View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Infoscience - École polytechnique fédérale de Lausanne

Novelty of Behaviour as a Basis for the Neuro-evolution of Operant Reward Learning

Andrea Soltoggio
Laboratory of Intelligent Systems
EPFL
Lausanne, Switzerland
andrea.soltoggio@epfl.ch

Ben Jones
School of Computer Science
University of Birmingham
Birmingham, B15 2TT, UK
b.h.jones@cs.bham.ac.uk

ABSTRACT

An agent that deviates from a usual or previous course of action can be said to display novel or varying behaviour. Novelty of behaviour can be seen as the result of real or apparent randomness in decision making, which prevents an agent from repeating exactly past choices. In this paper, novelty of behaviour is considered as an evolutionary precursor of the exploring skill in reward learning, and conservative behaviour as the precursor of exploitation. Novelty of behaviour in neural control is hypothesised to be an important factor in the neuro-evolution of operant reward learning. Agents capable of varying behaviour, as opposed to conservative, when exposed to reward stimuli appear to acquire on a faster evolutionary scale the meaning and use of such reward information. The hypothesis is validated by comparing the performance during evolution in two environments that either favour or are neutral to novelty. Following these findings, we suggest that neuro-evolution of operant reward learning is fostered by environments where behavioural novelty is intrinsically beneficial, i.e. where varying or exploring behaviour is associated with low risk.

Categories and Subject Descriptors

I.2.6 [**Learning**]: [Connectionism and Neural Nets]

General Terms

Algorithms

Keywords

Neuro-evolution, Adaptation, Learning, Neuromodulation, Artificial Life

1. INTRODUCTION

The actions of a neuro-controlled agent, either simulated or robotic, are driven by the outputs of a neural network [17, 11]. The outputs serve to control wheels, legs or other actuators whose movements result in certain actions within the environment. Although optimality in control is a coveted feature, there are situations where optimal control policies are not defined, for example in a new environment, or when the outcome of certain actions cannot be known in advance [10]. When facing an uncertain or new situation, an agent can either apply a default action, or alternatively perform a random action: both strategy are potentially equivalent. Situations of this kind are not infrequent: even in simple navigation tasks, the encounter of an obstacle presents a robot with different possibilities like passing the obstacle on the left side, or on the right side, but other options might also exist. If an agent performs constantly the same action, it can be said to display little novelty of behaviour; on the contrary, an agent that changes its pattern of actions can be said to display novelty of behaviour. When an obstacle is encountered for the first time, it is generally not known which sequence of actions would be more beneficial: a random or novel action would be on average as good as a fixed policy. Therefore the notion of novelty of behaviour [9] is not related to any measure of performance.

A different case is when a certain condition repeats itself—for example a robot returns to walk a previously encountered path. The outcome of the actions performed previously could be taken into account and used to improve the current performance, therefore displaying a learning behaviour. For example, a mechanism could reinforce beneficial actions, thereby making them more probable in the future, or making harmful actions less probable. The ability of adjusting behaviour according to the environmental feedback is a fundamental aspect of animal learning [8, 5]. Such skill is named *instrumental* or *operant conditioning*, and can be observed when an animal in a previously observed environment repeats with higher probability those actions that led to a reward, whereas those actions that led to punishment are progressively abandoned. Mathematical models of operant reward learning[1] have become popular in machine learning under the name of *reinforcement learning* [16]. In such frameworks, the task of an agent is to maximise the total reward collected in a lifetime. To do so, the agent may first adopt a predominantly exploratory behaviour, observing the rewards that derive from different actions. Over time the agent adjusts its strategy to perform those actions that have been discovered to lead to higher reward. In reward-based environments where decision making influences future reward, an important parameter is the trade-off between exploration and exploitation. Successful strategies, including

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
Copyright 2009 ACM 978-1-60558-325-9/09/07 ...$5.00.

[1] Another terminology to indicate operant or instrumental conditioning.

human decision making [4, 6], are based on a compromise between exploitation (to collect a safe, known amount of reward) and exploration (to sample the outcome of other actions). In some cases, for example in the presence of a new environment, or in developmental and learning phases in animals, a higher level of exploration can be adopted initially and then decreased with maturation. The balance between exploration and exploitation also depends on the speed of the change in environmental conditions.

