Evolving Neuromodulatory Topologies for Reinforcement Learning-like Problems

Andrea Soltoggio, Peter Dürr, Claudio Mattiussi and Dario Floreano

Abstract— **Environments with varying reward contingencies constitute a challenge to many living creatures. In such conditions, animals capable of adaptation and learning derive an advantage. Recent studies suggest that neuromodulatory dynamics are a key factor in regulating learning and adaptivity when reward conditions are subject to variability. In biological neural networks, specific circuits generate modulatory signals, particularly in situations that involve learning cues such as a reward or novel stimuli. Modulatory signals are then broadcast and applied onto target synapses to activate or regulate synaptic plasticity.**

Artificial neural models that include modulatory dynamics could prove their potential in uncertain environments when online learning is required. However, a topology that synthesises and delivers modulatory signals to target synapses must be devised. So far, only handcrafted architectures of such kind have been attempted. Here we show that modulatory topologies can be designed autonomously by artificial evolution and achieve superior learning capabilities than traditional fixed-weight or Hebbian networks. In our experiments, we show that simulated bees autonomously evolved a modulatory network to maximise the reward in a reinforcement learning-like environment.

I. Introduction

Neuromodulation in biological neural networks has been recognised to be a key factor in network dynamics. Experimental evidence shows that neuromodulation plays an important role in several neural substrates, from the invertebrate *Aplysia* to the human brain [1], [2].

Neuromodulation exerts a regulatory action on synaptic plasticity, suggesting a close relation with important functions such as memory, learning and adaptivity. The central role of these functions in neuroscience has brought considerable focus to the study of neuromodulation in biological systems [3] and to the formulation of computational models [4].

Studies on synaptic plasticity show that the well known homosynaptic Hebbian rule is not the only mechanism that leads to synaptic growth. Another relevant factor is the concentration of neuromodulators at the synapse level that seems determinant in the growth and stability of synaptic connections. In this case, plasticity is named heterosynaptic because it involves the activity of a third modulatory neuron. Figure 1 describes graphically the difference between homo- and heterosynaptic plasticity.

Several studies reviewed in [2] indicate that the pairing of pre- and postsynaptic activity with a modulatory signal leads to the activation of transcription factors at the synapse level, which in turn cause a permanent growth of the synaptic contact. This growth is referred as L-LTP (Late phase - Long Term Potentiation) because of the long decay time of the synaptic strength. L-LTP induced by neuromodulation is a cause of synaptic stability and, therefore, a potential candidate to explain memory functions involving neural wiring. Hence, modulatory systems seem to assume the function of learning switches that project connections to target synapses, instructing distinct neural areas to acquire input/output correlation at given times.

Although the micro-level synaptic effects of neuromodulation are topic of many studies, other important findings relate neuromodulation with behavioural phenomena in animals and humans. The implication of dopamine and other neuromodulators in learning, decision making and memory functions in the brain is currently an active research field [5], [6]. A significant experiment described in [7] shows relations in the acquisition of new tasks with dopamine release in monkeys' brains. Following studies relate dopamine with prediction errors in reinforcement learning-like environments [8], [9], [10]. This suggests a role of neuromodulation in temporal difference models of animal learning [11] and their similarities with Temporal Difference (TD) in reinforcement learning theory [12].

Computational models of modulatory dynamics in neural systems have been proposed with the aim of understanding the neural substrate underlying reinforcement learning-like behaviour [4], [13], [14], [15], [16], [17], [18], [19]. A substantial issue when devising a model is the design of sources and pathways of neuromodulation, i.e. how the modulatory signals are generated and which neural areas are targeted. As discussed in [17], reinforcement learning, actor-critic and reward-based neural models are loosely implemented after biological neural architectures. Although recent progress in neuroscience and neuroethology has made possible to identify modulatory centres and pathways in neural systems, a precise mapping and understanding is far from being achieved. Therefore, most of computational models of neuromodulation start from given assumptions regarding possible sources and pathways of modulatory signals. In [13] and [15], a simple neural architecture was devised to implement heterosynaptic plasticity for the neurocontroller of a simulated foraging bee in an uncertain environment. Although the genetic algorithm used in [15] was capable of enabling or disabling connections between inputs and the

Andrea Soltoggio is with the School of Computer Science, The University of Birmingham, Birmingham B15 2TT, United Kingdom, (email: a.soltoggio@cs.bham.ac.uk). Peter Dürr, Claudio Mattiussi and Dario Floreano are with the Laboratory of Intelligent Systems, EPFL, CH-1015 Lausanne, Switzerland, (email: peter.duerr@epfl.ch, claudio.mattiussi@epfl.ch, dario.floreano@epfl.ch).

output neuron, the neural architecture was constrained to the input neurons and one output neuron. Generally speaking, neuromodulatory architectures that have been designed so far are handcrafted and tentative, and do not guarantee to exploit the full potential of modulatory dynamics in neural networks.

In this paper, we propose a method to design autonomously neural network topologies with neuromodulation, and explore their capabilities without the constraint of a predetermined architecture. Our hypothesis is that, if neuromodulation increases the computational power, neurocontrollers with such characteristic would emerge autonomously in uncertain environments where learning and adaptivity give an advantage.

To test this hypothesis, it is essential to provide artificial evolution with an algorithm capable of feature selection and evolving neural topologies. Analog Genetic Encoding (AGE) [20], [21] is a method to encode neural topologies that provides such functionality. AGE has been proved efficient when combined with a genetic algorithm for the evolution of different kind of networks, namely neural networks [22], electronic circuits [20], [21] and Gene Regulatory Networks [23]. Here, AGE is applied to the evolutionary search of network topologies with neuromodulation.

