the plasticity rules and therefore results in quite compact genetic encoding. Furthermore, instead of relying on off-the-shelf learning algorithms, which introduce several constraints, this method discovers new "learning structures" by assigning different plasticity rules to various parts of the network. To support our claims, we describe some preliminary experiments carried out with an autonomous mobile robot where we genetically encode rules of morphology growth and of synaptic plasticity.

2   Evolution of Neural Morphologies

As we have mentioned in the previous section, encoding every detail of the architecture may generate extremely long genotypes, which are difficult to evolve. Therefore, typically only a few construction parameters are encoded to generate more or less complex network architectures from one-dimensional genotypes through a "development" mechanism. This process is referred to as *development* or *morphogenesis.*

Various development approaches to the construction of neural architectures have been proposed. Kitano (1990) for example, employed a developmental encoding based on a set of rewriting rules encoded on the genotype. The genotype is divided in blocks of five elements. Each block of five is interpreted as a rewriting rule that determines how the first symbol is developed in a matrix containing the other four symbols of the block. There are two types of symbols: terminals and non-terminals. A terminal symbol develops in a predetermined 2x2 matrix of 0's and 1's. A non-terminal symbol develops in a 2x2 matrix of symbols. The first block of the genotype builds the initial 2x2 matrix of symbols, each of which recursively develops using the rules encoded in the genotype until a matrix of 0's and 1's is built. This matrix represents the architecture

and connection pattern of the network.

In Kitano's approach the values of the synaptic strengths are tuned through a supervised learning algorithm known as Back Propagation of Error (Rumelhart, Hinton, & Williams, 1986) based on the difference between the desired output and the output of the network for every possible input pattern. This method is not applicable for artificial brains of autonomous systems that operate without external supervision.

In an original research work, Gruau (1992, 1994) proposed a genetic encoding method scheme for neural networks based on a cellular duplication and differentiation process. Gruau employed *Genetic Programming* to encode and evolve a set of instructions that rule cellular division and the generation of connections. The process starts with a single cell that undergoes a number of duplication and transformation operations ending up in a complete neural network. In order to solve the problem of defining synaptic strength, Gruau defined a library of possible neurons each with a specific pattern of synaptic connectivity. He also considered the possibility of generating repeated structures by using hierarchical tree representations where a node can point to other trees. Gruau defined this method as *Automatic Definition of Neural Subnetworks (ADNS)* (Gruau, 1994).He applied this method to evolve neural controllers for a simulated hexapod robot and compared the results obtained with and without ADNS. The genotype of evolved individuals was significantly more compact in the case of experiments with ADNS. Moreover, ADNS provided more structured phenotypical networks. A similar approach was proposed later by Kodjabachian and Meyer (1998).

Another strategy used in (Nolfi, Miglino, & Parisi, 1994; Husbands, 1994; Cliff & Miller, 1996) consists of encoding the properties and the position of the neurons, and the parameters controlling the synaptic growth process in a two-dimensional coordinate system. Networks are built up by placing neurons at specific positions of a 2D "brain surface", and then starting a synaptic growth process that sets up the connections of the network. The approach by Nolfi et al. (1994) includes in addition a maturation process, that is some parameters of the synaptic growth process are a function of the neuronal activity of the network, and consequently the interactions between the network and its environment determine the final structure of the architecture. This model was extended in (Cangelosi, Parisi, & Nolfi, 1994) by adding a cell division and migration process to the existing synaptic growth.

Other authors (Jakobi, 1995; Michel, 1997; Eggenberger, 1996, e.g.) have proposed more biologically-inspired approaches of genetically controlled cellular division to generate neural network architectures.

Despite the interesting mechanisms involved in most of these methods, their potential ability to generate neural networks capable of complex behaviors has not yet shown up. Indeed, all results obtained using these methods and reported in the literature so far are not more complex than those obtained with simple, reactive, and hand-designed neural networks (obstacle avoidance and light following). Also, not all methodologies seem to scale up well when it comes to evolve large neural morphologies (Gruau & Quatramaran, 1997; Cliff & Miller, 1996, e.g.).

We think that these problems are due to the fact that the genetic string must encode both the architecture and the strengths of the connections in the network. This has two consequences. On the one hand, the spatial development

and recombination of neurons and modules induced by the rules of growth and by the genetic operators may not always result in functional networks. On the other hand, the length of the genetic string grows with the complexity of the networks that can be expressed, but the number of *viable networks* that can be expressed by the genetic encoding for a given functionality does not grow at the same rate. In other words, both factors contribute to the fact that the search space becomes larger while the number of good solutions remains small.