In the interpretation of the authors, novelty of behaviour in an open-loop fashion[2], as outlined in the first paragraph, does not coincide with exploratory behaviour in the context of operant reward learning. This is because maintaining or changing a certain course of action—without considering feedback from the environment—are policies that do not involve learning or environmental adaptation. According to a common language definition, exploration is *"the action of travelling in or through an unfamiliar area in order to learn about it"* [1], which is more than novelty of behaviour. In this sense, learning is the driving reason of exploratory behaviour. Similarly, exploitation means to *"make full use of and derive benefit from a resource"* [1], therefore implying that the memorised value of a resource causes it to be exploited. It transpires that both terms exploration and exploitation imply the existence of reward and evaluative feedback as driving principles. It follows that *exploration* can be seen as the combination of *novelty* plus *learning*, and *exploitation* as *conservative* behaviour plus *memory.* In allowing for such a scheme, a reasonable assumption is that operant reward learning is built from components such as the ability to vary the pattern of actions (when exploring), preserving the pattern of actions (when exploiting), and finally alternating between both according to learnt and memorised reward cues. Thus, in the framework of neuro-evolution, operant reward learning is achieved if all those components can be evolved into a neural controller. In reinforcement learning algorithms, it is often assumed that a set of actions is available to an agent, and that the agent is capable of performing either exploration, by selecting random actions, or exploitation, as pre-built 'given' skills. Such is not the case in neural control where the skills of maintaining or changing the course of actions requires certain neural structures. Thus, a question is to what extent the evolution of operant reward learning depends on the underlying skills of preserving or changing the current behaviour. The hypothesis tested in this study is that novelty of behaviour and the environments where novelty is beneficial constitute an advantageous basis for the neuro-evolution of operant reward learning.

In an initial phase, two versions of a T-maze environment [3] are used to evolve either conservative or novel behaviours. Operant reward learning is evolved in a second phase by introducing reward items in the environments. The findings indicate that novelty of behaviour is a fundamental skill that increases the speed in neuro-evolution of operant reward learning. The agents that are capable of novel behaviour, despite being initially non-sensitive to reward stimuli, and despite being as exposed to reward stimuli as the conservative agents, display a significant evolutionary advantage when suddenly immersed in reward-based dynamic environments. From these findings, we conclude that the capability of displaying novel behaviour, even when not related to learning, is an advantageous basis for the subsequent evolution of sensitivity to, and meaningful use of, reward.

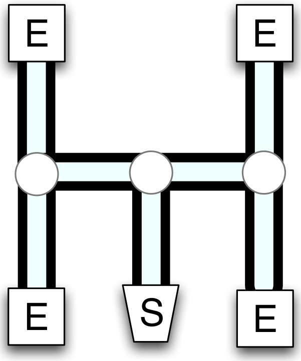

Figure 1: A double T-maze. The position S is Start, E are the locations where the maze-ends. Each turning point where a decision needs to be taken is marked by a circle.

The rest of the paper is organised as follows. In section 2, the environments where novel and conservative behaviours were initially evolved are described. It is then explained how reward items are introduced in the second phase. Section 3 illustrates the evolutionary settings for the evolution of the neuro-controllers. In section 4, the results show that evolution of reward-seeking behaviour for the novelty agents is faster than for the conservative agents. Section 5 concludes the paper outlining the main message of this research.

2. ENVIRONMENTS

A T-maze environment is described hereafter. Different fitness functions described in the following subsections are employed to evolve different behaviours and analyse the performance during evolution.

2.1 Navigation in T-mazes: Neutral and Novelty-Advantageous Environments

In a T-maze environment, an agent navigates a corridor, at the end of which is a T-turn that splits the corridor into two branches, one going to the left and one going to the right. At the T-turn, an agent needs to decide whether to turn left or right. More T-turns can be presented in sequence, thereby generating an n-branched T-maze. Figure 1 illustrates a double T-maze that includes two sequential turning points and four possible maze-ends. In a simple scenario, an agent leaves the start location (S) and navigates the maze until it reaches any of the maze-ends (E). Subsequently, the agent is re-positioned at the start location and commences another trial for a number of times. The sensory-motor signals considered for this navigation task were devised as the minimal set required to express the decision making processes that need to be undertaken during navigation. Graphical representations of input-output sensors and signals are given in Figures 2 and 3. The agent can move only forwards, and can perform three actions: turn left, go straight, or turn right. Corridors require forward motion: turns performed along corridors result in the agent crashing. Similarly, turn-

[2] I.e. when the behaviour does not adjust according to reward-feedback from the environment.