The chosen problem is a foraging task described in [13], [15]. The changing reward conditions necessitate a continuous update of the strategy to maximise the food intake. Therefore, optimal strategies in this reinforcement learning-like variable environment require online learning capabilities. The neuromodulatory architecture devised in [13], [15] has proved to be beneficial to the task, allowing artificial bees to associate a flower-colour to the current high rewarding flower. In this paper, instead of assuming a predetermined architecture, we carry out an evolutionary search of neuro-topologies, modulatory and input features, and learning rules. The results are qualitatively compared to those in [15]: the performance of controllers considerably outperform that of the previous handcrafted architecture. Moreover, one single neurocontroller can cope with a more complex, extended scenario than the one in [15]. An additional comparison is also made with evolutionary runs where neuromodulation was not allowed.

The analysis of the networks shows the effective modulatory dynamics that emerged from evolution and enabled to solve the foraging problem. Thus, the method proved its validity in the search of neural network topologies with neuromodulation.

The rest of the paper is organised as follows. Section II describes the problem of evolving topologies with neuromodulation and the proposed method using AGE and artificial evolution. Section III describes in detail the simulated bee and the artificial environment. Implementation details are listed in section IV. The results are illustrated in section V with emphasis on the evolved behaviour and an insight on the neuromodulatory dynamics. The paper ends with final remarks in the conclusions.

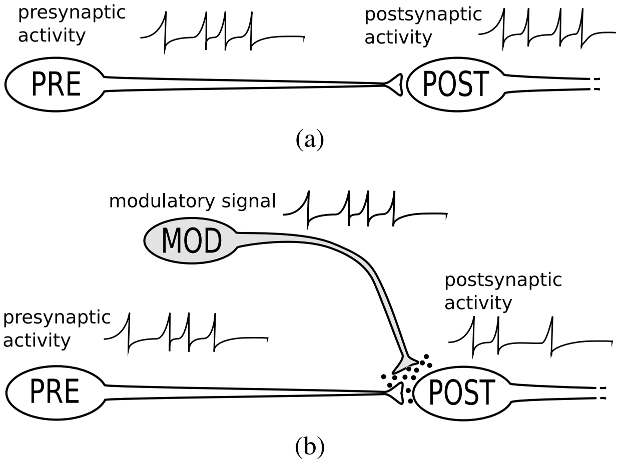

Fig. 1. (a) Homosynaptic mechanism: the connection strength is updated as function of pre- and postsynaptic activity only. (b) Heterosynaptic mechanisms: the connection growth is mediated by neuromodulation, i.e. the amount of modulatory signal determines the response to Hebbian plasticity. The dots surrounding the synapse represent the concentration of neuromodulatory chemicals released by the modulatory neuron. Neuromodulators such as acetylcholine (ACh), norepinephrine (NE), serotonin (5-HT) and dopamine (DA) have been identified.

II. Artificial Evolution of Neuromodulation

In the introduction we have underlined the research problem regarding the design of topology when devising neuromodulatory network models. We proposed to employ an evolutionary approach to evolve such topologies. To investigate the autonomous emergence of topologies with neuromodulation, an algorithm should be capable of 1) encoding two types of neurons, traditional excitatory/inhibitory neurons and modulatory neurons; 2) encoding weights and network topology among an arbitrary number of standard and modulatory neurons. In addition, because modulation is applied to regulate some form of synaptic plasticity, fixed or evolved plasticity rules need to be available to the neural network.

The design and the evolutionary search of network topologies have been the focus of research for many years [24]. Recently, different aspects on the evolution of neural networks have been taken into consideration to formulate advanced algorithms for the search of both topology and weights. At least two algorithms, NeuroEvolution of Augmenting Topologies (NEAT) [25] and neuroevolution with Analog Genetic Encoding (AGE) [20], have been established as efficient methods for evolving network topologies and weights, and their performance has been assessed with benchmarks and applications.

AGE was chosen for the topology search in this experiment. Following, an overview of the algorithm is given.

A. Analog Genetic Encoding (AGE)

Because AGE is an established method and it is used here exclusively as a tool, we will provide a concise description for the general understanding and the necessary information for reproducing the algorithm. For a further insight of AGE, we recommend the cited literature [20], [21], [22], [23].

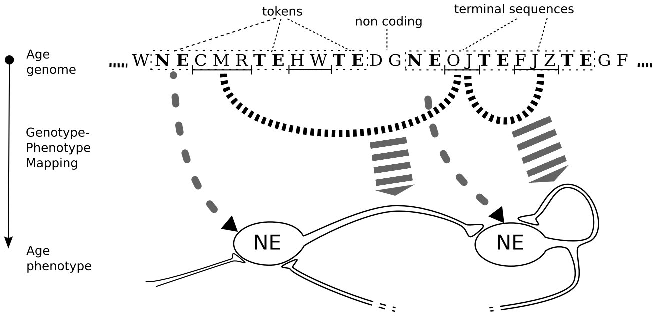

Fig. 2. Process of genotype-phenotype mapping in AGE for a neural network. The part of the genome shown here encodes two neurons signalled by the token NE. The genome is scanned sequentially and device tokens (NE) are decoded into network nodes, more specifically neurons in this example. The process is marked by dotted arrows. Terminal sequences (arbitrary sequences of characters that precede the terminal token TE) of each node are aligned (two at a time) to derive a measure of similarity between two sequences. This measure of similarity, also called alignment score, is mapped into the connection weight between the nodes to which the sequences belong. For example, when the input sequence of a device is aligned with the output sequence of the same device (at the right of the figure), the resulting weight represents the recurrent self connection. In this particular example, the first terminal sequence after the device token NE is the input, the second terminal sequence is the output. In general, according the user specifications, a device can have more inputs or outputs.