3 Evolution of Morphogenetic Plastic Networks

In this paper, we suggest to co-evolve the rules of synaptic adaptation along with the rules of neural morphogenesis. The rules of synaptic adaptation are variations of the Hebb rule whereby the strength of a synapse changes according to the correlated activities of the presynaptic neuron and of the postsynaptic neuron. The genetic string encodes both the rules by which a neural network develops in space and time and the rules by which the synaptic connections vary their strength while the organism interacts with the environment. There are at least three advantages in doing so.

- plastic synapses could provide a flexible interface between modules combined by the process of morphogenesis and genetic crossover by adapting at run time their strengths to accomodate the functionality of connected modules. This would provide a higher number of functional networks for a given genetic encoding, resulting in smoother fitness landscapes, and consequently improve evolvability of the system;

- it is not necessary to resort to off-the-shelf learning algorithms to fine tune the synaptic strengths, such as Back Propagation of Error (Rumelhart et al., 1986), because their values are adjusted by the evolved choice of hebbian rules which operate using only local information. This makes the approach applicable to a larger number of situations, which are not constrained by a given learning paradigm (supervised, e.g.) or architecture required by conventional learning algorithms.

- it is no longer necessary to encode synaptic strengths because their value is determined "at run time" by the evolved plasticity rules. If the genetic specification for a neuron encodes also the type of hebbian plasticity that all incoming synapses use, the length of the genetic string grows only with the number of neurons in the network, not with the number of all possible synapses. Furthermore, if a neuron-based encoding is combined with some morphogenetic rule, the length of the genetic string could in theory be scale invariant.

In this article we focus on the first two aspects of the list above. Part of the third aspect (neuron-based encoding, without morphogenesis) has already been described elsewhere (Floreano & Urzelai, 2000) and will be mentioned in the discussion.

In order to do so, we have used a developmental approach to the construction of modular neural networks inspired upon the *matrix rewriting scheme* proposed by Kitano (1990) and briefly described in the previous section. We intend to show that:
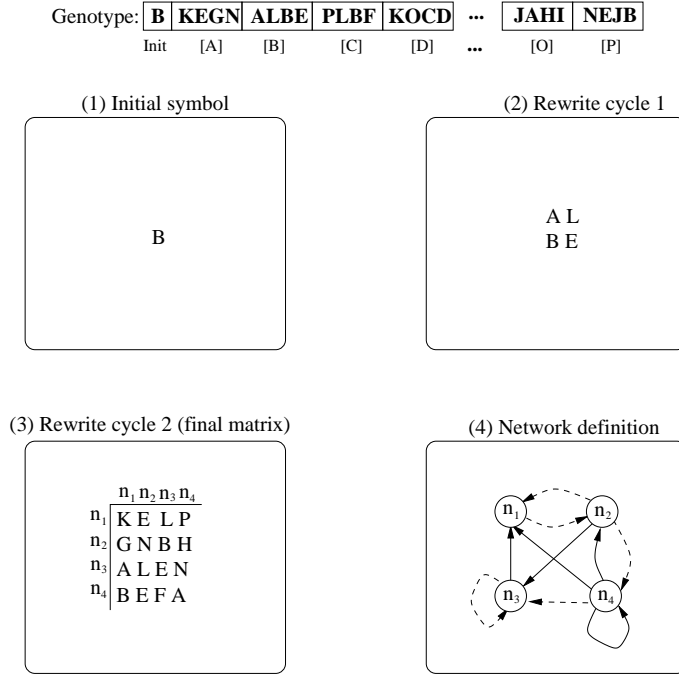
4

Genotype: | **B** | **KEGN** | **ALBE** | **PLBF** | **KOCD** | ··· | **JAHI** | **NEJB** |

Init   [A]   [B]   [C]   [D]   ···   [O]   [P]

(1) Initial symbol

B

(2) Rewrite cycle 1

A L
B E

(3) Rewrite cycle 2 (final matrix)

$$
\begin{array}{c|cccc}
 & n_1 & n_2 & n_3 & n_4 \\
\hline
n_1 & K & E & L & P \\
n_2 & G & N & B & H \\
n_3 & A & L & E & N \\
n_4 & B & E & F & A \\
\end{array}
$$