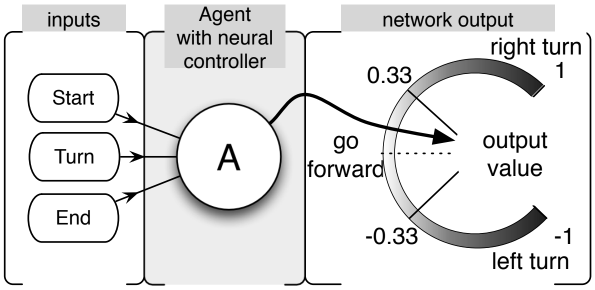

Figure 2: Inputs and output of the neural controller. The *Turn* signal is high (value 1) at turning points, and low otherwise (value 0). The *Start* signal is high (value 1) when the agent is at the start location, low otherwise (value 0). The End signal is high when the agent reaches a maze-end in the maze, low otherwise. The activation of one output neuron, in the range (-1,1), determines the navigation direction: if the output is greater than 1/3, the agent turns left or right according to the sign. If the activation of the output neuron is less than 1/3 in absolute value, the agent proceeds in a straight direction. The input signals, output signal and hidden-neuron signals are affected by 4% uniform random noise.

ing points require either a left or a right turn; straight motion at a turning point results in a crash. Such a simple representation was devised to avoid hidden, emergent memory features from the interaction between the agent and the environment.

The lifetime of an agent is based on a fixed number of departures (200) from the starting point. Each departure is called a trial. Two hundred trials were given to an agent's lifetime. The fitness function counted the number of times the agent reached a maze-end without crashing, collecting a fitness of 0.2 each time a maze-end was reached. In the case of a crash, the agent is re-positioned at the starting point. The fitness was maximised when a correct navigation without any crashing was achieved, and it was independent of any specific turning direction taken by the agent. Therefore the fitness is not affected if the agent continues to visit the same maze-end or changes it from trial to trial. In this respect, the fitness is neutral to both novel and conservative behaviours. Let us call this case the novelty-neutral case (NOV-NE).

In a second case, a novelty-advantageous environment, inspired by the concept of novelty search in [9], is created by introducing a depletion mechanism for the bonus collected at the maze-end. Immediately after a visit to a maze-end, the bonus associated with it is depleted by a certain amount. A number of trials are required to *recharge* it. Considering a maze-end k, the bonus $b_k(t)$ at trial t is given according to the rule

$$\begin{cases} b_k(t) = MaxBonus - d_k(t) & \\ d_k(0) = 0 & \\ d_k(t+1) = 0.5 \cdot d_k(t) & \text{if ending point is not visited} \\ d_k(t+1) = 0.5 \cdot MaxBonus & \text{if ending point is visited} \end{cases}$$

where $d_k(t)$ is a depletion quantity that is assigned $0.5 \cdot MaxBonus$ when an agent visits the maze-end k, and decreases exponentially, therefore regenerating the bonus when

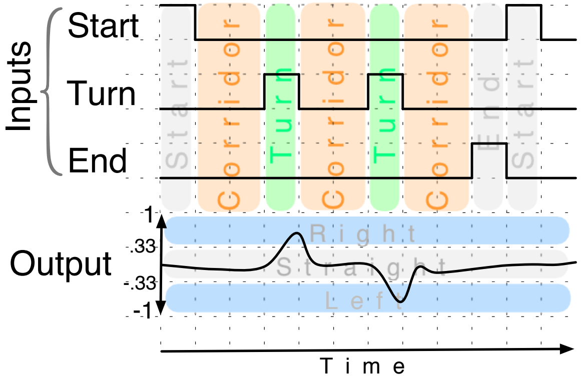

Figure 3: Example of Input/Output signals for a correct navigation of the agent. The output signal remains less than 0.33 in absolute value when navigating the corridors, and indicates a right or left turn when the *turn* signal is high.

the agent visits another maze-end. $MaxBonus$ was set to 0.2. Accordingly, when the maze-end k is visited for the first time, the agent receives a bonus of 0.2, but in consequence of that visit, the bonus drops to 0.1. Each trial in which the ending k remains unexploited, the bonus regenerates to 0.15, then to 0.175, and so on approaching $MaxBonus$.