AGE represents an analog network by means of an artificial genome where nucleotides are expressed by the characters of an alphabet Ω, for instance the letters A-Z. Nodes in the network, also called devices, are encoded by particular sequences of characters, the tokens. Each token signals the presence of a device that is decoded into a network node in the phenotype; different types of devices – representing different kind of network nodes – can be present in the genome. Figure 2 shows the genotype-phenotype mapping process.

Each device has a certain number of inputs and outputs that, in the case of neurons, represent dendrites and axon projections. Inputs and outputs of devices are encoded with terminal sequences, i.e. arbitrary sequences of characters that follow device tokens (NE in the figure) and are limited by a terminal token (TE). Once all the network nodes are extracted, the connections among them is derived applying the following procedure: the output terminal sequence of a device is aligned with the input terminal sequences of all other devices; each alignment produces an alignment score – an index of similarity between the two terminals – that is consequently mapped into a connection weight. The mapping from alignment scores to network weights is done through a quantisation process where alignment scores in a given range are converted into a range of real-valued weights. Alignment scores under a certain threshold result in no connection between two nodes. Therefore, terminal sequences encode implicitly the neural topology.

Different kinds of networks can be represented according to the device specification given by the final user.

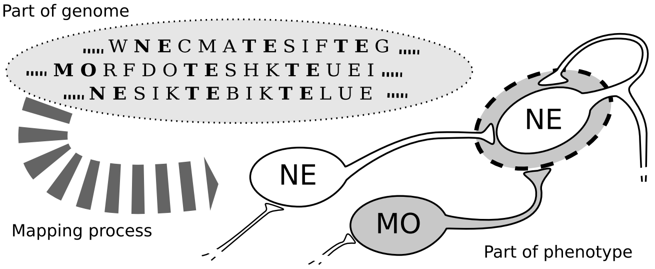
Part of genome
.... W N E C M A T E S I F T E G ....
.... M O R F D O T E S H K T E U E I ....
.... N E S I K T E B I K T E L U E ....
Mapping process
NE
NE
MO
Part of phenotype

Fig. 3. Example of AGE phenotype when modulatory neurons (token MO) are added to the network alongside standard neurons (token NE). The projection from a modulatory neuron to a standard neuron is indicated by a dashed circle around the postsynaptic neuron. For all synapses connecting to the postsynaptic neuron, plasticity is regulated by the modulatory signal.

B. AGE and Settings for Neuromodulatory Topologies

For our purpose, two different devices were specified to encode standard and modulatory neurons.

Figure 3 shows an example of a part of a phenotype where two standard neurons and one modulatory neuron are decoded from the genome, assuming NE and MO as device tokens.

For the experiment in this paper, neurons have a discrete time dynamic. The output $O_l(t)$ of neuron l is equal to $2/[1+\exp(A(t-1))]-1$ for standard neurons and $1/[1+\exp(A(t-1)-1)]$ for modulatory neurons, with $A_l(t) = 3 \cdot \sum(w_{jl} \cdot O_j(t))$, where w_{jl} is the connection weight from the standard neuron j to the neuron l. According to these definitions, standard neurons have a sigmoid output in the interval [-1,1], whereas modulatory neurons produce an output in the interval [0,1] and have an implicit bias of -1. It is important to note that this setting has the purpose of having modulatory neurons that exert very low modulation unless excited by positive signals.

Modulatory neurons that project on standard neurons do not contribute to neuronal activity. The modulatory signal has a regulatory function on the synaptic plasticity of the receiving neuron. The plasticity rule used by a neurocontroller is evolved alongside the network. Given two standard neurons j and l, an existing connection from j to l is updated according to the following equation

$$\Delta w_{jl}(t) = mo(t) \cdot \eta \cdot \\ \cdot [A \cdot V(t)P(t) + B \cdot V(t) + C \cdot P(t) + D] \tag{1}$$

where $mo(t)$ is the modulatory signal, η is a scaling parameter, and the term between square brackets is the set of plasticity rules. $V(t)$ is the presynaptic value (output of neuron j), $P(t)$ the postsynaptic value (output of neuron l), A, B, C, D are evolvable parameters that express the coefficients of the plasticity rules. The modulatory signal $mo(t)$ perceived by the postsynaptic neuron (l) is the sum of all modulatory signals delivered to that particular neuron.

III. The Reinforcement Learning-like Problem

As outlined in the introduction, neuromodulation is considered a key feature for neural systems dealing with uncertain environments, where associations between actions and reward change over time. For this reason, an artificial environment aimed to the study of neuromodulation should include such characteristics.

Foraging tasks of bees and bumblebees are well known problems that require learning and adaptivity. The flight to flower fields for nectar collection is a risky activity: predators determine a high mortality rate during foraging missions. Therefore, bees need to maximise the nectar intake by visiting preferably flowers that yield high quantities of nectar. However, different flowers provide variable quantities of nectar depending on the time of the day, season, weather conditions and other variable environmental factors.

These conditions determine a reinforcement learning-like environment where the nectar intake upon landing represents a measure of reward. The type of flower, often discernible by the colour, is a conditioned stimulus that becomes a predictor of an expected reward. Hence, reward expectations determine a strategy aimed to maximise the total reward over a certain number of trials. Upon changes of reward contingencies, high rewarding flowers turn into low rewarding, thus, reward expectations are not fulfilled resulting in prediction errors.

To support this view, an identified interneuron in honeybees appears to deliver gustatory stimuli representing reward values upon nectar collection [26]. This finding and following studies [27], [28], [29] contribute to the explanation of associative learning in the neural substrate of the honeybee. A computational model that tries to reproduce the biological evidence of reinforcement learning and neuromodulation is described in [13]. Later, the same experimental setting was used in [15] to optimise a neuromodulatory network by means of a genetic algorithm. Here, we adopt the same simulated bee and artificial uncertain environment.