(4) Network definition

Figure 1: Encoding of development instructions for the evolution of neural morphologies. The genotype (top) encodes for 16 blocks of 4 symbols each. Each block corresponds to a symbol of the alphabet $\mathcal{A} = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P\}$ and determines how the symbol develops in a 2x2 matrix. The initial symbol of the development process is encoded in the first position of the genotype. Matrix rewriting begins with a 1x1 matrix containing the initial symbol (1), which then undergoes p parallel rewriting cycles (only two rewriting cycles, 2 and 3, in this example). A rewriting cycle consists in developing each symbol of the matrix in a 2x2 submatrix. After the final rewriting cycle (3), the resulting matrix is interpreted as a neural network (4). Each symbol of the final matrix determines whether the neuron n_i is connected to the neuron n_j in the resulting neural network, as well as the *sign* and the *properties* of the connection (see table 1). Positive synapses are represented as solid lines and negative synapses as dashed lines.

5

1. encoding the hebbian rules of synaptic plasticity makes the developing network more evolvable and generates better solutions than encoding the synaptic strengths;

2. the method can be used to develop the neural controller of an autonomous robot, where by definition one cannot use a supervised learning algorithm to fine-tune the synaptic connections.

Our particular implementation of the matrix rewriting algorithm is based on an alphabet of 16 symbols:

$$\mathcal{A} = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P\}$$

During the process of morphogenesis, such symbols determine how the network develops in space and time; at the end of the growth process, they specify the arrangement and properties of the synaptic connnections among the neurons of the resulting network.

The genotype encodes for 16 blocks of 4 symbols each. Each block corresponds to a symbol of the alphabet $\mathcal{A}$ and determines how the symbol develops in a 2x2 matrix. The initial symbol of the development process is encoded in the first position of the genotype (figure 1, top). Therefore, each genotype encodes for $1 + 4 * 16 = 65$ symbols.

Matrix rewriting begins with a 1x1 matrix containing the initial symbol encoded on the genotype (figure 1, 1), which then undergoes p parallel rewriting cycles (only two rewriting cycles, 2 and 3, in the example of figure 1). Such a rewriting cycle consists in developing each symbol of the current matrix in a 2x2 submatrix (e.g. $[B] \longrightarrow [ALBE]$ is the first rewriting cycle in figure 1). Hence for a fixed number of p rewriting cycles, a given development process always produces a 2^px2^p matrix.

After the final rewriting cycle (figure 1, 3), the resulting matrix is interpreted as a neural network (figure 1, 4). Each symbol of the final matrix determines whether the neuron n_i is connected to the neuron n_j in the resulting neural network, as well as the *sign* and the *properties* of the connection. Two types of properties can be specified depending on the nature of the synaptic connections. In the case of adaptive synapses, the symbol specifies the learning rule used to adapt the synapse, which is randomly initialized at the beginning of individual's life. Instead, in the case of genetically-determined synapses, the symbol specifies the strength of the synaptic connection and this value is constant during individual's life (table 1).

The adaptation rules specified here by the symbols are *plain Hebbian, presynaptic*, *postsynpatic*, and *covariance* rules. Adaptive synapses are randomly initialized at the beginning of individual's life and are updated every 100 ms using these adaptation rules. These rules of Hebbian plasticity capture some of the most common mechanisms of local synaptic adaptation found in the nervous systems of mammalians (Willshaw & Dayan, 1990). The following mathematical constraints were added. Synaptic strength could not grow indefinitely, but was kept in the range $[0, 1]$ by means of a self-limiting mechanism which depended on synaptic strength. Because of this self-limiting factor, a synapse could not change sign, which was genetically specified, but only strength. Each synaptic weight w_{ij} is randomly initialized at the beginning of the individual's life and is updated after every sensory-motor cycle (100 ms),

Symbol	Connected?	Sign	Rule	Strength
A	yes	+	Plain Hebb	0.25
B	yes	+	Presynaptic	0.50
C	yes	+	Postsynaptic	0.75
D	yes	+	Covariance	1.00
E	yes	-	Plain Hebb	0.25
F	yes	-	Presynaptic	0.50
G	yes	-	Postsynaptic	0.75
H	yes	-	Covariance	1.00
I	no			
J	no			
K	no			
L	no			
M	no			
N	no			
O	no			
P	no			

Table 1: Symbol encoding in the matrix rewriting scheme. Each symbol specifies the existence, the sign, and the properties of a connection. In the case of adaptive synapses, the symbol specifies the learning rule used to adapt the synapse, whose intial strength is randomly set at the beginning of an individual's life. Instead, in the case of genetically-determined synapses, the symbol specifies the strength of the synaptic connection and this value is constant during an individual's life.