Agents having the tendency to change their navigation patterns will gain on average a higher fitness than those that visit the same maze-end. In this way, novelty of behaviour is elicited by the fitness function. Let us call this case the novelty-advantageous environment (NOV-AD).

To summarise, in the neutral (NOV-NE) environment, the visiting of different maze-end is risk-free, but it does not elicit any advantage either. Agents collect the same fitness provided that they navigate correctly to a maze-end without crashing. On the contrary, in the NOV-AD scenario, a changing navigation pattern provides an advantage in fitness. The purpose of these two environments is to evolve agents that display predominantly changing navigation patterns (in NOV-AD), or that display no bias towards stable or variable navigation patterns (in NOV-NE).

2.2 Introducing Reward Items

In more complex, reward-based scenarios, agents are exposed to reward stimuli, such as food items, the locations of which change with time. In these problems, it is assumed that the total number of reward items collected needs to be maximised [10]. The T-maze described above can be modified to contain reward items. These will be located at the maze-ends (E) in the maze. One high reward item (value 1) is located at one of the maze-ends, and low reward items (value 0.2) are placed in the remaining three maze-ends. The agents are now equipped with an additional 'reward' sensor that returns the amount of reward collected when reaching a maze-end. The fitness of each agent now corresponds to the total amount of reward collected during a lifetime; therefore, the agents undergo selection pressure to maximise the total reward intake. In this new condition, the optimal strategy involves an initial search phase (or explorative phase) to identify the location of the high reward, and an exploiting phase, when the location of the high reward has been identified, consisting of a repeated navigation to the known maze-end. This capability of modifying a behavioural response according to reward information in the environment manifests itself as observable operant reward learning. At the behavioural level, operant reward learning represents a skill that makes a profitable use of exploratory and exploiting behaviours.

An important environmental feature also introduced here is that the location of the high reward is not kept fixed, but is varied from time to time, and precisely every 50±15 trials. If the location of the reward was kept fixed, such information would be encoded into the genotype throughout the generations, and the agents would adopt a fixed strategy. Instead, the high reward can be initially placed at random at any of the maze-ends, and after a certain number of trials moved to another random maze-end. In this dynamic environment, an agent maximises the reward intake only if is capable of detecting reward items, and alternates between exploration and exploitation according to this information, exploring when the high reward cannot be found, and exploiting a specific maze-end once the high reward has been found. Let us call this a *reward-based dynamic environment* (RBD). It is important to note that in this environment, a random sequence of choices results in the same fitness as a fixed sequence: therefore the RBD environment is novelty-neutral and can also be labelled NOV-NE-RBD.

3. NEURO-EVOLUTION OF CONTROLLERS

3.1 The Neural Model

For the experiments in this paper, a simple discrete time neural model was used, where the activation value a_i of a neuron i at time t is given by:

$$a_i(t) = \sum \Big(w_{ji} \cdot o_j(t-1) \Big) + a_i^b \quad , \qquad (1)$$

where a_i^b is a bias value, w_{ji} is the connection weight from neuron j to i, and o_j is the output of a neuron j given by

$$o_j(t) = tanh(a_j(t)) + n \quad , \qquad (2)$$

where n is a uniform 4% noise. Equations 1 and 2 were applied for three steps for each update of the inputs and output to allow the input signals to propagate through the network. At the end of the three steps, the output of the network was sampled to observe an action as explained in Figures 2 and 3. Each maze location, i.e. start, corridors, maze-end, was presented to the agent for one input-output step.

With this model, learning and memory, as required by the environment above (Section 2.2), can be implemented by means of recurrent connections that allow for certain nodes in the network to retain information about prior activations. Adaptation and memory can also be implemented by using plasticity mechanisms that change the connection weights among nodes. Which approach leads to better evolution of adaptive behaviour is still debated [7, 15]. Here, networks are endowed with the possibility of enabling, by means of evolution, a general plasticity rule given by [14]:

$$\Delta w_{ji} = \eta \cdot [A o_j o_i + B o_j + C o_i + D] \quad , \qquad (3)$$

where A,B,C,D and η are evolvable parameters. This plasticity rule combines a correlation mechanism (term A), a presynaptic mechanism (term B), a postsynaptic mechanism (C) and a decay term (D). By tuning these parameters, it is possible to select a plasticity rule from a large variety of mechanisms. This rule has been used in a number of studies on plasticity mechanisms for the neuro-evolution of learning and memory in simple problems [10, 14, 13]. With the above settings, neural controllers can implement learning and memory by means of recurrent connections, plastic weights or a combination of the two. A certain combination is likely to emerge if it has a selective advantages.