A. *The Simulated Bee*

A bee flies in a simulated 3D space with a flower field of 60 by 60 meters drawn on the ground. Two types of flowers are represented on the field by blue and yellow 1-meter square patches. The outside of the field and the sky are represented by grey colour.

During its lifetime, the bee performs a number of flights starting from a random height between 8 and 9 meters. The bee flies downwards in a random direction at a speed of 0.5m/s. A single cyclopean eye (10-degree cone view centred on the flying direction) captures the image seen by the bee. The image is processed to obtain the percentages of blue, yellow and grey colours that are fed into the neural controller.

For each time step (1 sec sampling time) the bee decides whether to continue the flight in the current direction or to change it to a new random heading. The activation value $\mathrm{A}(t)$ of an output neuron determines the probability of changing direction given by $\mathrm{P}(t) = [1 + \exp(\mathrm{m} \cdot \mathrm{A}(t) + \mathrm{b})]^{-1}$, where m and b are evolvable parameters. Figure 4 shows a portion of the 3D space where the flight is simulated.

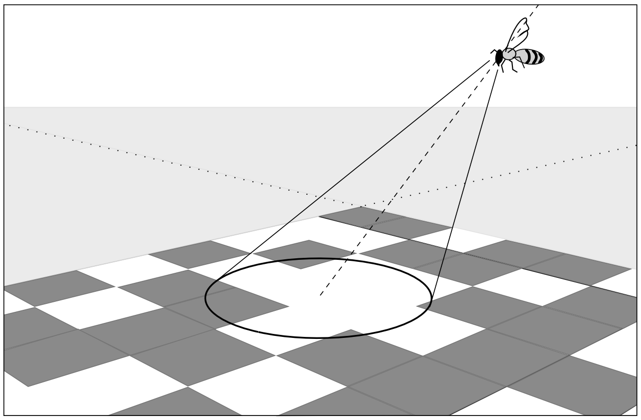

Fig. 4. View on the flying 3D space and the simulated bee. Blue and yellow flowers are represented by dark and light squares. The bee flies downwards in any random direction and approaches the field under its view cone. The dashed line shows a possible landing trajectory.

TABLE I

Rewarding policies. P indicates the probability of the reward.

Scenario	Nectar of the high rewarding flower	Nectar of the low rewarding flower
1	$0.8\mu l$	$0.3\mu l$
2	$0.7\mu l$	$1.0\mu l$ with P=0.2 $0.0\mu l$ with P=0.8
3	$1.6\mu l$ with P=0.75 $0.0\mu l$ with P=0.25	$0.8\mu l$ with P=0.75 $0.0\mu l$ with P=0.25
4	$0.8\mu l$ with P=0.75 $0.0\mu l$ with P=0.25	$0.8\mu l$ with P=0.25 $0.0\mu l$ with P=0.75

B. *Scenarios*

The two flowers, characterised by blue and yellow colours, yield a certain amount of nectar that is provided to the bee upon landing in the form of a reward input. Rewards can be given on a deterministic or probabilistic basis.

Here we introduce four possible scenarios according to four possible reward policies as in [15]. Scenario 1 provides two deterministically rewarding flowers; scenario 2 has one deterministically and one probabilistically rewarding flower; scenario 3 and 4 have both probabilistically rewarding flowers. Regardless of the reward policies, each scenario has a high rewarding and a low rewarding flower, meaning that one flower yields in average more nectar than the other. Table I provides the numerical values of rewards in each of the four scenarios.

An optimal strategy is required to associate a flower-colour with the currently high rewarding flower. Note that a deterministically rewarding flower provides a mean reward that corresponds to the value received on the single trial, whereas probabilistically rewarding flowers require more trials to obtain an estimated average reward. As a consequence, scenario 1 and 2 constitute an easier problem to solve than scenario 3 and 4. The evolved bees in [15] solved only scenarios 1 and 2 although the evolutionary search was attempted on probabilistically based scenarios as well.

Initially, the blue and yellow colours are assigned to th high and low rewarding flowers respectively, or vice versa o a random basis. During the scenario, the colours are inverted thus changing the association between colour and high/lov reward. The random initial assignment and the followin switch of colours introduce uncertainty in the environment.

Note that the numerical values for quantities of necta shown in Table I have been chosen carefully to exclude trivia strategies based on the preference for a given value or interva of values.

The lifetime of a bee is simulated by presenting scenario 1, 2 and 3 sequentially. Scenario 4 is used for testing only. Three hundred flights are performed with scenario switching points at flights 101 ± 15 and 201 ± 15. The colours of flowers are inverted about half way through each scenario at flights 51 ± 15, 151 ± 15 and 251 ± 15. Colours are also inverted at scenario switching-points with probability 0.5: this is to avoid a predictable pattern of the high rewarding flower.

IV. Implementation

Three input neurons provide the percentage of grey, blue and yellow colours seen at each time step. An input neuron for the reward provides a measure of the nectar collected upon landing. The reward input is 0 during the flight, and assumes the value of the nectar content at the landing step only. Additionally, a landing signal that assumes value 1 upon landing and remains 0 during the flight is provided. The landing signal is particularly important to indicate when the expected reward is due and therefore allow the neural network to detect a prediction error. In [15], differential colour inputs were provided to the neurocontroller. We also made differential inputs available to evolution to assess their utility. An output neuron controls the actions of the bee. A constant input set to 1 served as bias. Connection weights are in the range [0.3, 30] obtained with logarithmic quantisation from alignment scores in the interval [16,36]. Alignment scores are computed according to the scoring matrix described in [20, page 89].

Seven parameters are evolved with the neurocontroller: parameters m and b for the probability of direction change; parameters A, B, C, D and η from equation 1. Parameters are represented as real values in the following range: [5,45] for m, [0,5] for b, [-1,1] for A, B, C, D and [0.05,50] for η.