$$w_{ij}^t = w_{ij}^{t-1} + \eta \Delta w_{ij},$$

where $0.0 < \eta < 1.0$ is the learning rate and Δw_{ij} is one of the four modification rules specified in the genotype:[1]

1. *Plain Hebb rule*: can only strengthen the synapse proportionally to the correlated activity of the pre- and post-synaptic neurons.

$$\Delta w = (1 - w) \, xy \tag{1}$$

2. *Postsynaptic rule*: behaves as the plain Hebb rule, but in addition it weakens the synapse when the postsynaptic node is active but the presynaptic is not.

$$\Delta w = w \, (-1 + x) \, y + (1 - w) \, xy \tag{2}$$

3. *Presynaptic rule*: weakening occurs when the presynaptic unit is active but the postsynaptic is not.

$$\Delta w = wx \, (-1 + y) + (1 - w) \, xy \tag{3}$$

4. *Covariance rule*: strengthens the synapse whenever the difference between the activations of the two neurons is less than half their maximum activity,

---

[1] These four rules co-exist within the same network.

7

Figure 2: A mobile robot Khepera equipped with a vision module gains fitness by staying on the gray area only when the light is on. The light is normally off, but it can be switched on if the robot passes over the black area positioned on the other side of the arena. The robot can detect ambient light and the color of the wall, but not the color of the floor.

otherwise the synapse is weakened. In other words, this rule makes the synapse stronger when the two neurons have similar activity and makes it weaker otherwise.

$$\Delta w = \begin{cases} (1-w)\mathcal{F}(x,y) & \text{if } \mathcal{F}(x,y) > 0 \\ (w)\mathcal{F}(x,y) & \text{otherwise} \end{cases} \qquad (4)$$

where $\mathcal{F}(x,y) = \tanh(4(1-|x-y|)-2)$ is a measure of the difference between the presynaptic and postsynaptic activity. $\mathcal{F}(x,y) > 0$ if the difference is bigger or equal to 0.5 (half the maximum node activation) and $\mathcal{F}(x,y) < 0$ if the difference is smaller than 0.5.

3.1   Robotics Experiments

In this section we describe a set of experiments on evolution of network morphologies with synaptic plasticity for an autonomous robot. A mobile robot Khepera equipped with a vision module is positioned in the rectangular environment shown in figure 2. A light bulb is attached on one side of the environment. This light is normally off, but it can be switched on when the robot passes over a black-painted area on the opposite side of the environment. A black stripe is painted on the wall over the light-switch area.

Each chromosome of the population is decoded into a corresponding network morphology whose input and output neurons are connected to the robot through a serial connection with rotating contacts (to prevent twisting of cables) and tested for 500 sensory motor cycles, each cycle lasting 100 ms. At the beginning of an individual's life, the robot is positioned at a random position and orientation and the light bulb is off.

The fitness function is given by the number of sensory motor cycles spent by the robot on the gray area beneath the light bulb *when the light is on* divided by the total number of cycles available (500). In order to maximize this fitness function, the robot should find the light-switch area, go there in order to switch
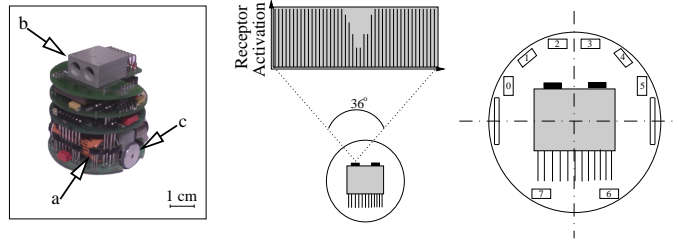
8

Figure 3: The Khepera robot used in the experiments. Infrared sensors (a) measure object proximity and light intensity. The linear vision module (b) is composed of 64 photoreceptors covering a visual field of 36° (center). The output of the controller generates the motor commands (c) for the robot. Right figure shows the sensory disposition of the Khepera robot.

the light on, and then move towards the light as soon as possible, and stand on the gray area. Since this sequence of actions takes time (several sensory motor cycles), the fitness of a robot will never be 1.0. Also, a robot that cannot manage to complete the entire sequence will be scored with 0.0 fitness. A light sensor placed under the robot is used to detect the color of the floor—white, gray, or black— and passed to a host computer in order to switch on the light bulb and compute fitness values. The output of this sensor is *not* given as input to the neural controller. After 500 sensory motor cycles, the light is switched off and the robot is repositioned by applying random speeds to the wheels for 5 seconds. Notice that the fitness function does not explicitly reward this sequence of actions (which is based on our external perspective), but only the final outcome of the sequence of behaviors chosen by the robot.