3.2 The Evolutionary Algorithm

A basic Evolution Strategy [2] was employed here to evolve neural networks with unconstrained topologies, i.e. with an arbitrary number of neurons and arbitrary connections among themselves. A matrix $(n+l, n)$ of real values, where n was the number of neurons and l the number of inputs, encoded the network weights. The parameters for the plasticity rule A, B, C, D and η were evolved in the range [-1,1] for A-D, and [-100,100] for η. Genes, expressed in the range [-1,1], were transformed into phenotypical values with a cubic function, and then multiplied by 10 to produce weight values in the range [-10,10]: this produced an initial bias towards small values, which can then increase under selection pressure. A Gaussian noise with standard deviation 0.5% on all genes was applied when mapping the genotype into the phenotype to increase the robustness of the solutions. A threshold value of 0.1 and 0.01 was applied respectively to the weights and the parameters in Equation 3: this feature is important to evolve partially connected networks, and to enable or disable terms in the plasticity rule. Insertion and deletion of neurons were applied with probability 0.02, and were implemented by adding or removing a row and a column from the weight matrix: with this feature, the algorithm was able to search neural topologies of networks having a variable number of neurons. A maximum number of neurons was set to 12.

Mutation was performed on all individuals at each generation by applying to all genes a value $m = \pm exp(-200 \cdot u)$, where u is a random number drawn from a uniform distribution in [0,1] [12]. Crossover was not applied.

A spatial tournament selection mechanism was used: the population was placed sequentially on an array, which was divided at each generation into adjacent segments of size 4 (with random offset at each generation). The tournament winner then replaced the neighbours. Given this selection mechanism, individuals spread their genes only linearly over successive generations. This had the effect of preserving the population diversity thereby facilitating the evolution of diverse topological structures. A population of 800 individuals was employed in all experiments.

4. RESULTS

The whole simulation setup was implemented in C++. Multiple parallel runs with different seeds for the random number generator were executed to provide statistical significance.

4.1 Evolving Novelty of Behaviour

Two separate sets of runs were carried out for the two environments NOV-NE and NOV-AD (see section 2.1). For each set, 30 independent runs were executed. Each fitness evaluation was composed of 4 agent's lives. Figure 4 shows

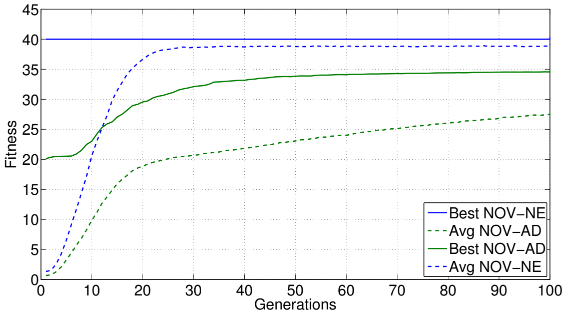

Figure 4: Evolution of navigation skills in the NOV-NE and NOV-AD environments. With the NOV-NE conditions, the maximum fitness is reached immediately, whereas in the NOV-AD environment, the additional requirement of novelty of behaviour resulted in a slower progress in performance. The fitness in NOV-AD does not reach the level in NOV-NE because the bonuses do not recharge completely.

the progress of the best and average fitness values, computed as the median over all 30 independent runs. In the NOV-NE environment, the maximum fitness is reached from the start, whilst the average fitness increases rapidly, indicating that evolution is scarcely necessary because optimal solutions are found in the first random generation. In the NOV-AD environment, evolutionary progress is evident from the progressive increment of both the best and average fitness over successive generations.

The analysis of the solutions indicated that in the NOV-NE environment, one connection weight from the *turn* input to the output is sufficient to reach optimal performance. A negative connection weight means that the agent turns always left, whilst a positive connection weight results in the agent always turning right. This appears to be the successful network topology. It is interesting to note that although changing the turning direction would not decrease the fitness, such solutions were possibly more complex; the simplest solution of having a single connection prevailed. In the NOV-AD environment, a varying turning instruction from the output neuron must be part of the skill-set in order to maximise the fitness. This appears more difficult to evolve: upon inspection, the solutions reveal that the plasticity rules acted on the weights in order to alter the sign of the signal that propagates from the *turn* input to the output. In a well-performing network, the *turn* signal always reaches the output neuron with enough intensity to achieve an output signal higher than 0.33 (in absolute value) in order to perform a turn (see Figures 2 and 3). Evolution in the NOV-AD environment indicated that a mechanism for achieving novel or random behaviour is more complex to evolve than the structure for a reactive, stable behaviour.