The search on the AGE genome is performed by a standard, fairly configurable evolutionary algorithm [30]. For this experiment we set a population size of 100. The fitness is the amount of nectar collected by each individual during the evaluation. The truncation selection mechanism is applied to select the 50 best individuals from the population. The best individual is kept unchanged in the population. Recombination probability is 0.1. Mutation on the AGE genome is performed by nucleotide substitution and insertion that operate on a single nucleotide, fragment duplication and transportation that operate on sequences of more nucleotides (fragments) with probability $4.0 \cdot 10^{-4}$. A slightly higher probability of $4.5 \cdot 10^{-4}$ was applied to nucleotide and fragment deletion. Genomes of generation zero are initialised with two neurons for each type and random terminal sequences of length 25, i.e. random connection weights.

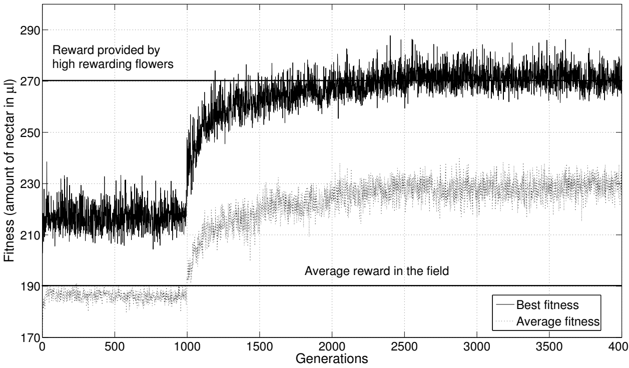

Fig. 6. Best and average fitness in one run.

V. Results

Fifty independent runs were executed. The runs terminated after 4000 generations. Forty-five out of the 50 runs discovered an online learning strategy. Figure 6 shows a typical example of fitness graph. The discovery of a strategy is indicated by a jump in the fitness values. Jumps in different runs occur at various times during evolution, some at an early stage, some later. However, once a strategy is found, the fitness values increase relatively quickly.

The average reward in the field ($190\mu l$ per lifetime) is the threshold that indicates when an association between reward and flower-colour is discovered. The maximum fitness is not well defined given the stochastic nature of rewards in scenario 3. A reference value, however, is given by $270\mu l$ that is the sum of average rewards provided by optimal choices during a lifetime.

Two additional sets of experiments without neuromodulation were executed for comparison. Twenty runs were performed with neuromodulation switched off, therefore evolving topologies of fixed-weight networks. Other twenty runs were executed with a constant neuromodulatory value of 1 for all neurons, therefore evolving topologies with plasticity fully enabled. Only two runs out of forty (all from the fixed-weight case) displayed a learning strategy, allowing to cross the $190\mu l$ threshold. However, even in these successful runs, performance was low as controllers displayed learning in scenario 1 or 2 only, while failing on the more difficult probabilistically rewarding scenario 3.

A. Adaptivity of Networks to Scenarios

At the end of the evolutionary search, the controllers were tested on the 3-scenario life used for evolution. Figure 5(a) shows the behaviour of one bee. At contingencies and scenario switching-points[1], the bee requires a certain number of flights to change its preference. However, the correct

[1] The variability of switching-points during evolution was removed during testing to have equally long scenarios.

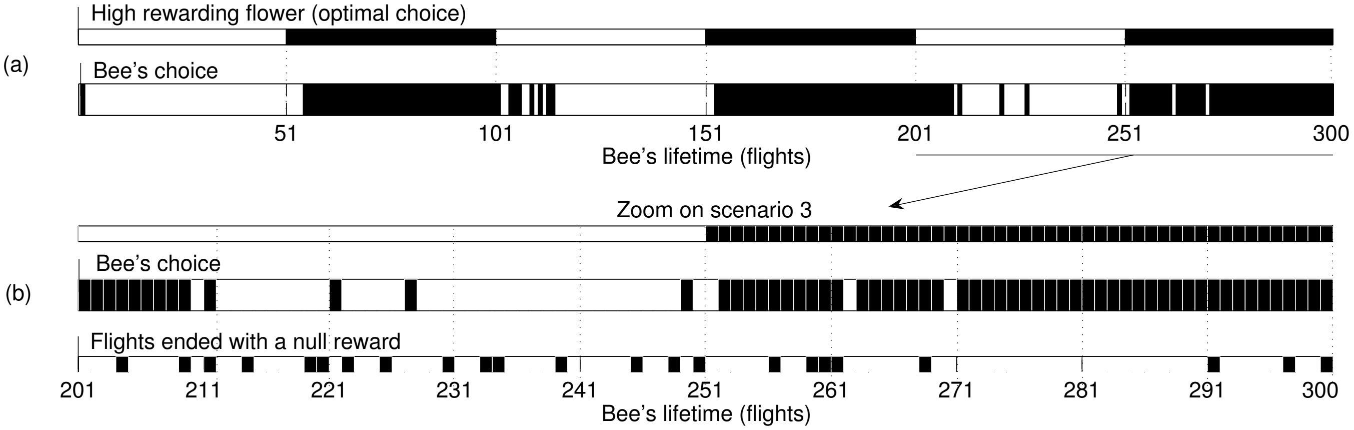

Fig. 5. Behaviour of a bee during a 300-flight lifetime. (a) The choice of flower for each of the 300 flight is reported on the horizontal time-scale. The top bar indicates the colour of the high-rewarding flower, i.e. the optimal choice. The second bar shows the choice made by the evolved bee. (b) Zoom in of scenario 3 (last hundred flights): an additional horizontal bar at the bottom shows the flight in which the bee collected a null reward.