The neural network is composed of a number of McCulloch-Pitts neurons with sigmoid activation function. The number of neurons in the network and their connectivity pattern is determined by the matrix rewriting approach defined above. In these experiments we have limited the number of rewriting cycles to four, which generate networks of sixteen neurons. Ten of these neurons can receive input from the following sensors of the robot (figure 3):

1. *Infrared light*: the active infrared sensors positioned around the robot (figure 3, a) measure the distance from objects. Their values are pooled into four pairs and the average reading of each pair is passed to a corresponding neuron.

2. *Ambient light*: the same sensors are used to measure ambient light too. These readings are pooled into three groups and the average values are passed to the corresponding three light neurons.

3. *Vision*: the vision module (figure 3, b) consists of an array of 64 photoreceptors covering a visual field of 36° (figure 3, center). The visual field is divided up in three sectors and the average value of the photoreceptors (256 gray levels) within each sector is passed to the corresponding vision neuron.

The activations of two other neurons are read to set the rotation speeds of the wheels (figure 3, c). The remaining four neurons are not directly connected
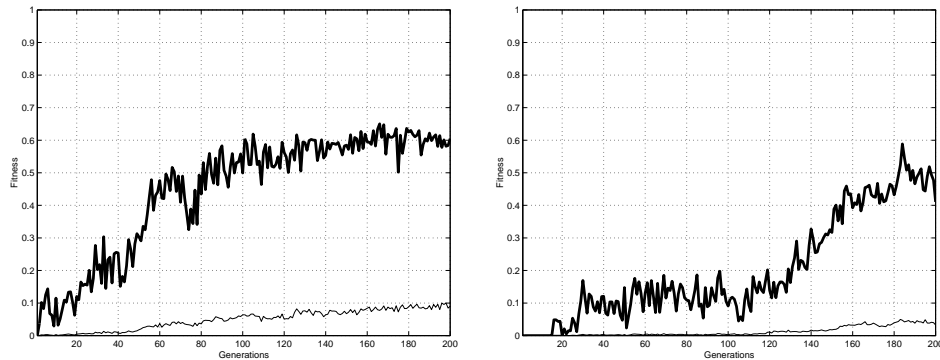
Figure 4: Comparison of different types of encoding for the evolution of matrix-rewriting instructions in the "light-switching" problem. **Left**: Adaptive synapses with Node Encoding. **Right**: Genetically-determined synapses with Synapse Encoding. Thick line=best individual; thin line=population average. Each data point is an average over 10 replications with different random initializations. Population size is 100 and 20 best individuals reproduce by making 5 copies. Crossover probability is 0.2 and mutation probability is 0.05 (per bit).

to sensors and motors. Every 100 ms new sensory values are passed to the input neurons, all neurons in the network are updated, and the rotation speeds of the robot wheels are set for the next 100 ms.

A set of experiments has been carried out aiming at comparing adaptive synapses to genetically-determined synapses on the matrix rewriting approach for the "light-switching" environment described above. Every replication is composed of a population of 100 individuals, which is evolved for 200 generations by reproducing 20 best individuals (5 copies each). Each individual is tested three times, crossover probability is 0.2, and mutation probability is 0.05 (per bit).

The experimental results (figure 4) indicate that evolution of morphogenetic networks with genetically-determined synapses is more difficult because fitness score remains very low for more than 100 generations, and the final performance is lower than in the case of evolution of morphogenetic networks with plastic synapses.

Performance difference is more evident if we look at the behaviors generated by evolved networks. Behaviors generated by networks with adaptive synapses (figure 5, left column) are very efficient in finding the light-switching area and in following the light, and they remain on the fitness area once that they reach it. On the contrary, networks evolved with genetically-determined synapses (figure 5, right column) display, in general, minimalist behaviors. The individual showed at the top row, for example, performs a wall-following behavior whereby it runs over the light-switching area and stops when it is close to the light. Although this individual accomplishes the task, it is not very efficient since its performance is highly determined by the initial position of the robot[2].