Why did evolution not use neural noise to implement randomness in the decision processes? In theory this could be achieved. However, the requirement for the output value states that at turning points the output must be higher than 0.33 in absolute value, lest the robot crashes. A neural noise of 4% is insufficient to achieve this value and impose a turning direction. On ther other hand, it is possible to imagine a workaround with a neural structure that amplifies the neural noise to reach the threshold of 0.33. However, this possible solution appears to be difficult to implement or evolve. Figure 5 shows examples of two networks, one evolved in the NOV-NE environment and the other evolved in the NOV-AD environment for 100 generations. The network from the NOV-AD environment has plastic weights that change during execution as shown in Figure 6. With respect to this model, the evolutionary runs indicated that oscillatory dynamics, changing the weights from the turn input to the output, are suitable solutions to achieve novelty of behaviour in the NOV-AD environment.

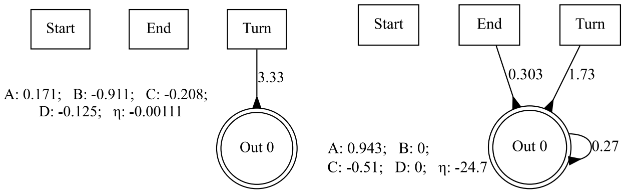

Figure 5: Examples of two networks having conservative behaviour (left) and novel behaviour (right). The values A,B,C,D, and η that determine the plasticity rule of Equation (3) were evolved. The sole weight in the network on the left does not change over time due to the low value of η. In this particular example (left drawing), a fixed positive weight results in this network to turn always right. On the contrary, in the network on the right, the change of weights over time causes a varying behaviour and the alternation of the maze-end visited at each trial. The weight-change over time of these networks is shown in Figure 6.

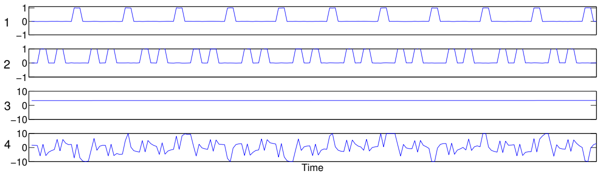

Figure 6: Activities and weights in NOV-NE and NOV-AD networks: Row 1 is the Ending point signal that is high when a maze-end is reached. Row 2 is $Turn$, the signal that becomes high when a turn is encountered. Row 3 is the weight $Turn - Out$ in the network of Figure 5(top). This weight does not change over time, and the agent turns right at all times. Row 4 is the weight $Turn - Out$ for the network in Figure 5(bottom). The weight-change over time results in the agent to choose different turning directions from trial to trial.

4.2 Evolution of Learning

The populations that evolved in the NOV-NE and NOV-AD environments as described in the previous section continued the evolution in a reward-based dynamic environment (RBD). This new environment, as described in section 2.2, presents dynamic reward items that change their location during an agent's lifetime. In this environment the agents are required to change their navigation patterns in order to visit all four maze-ends in a double T-maze, but at the

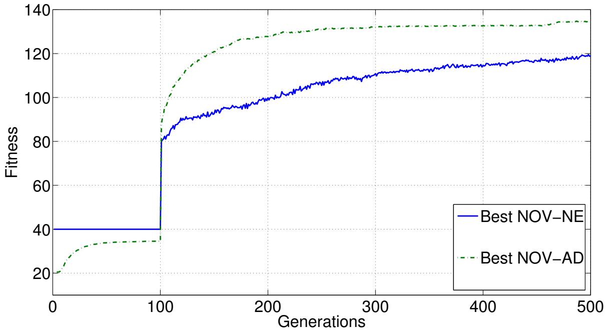

Figure 7: Evolution of operant reward learning for the agents coming from the NOV-NE and NOV-AD environments placed (after generation 100) into the RBD environment. The line represents the median value of the best fitness of 30 independent runs. Once the reward items are introduced at generation 100, and the fitness becomes the total value of reward collected during a lifetime, the agents coming from the NOV-AD environment appeared faster in evolving reward learning skills.