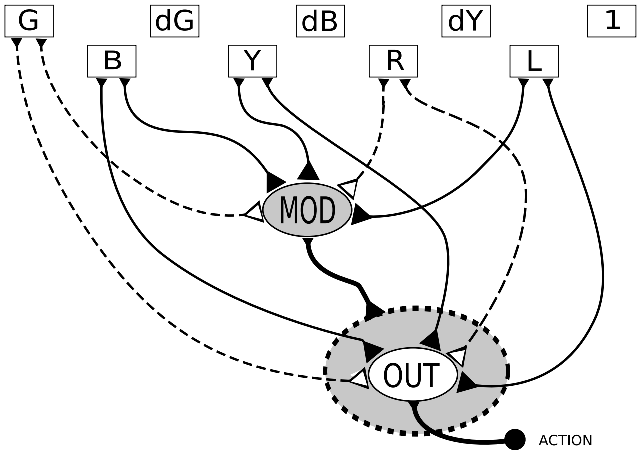

Fig. 7. Network topology of a well-performing bee. The square boxes on top represent the input neurons where G, B and Y are the percentages of grey, blue and yellow colours seen by the bee; dG, dB, dY represent differential colour values at each step. R and L are the reward and landing signals. The square labelled ”1” is a constant input of 1 that provides a bias to the neurons. Continuous lines with black triangles indicate positive connections, dashed lines with white triangles negative connections. Dashed circles around a neuron indicate that the neuron is reached by a neuromodulatory connection and the synapses that connect to that neuron undergo synaptic plasticity according to equation 1. The initial weights are: G-Out: -0.37; G-Mod: -0.37; B-Out: 0.175; Y-Out: 0.30; B-Mod: 0.60; Y-Mod: 0.60; R-Mod: -0.3; R-Out: -14.66; L-Mod: 1.95; L-Out: 9.56. Evolvable parameters are: A: -0.79; B: 0.0; C: 0.0; D: -0.038; η : 0.79; m: 42.47; b: 4.75.

association between colour and high rewarding flower is always achieved.

It is interesting to note that the bee seems to take longer to switch preference when the scenario changes (at flights 101 and 201), whereas it changes preferences more rapidly when the colour is inverted (at flights 51, 151, and 251). This is because strategies vary considerably between scenarios, for example requiring to avoid a zero-rewarding flower in scenario 2, but not so in scenario 3[2]. Figure 5 suggests that the bee has remarkable learning capabilities that do not just allow the association of colour stimulus and reward, but also the determination of a better rewarding flower on the basis of long term historical information from sampling. To support further this conclusion, we plotted the flights that ended with a zero-reward in Figure 5(b). The zoom on scenario 3 show that when a flower has been chosen, the bee insists visiting the same flower in spite of zero-rewards that are occasionally collected. However, the deceiving experience of more zero-rewards in a row makes the bee switch flower at flight 262, after sequentially collecting a null reward from the good flower three times. Yet, the preference is switched back immediately to the correct one.

Scenario 1,2 and 3 constituted the simulated lifetime of the bee during evolution. A more challenging test was carried out on the unseen scenario 4: the two flowers yield the same reward but have different probabilities of returning a zero-reward (see Table I). Surprisingly, Figure 8 shows that the bee is able to learn which flower returns a high mean in the long run. The test was tried twice with considerably different numerical values of reward.

B. *Network Analysis*

To understand the neural principles and the main characteristics of the evolved solutions, we examined the components and connections of the best 5 networks of each successful run, in total 225 networks. Because each independent run was free to evolve any topology, plasticity rules and modulatory structure, a comparison of different solutions is difficult. However, we noticed that successful controllers presented some common features. Figure 7 shows an example of an evolved network.

Differential inputs are used at 10% only, suggesting that these inputs proposed in [15] are not necessary. The reward

[2] While a null reward in scenario 2 is given only by the low rewarding flower, in scenario 3 the high rewarding flower gives occasionally a null reward, see Table I

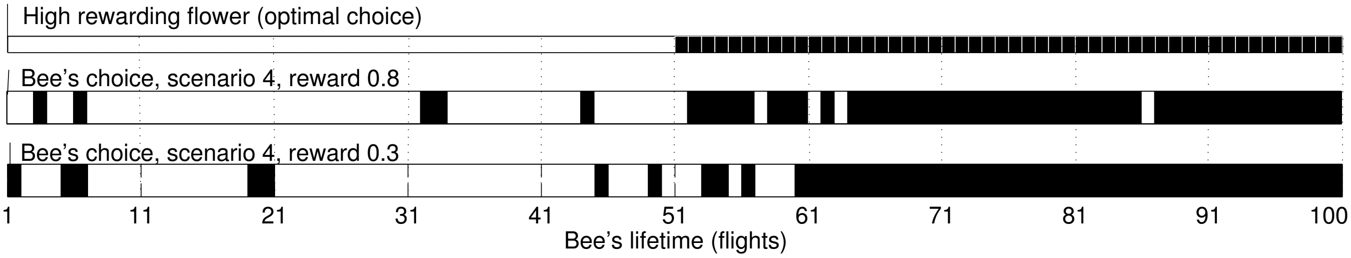

Fig. 8. The bee is tested twice on the unseen probabilistically rewarding scenario 4 with rewards $0.8\mu l$ and $0.3\mu l$.

signal (R) is used in 100% of controllers. This is due to the fact that only by listening to the reward signal the network can discover the high rewarding flower and detect changing contingencies. The landing signal (L) is present in 220 networks, indicating that evolution found this signal beneficial. At a further analysis, we found that in approximately 75% of solutions, the landing signal projects excitatory connections to modulatory and standard neurons, while the reward input sends inhibitory signals. Thus, the modulatory signal is activated by landing, and enables the network to learn new input/output correlations. Simultaneously, the reward signal corrects the synapse update according to a measure of good/bad surprise.