---

[2]Since the robot performs a wall-following behavior, it needs to perform a complete tour

f = 0.560, <f>^{10} = 0.458   f = 0.292, <f>^{10} = 0.312

f = 0.496, <f>^{10} = 0.434   f = 0.000, <f>^{10} = 0.116
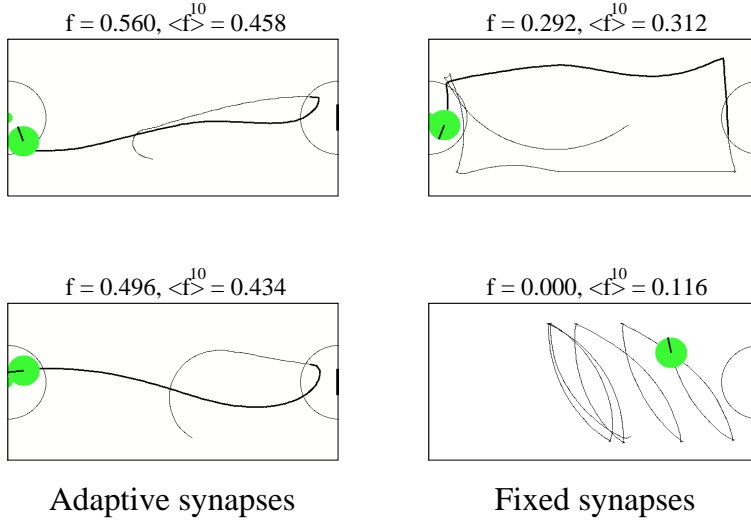
Adaptive synapses   Fixed synapses

Figure 5: Comparison of behavior of best individual of the last generation for the evolution of matrix-rewriting instructions in the "light-switching" problem (two different replications are shown). **Left column**: Adaptive synapses with Node Encoding. **Right column**: Genetically-determined synapses with Synapse Encoding. When the light is turned on, the trajectory line becomes thick. The corresponding fitness value is printed on the top of each box along with the average fitness of the same individual tested ten times from different positions and orientations.

Moreover, a change in the shape of the environment, or in the position of the light-switching area or the fitness area would have fatal consequences for this individual. Clearly, the strategy evolved by adaptive controllers is more performant, since they have developed strong attraction towards the light-switching area and towards the fitness area once the light is switched on. In addition, the performance displayed by some genetically-determined individuals is very dependent on the initial position of the robot. The individual showed at the bottom row, for example, performs loopy trajectories that are not efficient in switching the light on if the initial position of the robot is relatively far from the light-switching area.

4 Discussion

The experiments described in this paper show that co-evolution of synaptic plasticity rules along with morphogenesis rules provide at least two advantages with respect to evolution of morphogenesis rules alone (where synaptic strength is genetically determined): a) better solutions in terms of fitness values and behavioral abilities, and b) better evolvability.

The better quality of evolved neural networks can be explained by the fact

---

of the environment before getting to switch the light on if its initial position is close and its orientation is opposite to the light-switching area.

that forcing the networks to adapt their synaptic strengths at run time starting from random values, instead of relying on evolved synaptic strengths, selects individuals for their ability to develop at run time and very quickly the competencies required by the environment while interacting with it. This makes the evolved individuals much more robust to situations that have not been experienced during evolution (Floreano & Urzelai, 2000). Also, it prevents evolution from genetically encoding behavioral abilities in the properties of the network. Instead, when synaptic strengths are genetically encoded and evolved without ontogenetic plasticity, artificial evolution often discovers viable, but sub-optimal solutions, such as the wall-following behavior displayed above. This seems to be the case also when the architecture of the network is predefined; in addition, some experiments have shown that when an extra gene is added for each neuron in the network to decide whether to take on predefined synaptic strengths or rules of synaptic plasticity, most neurons express the latter option (Floreano & Urzelai, 2000).

The better evolvability of plastic morphogenetic networks could be explained by the fact that synaptic connections have the flexibility to develop at run time the range of values that are most suitable for the developing architecture. Instead, in the case of genetically-encoded synaptic strengths, evolution has the harder task of finding at the same time both a viable architecture and a set of synaptic strengths that matches the architecture. Plastic synapses may also better withstand and possibly exploit the effects of genetic recombination by adapting their values in order to accomodate sub-modules inherited from other individuals. At this stage, this is only a speculation, but it could be easily tested by running experiments with various crossover probabilities and performing genetic analysis to discover whether neural building blocks are preserved and propagated through generations among the best individuals.