All the networks have at least one modulatory neurons and one standard neuron for the output. In average, each network has 1.11 modulatory neurons and 1.13 standard neurons. This means that the complex foraging task can be solved by a simple neural architecture when neuromodulation is provided.

Figure 9 gives an important insight on the neural dynamics. The modulatory signal saturates at landing, instructing the network to update synaptic weights. A low level of modulation is present during the flight as well, allowing for a slow decay of synaptic weights, and reflecting a decay of expectation in absence of reward. Most interesting is also the fact that neuromodulation drops to zero at times: this happens when the bee sees grey colour outside the field. Because the outside of the field provides null reward in all scenarios, and it is not subject to contingency change, synaptic plasticity - and thus learning - is switched off. In other words, the evolved network with neuromodulation enables learning only when the environmental contingencies require adaptation.

VI. Conclusions

Starting from the biological evidence on neuromodulatory dynamics, we suggest that Artificial Neural Networks (ANNs) learning capabilities can be enhanced with the inclusion of such models for synapse plasticity.

Here, we introduce a neural model of heterosynaptic plasticity and search the topology space with an evolutionary algorithm and Analog Genetic Encoding (AGE). The results show that the neurocontrollers autonomously discover neuromodulation during evolution and maximise the total reward in an uncertain foraging environment. Our solutions proved to acquire a general learning strategy capable of coping with more scenarios. These results outperform the neural controllers with fixed architecture described in [15] that solved only a subset of the proposed scenarios. It is remarkable that one controller do not only solve equally well all scenarios used during evolutions, but also cope successfully with a qualitatively different unseen scenario, regardless of numerical reward values. Additional experiments run for comparison without neuromodulation performed extremely poorly both in the case of fixed-weight networks and traditional Hebbian learning networks.

We showed that the key feature of neuromodulation consists in activating plasticity only at critical time steps, for example at landing when the reward stimulus is due, modulating synaptic update during flight and deactivating learning when it is not required.

Although the behaviour of the evolved bees displays a complex reinforcement learning dynamic, the neural controllers designed by evolution are compact and utilise few neurons. This suggests that neuromodulation provides an efficient tool to implement subsymbolic reinforcement learning mechanisms.

Our study showed that the evolution of neuromodulatory structures is possible and provides solutions with remarkable computational power. This opens the possibility of investigating the use of such networks for increasingly complex learning problems. Moreover, our evolutionary search of topologies was motivated by the evidence that such structures play an important role in biological neural substrates. Thus, the modulatory network topologies discovered by artificial evolution represent valid computational models for neuroscience and ethology.

Acknowledgements

Many thanks to Yael Niv for providing help in reproducing the original experimental settings. A special thanks to John Bullinaria who made this work possible with his support and advice.

References

[1] A. C. Roberts and D. L. Glanzman, "Learning in aplysia: looking at synaptic plasticity from both sides," *Trends in Neuroscience*, vol. 26, no. 12, pp. 662–670, December 2003.

[2] C. H. Bailey, M. Giustetto, Y.-Y. Huang, R. D. Hawkins, and E. R. Kandel, "Is heterosynaptic modulation essential for stabilizing hebbian plasticity and memory?" *Nature Reviews Neuroscience*, vol. 1, no. 1, pp. 11–20, October 2000.

[3] E. Marder and V. Thirumalai, "Cellular, synaptic and network effects of neuromodulation," *Neural Networks*, vol. 15, pp. 479–493, 2002.

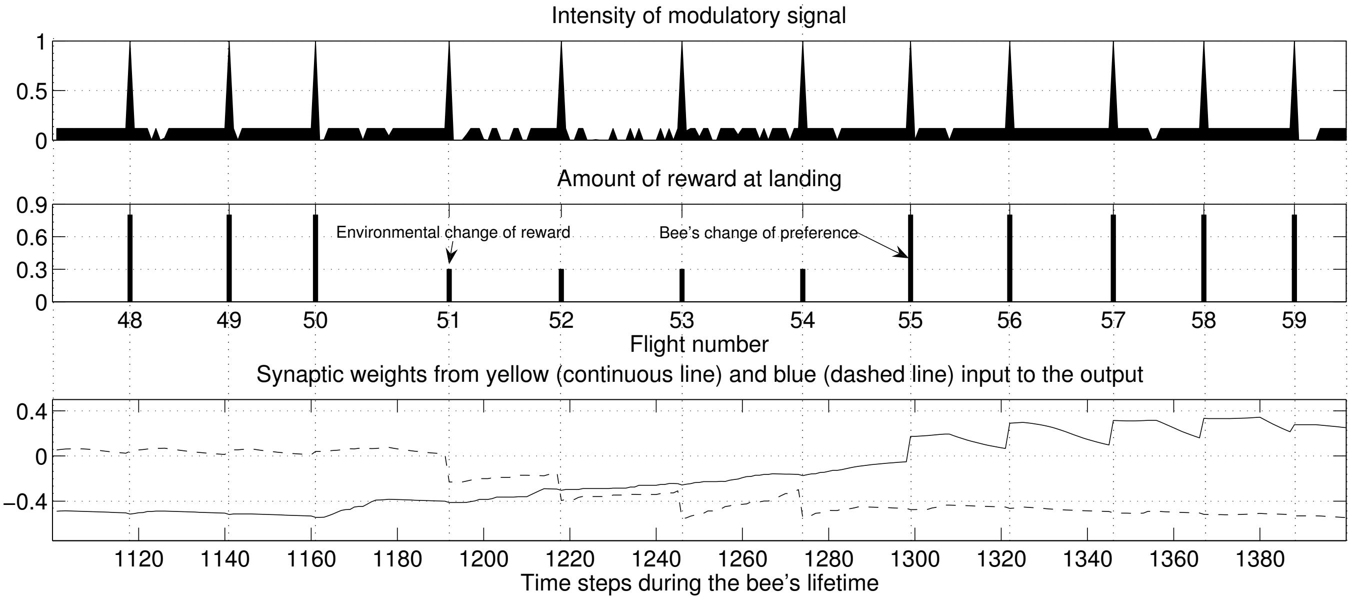

Fig. 9. Analysis of neural activity and weights. These figures captures a snapshot of the neural states of the network in Figure 7 while simulating the bee's lifetime reported in Figure 5. The top graph reports the intensity of the signal from the only modulatory neuron. The middle graph the amount of reward at the time of landing. The bottom graph shows the synaptic weights of colour inputs to output that determine the preference of the bee for one flower-colour. Because the modulatory signal remains low during the flight and increases at landing, a faster synaptic update occurs at landing.

[4] J.-M. Fellous and C. Linster, "Computational models of neuromodulation," *Neural Computation*, vol. 10, pp. 771–805, 1998.

[5] P. R. Montague, S. E. Hyman, and J. D. Cohen, "Computational roles for dopamine in behavioural control," *Nature*, vol. 4, pp. 2–9, 2004.

[6] N. D. Daw, J. P. O'Doherty, P. Dayan, B. Seymour, and R. J. Dolan, "Cortical substrates for exploratory decisions in humans," *Nature*, vol. 441, no. 15, pp. 876–879, June 2006.

[7] W. Schultz, P. Apicella, and T. Ljungberg, "Responses of Monkey Dopamine Neurons to Reward and Conditioned Stimuli during Successive Steps of Learning a Delayed Response Task," *The Journal of Neuroscience*, vol. 13, pp. 900–913, 1993.

[8] P. R. Montague, P. Dayan, and T. J. Sejnowski, "A Framework for Mesencephalic Dopamine Systems Based on Predictive Hebbian Learning," *The Journal of Neuroscience*, vol. 16, no. 5, pp. 1936–1947, March 1996.

[9] W. Schultz, P. Dayan, and P. R. Montague, "A Neural Substrate for Prediction and Reward," *Science*, vol. 275, pp. 1593–1598, 1997.

[10] W. Schultz, "Predictive Reward Signal of Dopamine Neurons," *Journal of Neurophysiology*, vol. 80, pp. 1–27, 1998.

[11] R. S. Sutton and A. G. Barto, *Time-Derivative Models of Pavlonian Reinforcement*. MIT Press, 1990, ch. 12, pp. 497–537.

[12] ——, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.

[13] P. R. Montague, P. Dayan, C. Person, and T. J. Sejnowski, "Bee foraging in uncertain environments using predictive hebbian learning," *Nature*, vol. 377, pp. 725–728, October 1995.

[14] R. E. Suri, J. Bargas, and M. A. Arbib, "Modeling functions of striatal dopamine modulation in learning and planning," *Neuroscience*, vol. 103, no. 1, pp. 65–85, 2001.

[15] Y. Niv, D. Joel, I. Meilijson, and E. Ruppin, "Evolution of Reinforcement Learning in Uncertain Environments: A Simple Explanation for Complex Foraging Behaviours," *Adaptive Behaviour*, vol. 10, no. 1, pp. 5–24, 2002.

[16] K. Doya, "Metalearning and neuromodulation," *Neural Networks*, vol. 15, no. 4-6, pp. 495–506, 2002.

[17] D. Joel, Y. Niv, and E. Ruppin, "Actor-critic models of the basal ganglia: new anatomical and computational perspectives," *Neural Networks*, vol. 15, pp. 535–547, 2002.

[18] O. Sporns and W. H. Alexander, "Neuromodulation and plasticity in an autonomous robot," *Neural Networks*, vol. 15, pp. 761–774, 2002.

[19] K. Doya and U. Eiji, "The Cyber Rodent Project: Exploration and Adaptive Mechanisms for Self-Preservation and Self-Reproduction," *Adaptive Behavior*, vol. 13, no. 2, pp. 149–160, 2005.

[20] C. Mattiussi, "Evolutionary synthesis of analog networks," Ph.D. dissertation, Laboratory of Intelligent System (LIS) - EPFL - Lausanne, Switzerland, Lausanne, 2005. [Online]. Available: http://library.epfl.ch/theses/?nr=3199

[21] C. Mattiussi and D. Floreano, "Analog Genetic Encoding for the Evolution of Circuits and Networks," *IEEE Transactions on Evolutionary Computation*, vol. to appear, 2007.

[22] P. Dürr, C. Mattiussi, and D. Floreano, "Neuroevolution with Analog Genetic Encoding," in *PPSN 2006*, vol. 9, 2006, pp. 671–680. [Online]. Available: http://ppsn2006.raunvis.hi.is/

[23] D. Marbach, C. Mattiussi, and D. Floreano, "Bio-mimetic Evolutionary Reverse Engineering of Genetic Regulatory Networks," in *EvoBIO: Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, vol. LNCS 4447, 2007, pp. 155–165.

[24] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, September 1999.

[25] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, May 2002.

[26] M. Hammer, "An identified neuron mediates the unconditioned stimulus in associative olfactory learning in honeybees," *Nature*, vol. 366, pp. 59–63, November 1993.

[27] R. Menzel, "Memory dynamics in the honeybee," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioural Physiology*, vol. 185, pp. 323–340, 1999.

[28] ——, "Searching for the Memory Trace in a Mini-Brain, the Honeybee," *Learning and Memory*, vol. 8, pp. 53–62, 2001.

[29] R. Menzel and M. Giurfa, "Cognitive architecture of a mini-brain: the honeybee," *Trends in Cognitive Sciences*, vol. 5, no. 2, pp. 62–71, February 2001.

[30] T. Bäck, D. B. Fogel, and Z. Michalevicz, Eds., *Handbook of Evolutionary Computation*. Oxford: Oxford University Press, 1997.