Figure 6 shows an example of an evolved neural network whose genotype specifies both morphology and plasticity rules (it corresponds to the individual displayed on the top left corner of figure 5). For sake of clarity, the network has been displayed with the same format used in the third frame of figure 1. This network displays a high degree of replication and modularity. The first four rows of synapses are repeated at rows 9 to 12. Also, blocks of four columns are repeated across the whole length of these four rows. Although we have not performed an analysis of the functionality of these small synaptic modules, it is interesting to notice that best evolved individuals exploit the possibility to maintain, repeat, and combine small building blocks of synaptic configuration.

The matrix-rewriting algorithm described in this article has been used only to support our argument on co-evolution of morphogenesis and synaptic plasticity, not as a starting point to investigate biological morphogenesis. As a matter of fact, this algorithm has a number of drawbacks. Although it can grow networks of infinite size, not all network morphologies can be expressed using this limited number of rewriting rules. Furthermore, for sake of comparison with the case where synaptic strengths are genetically encoded, this implementation requires the specification of the properties of individual synapses, which is not very efficient for evolving modularity at the neural level (instead than at the synaptic level). Finally, the process of morphogenesis happens instantaneously before connecting sensory and motor neurons to the robot. It would be interesting to let the network develop over time and vary the time constants of synaptic plasticity and neural growth, similar to a process of maturation. Preliminary
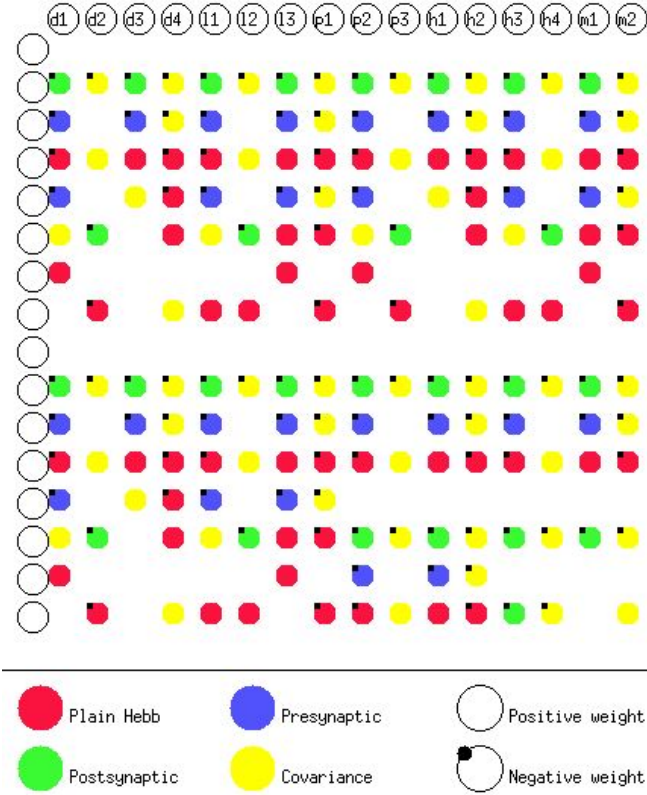
Figure 6: Evolved neural network whose genetic encoding specifies the rules of morphogenesis and of synaptic plasticity (corresponding to the individual trajectory plotted on the top left corner of figure5). Large circles on the left-most column represent the neurons in the network; the circles on the top row are the same neurons at the previous time step. Small coloured circles represent the presence of a synaptic connection between two neurons (incuding self-connections). The labels within the top row of neurons indicate their type: d = receive input from distance sensors; l = receive input from light sensors; p = receive input from photoreceptors; h = hidden neurons, i.e. neurons that are not directly connected to the sensory-motor system of the robot; m = their activation levels are used to set the speed of the two wheels.

computational investigations in this direction (Nolfi et al., 1994), but without synaptic plasticity, have shown that the maturation process is partially determined by early interactions with the environment and provides the organism with some adaptive power.

5 Conclusions

The results reported in this paper indicate that a morphogenetic evolutionary approach can benefit by co-evolution of rules of synaptic plasticity along with rules of neural morphogenesis. In particular, the local adaptive properties of the synapses make the system more evolvable than in the case where such synapses are fixed. Furthermore, since it is no longer necessary to specify initial synaptic strengths because these develop at run time while the system interacts with the environment, it is possible to apply this approach to a large variety of morphogenetic approaches while maintaining a compact genetic encoding.

In work described elsewhere (Floreano & Urzelai, 2000), we have started to investigate compact genetic encodings where the smallest encoded unit is the neuron, instead of the synapse. Five bits encode the type of neuron (excitatory or inhibitory), the type of plasticity rule that applies to all incoming synapses for that neuron (four hebbian rules on 2 bits), and the learning rate for all incoming synapses (four values on two bits), respectively. Preliminary experiments with predefined architectures show that this type of genetic encoding is more evolvable and more scalable to large networks than one where the smallest unit is the synapse. This is because in a neuron-based encoding the genetic length necessary for encoding a fully connected network of N neurons is proportional to N whereas in a synapse-based encoding the length is proportional to N^2. Current work is aimed at developing further the neuron-based encoding by adding morphogenesis and temporal development.

Acknowledgements

Reference

Cangelosi, A., Parisi, D., & Nolfi, S. (1994). Cell division and migration in a genotype for neural networks. *Network, 5,* 497–515.

Cliff, D., & Miller, G. F. (1996). Co-evolution of Pursuit and Evasion II: Simulation Methods and Results. In Maes, P., Matarić, M., Meyer, J., Pollack, J., Roitblat, H., & Wilson, S. (Eds.), *From Animals to Animats IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior.* MIT Press-Bradford Books, Cambridge, MA.

Eggenberger, P. (1996). Cell interactions as a control tool of developmental processes for evolutionary robotics. In Maes, P., Matarić, M., Meyer, J., Pollack, J., Roitblat, H., & Wilson, S. (Eds.), *From Animals to Animats*

IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, pp. 440–448. MIT Press-Bradford Books, Cambridge, MA.

Fahlman, S. E., & Lebiere, C. (1990). The Cascade-Correlation Learning Architecture. In Touretzky, D. S. (Ed.), *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann, San Francisco.

Floreano, D., & Urzelai, J. (2000). Evolutionary robots with online self-organization and behavioral fitness. *Neural Networks, 13*, 431–443.

Grossberg, S. (1987). Competititve learning: From interactive activation to adaptive resonance. *Cognitive Science, 11*, 23–63.

Gruau, F. (1992). Genetic systems of boolean neural networks with a cell rewriting developmental process. In Whitley, D., & Schaffer, J. D. (Eds.), *Combination of Genetic Algorithms and Neural Networks*. IEEE Computer Society Press, Los Alamitos, CA.

Gruau, F. (1994). Automatic definition of modular neural networks. *Adaptive Behavior, 3*, 151–183.

Gruau, F., & Quatramaran, K. (1997). Cellular encoding for interactive evolutionary robotics. In Husbands, P., & Harvey, I. (Eds.), *Proceedings of the 4th European Conference on Artificial Life* Cambridge, MA. MIT Press.

Husbands, P. (1994). A force field development scheme for use with genetic encodings of network-based sensorymotor control systems. Tech. rep. CSRP-326, School of Cognitive and Computing Science, University of Sussex.

Jakobi, N. (1995). Harnessing morphogenesis. Tech. rep. CSRP-423, School of Cognitive and Computing Science, University of Sussex.

Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems, 4*, 461–476.

Kodjabachian, J., & Meyer, J. (1998). Evolution and development of neural controllers for locomotion, gradient-following, and obstacle avoidance in artificial insects. *IEEE Transactions in Neural Networks, 9*, 796–812.

Kramer, A. H., & Sangiovanni-Vincentelli, A. (1989). Efficient parallel Learning Algorithms for Neural Networks. In Touretzky, D. S. (Ed.), *Advances in Neural Information Processing Systems 1*. Morgan Kaufmann, San Francisco.

Michel, O. (1997). Evolutionary neurogenesis applied to mobile robotics. In Patel, M., & Honovar, V. (Eds.), *Advances in Evolutionary Synthesis of Neural Systems*. MIT Press, Cambridge, MA.

Nolfi, S., Miglino, O., & Parisi, D. (1994). Phenotipic Plasticity in Evolving Neural Networks. In Nicoud, J.-D., & Gaussier, P. (Eds.), *Proceedings of the conference From Perception to Action*. IEEE Computer Press, Los Alamitos, CA.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Representations by Back-Propagation of Errors. *Nature, 323*, 533–536.

Siddiqi, A. A., & Lucas, S. M. (1998). A comparison of matrix rewriting versus direct encoding for evolving neural networks. In *Proceedings of the IEEE International Conference on Evolutionary Computation 1998 (ICEC'98)* Piscataway, NJ. IEEE Press.

Strogatz, S. H. (2001). Exploring complex networks. *Nature, 410*, 268–276.

Willshaw, D., & Dayan, P. (1990). Optimal plasticity from matrix memories: What goes up must come down. *Neural Computation, 2*, 85–93.