same time they should be able to persist in visiting a given maze-end when that yields a high reward item. A new *reward* input was made available to the networks: initially it was not connected to any inner neurons, and evolution had the task of inserting appropriate connection-weights to reach reward learning. Only by retrieving the signal from the reward input, and using it appropriately, it was possible to maximise the fitness value. Given the four maze-ends, the high reward was placed randomly at one maze-end, and switched to another random location each 50±15 trials. This allowed the placement of the high reward in all four maze-ends across the whole number of 200 trials. Three maze-ends had a low reward of 0.2. The purpose was to evolve agents capable of pursuing the high reward by a combination of exploration and exploitation, therefore manifesting operant reward learning.

Figure 7 shows the fitness progress for the best and average fitness, for the agents coming from the NOV-NE and NOV-AD environments. The two populations were now evolving in identical environments. From the fitness graph, we observe that the agents that were previously trained to perform novelty of behaviour appear to evolve operant reward learning on a faster evolutionary scale than those that were evolved in the NOV-NE environment.

4.3 Eliciting Novelty in a Reward-Based Environment

In the previous experiments, the RBD environment was novelty-neutral. This is a feature that appears sound in an artificial evolutionary environment: if an agent has not acquired any learning skill yet, why would a random action be better than a default, fixed one? Contrary to intuition, and following the results in the previous section, we are encouraged to introduce a novelty advantage into the RBD environment. Accordingly, a random or changing behaviour should be better than a fixed behaviour among those agents that are not capable of learning yet. Whilst the previous RBD environment had high and low reward items with fixed values of 1.0 and 0.2, the newer environment now had a depletion mechanism imposed upon the rewards according to the rule described in section 2.1. Depletion of reward items can also be seen as depletion of food resources in nature when those are continuously exploited.

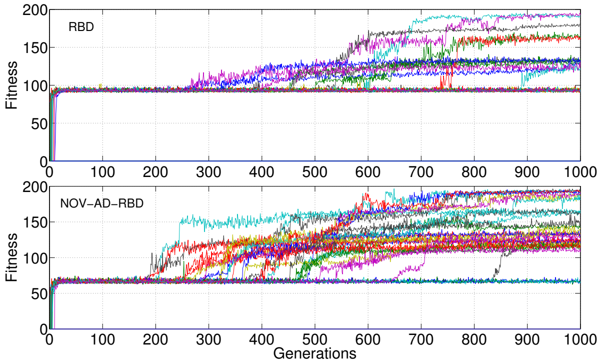

Figure 8: The fitness values of all the 30 runs are shown for agents evolving in the RBD environment (top) and NOV-AD-RBD (bottom). In NOV-AD-RBD (bottom), a reward depletion mechanism was introduced to elicit novelty of behaviour: due to this the fitness for the first hundred generations is lower. Nevertheless, in this second environment, the agents manifest a better evolution of learning as can be observed by a higher number of fitness curves (runs) reaching higher values. In NOV-AD-RBD, the amount of the high reward item is subject to depletion during exploitation, therefore, the amount of the high reward in this environment was set to 1.1 instead of 1.0 to have equal maximum fitness as in RBD.

In an RBD environment with reward depletion, a novelty-inclined agent, which continuously changes the maze-end visited, collects on average more fitness than a conservative agent. However, this is true only when learning has not yet been evolved. Once the agents evolve operant reward learning and are capable of identifying the location of the high reward, the gain derived by exploiting the high reward (valued 1.0 versus 0.2 of the low reward) is much higher than the loss of 0.1 due to depletion caused by continuous exploitation of the high-rewarding source. The RBD environment, when enriched with reward depletion, was named novelty advantageous reward-based dynamic environment (NOV-AD-RBD). The RBD and the NOV-AD-RBD environments can be seen as similar environments where the gain in changing behaviour is increased in the NOV-AD-RBD: yet, changing without learning is less advantageous than learning.

Progression of the evolutionary runs for the two environments are shown in Figure 8. The fitness progress over 30 runs indicates that the operant reward learning evolves better in the NOV-AD-RBD environment, where novelty of behaviour is advantageous. However, it is important to note that an optimal controller performs very limited exploration: over a lifetime of 200 trials, the location of the high reward is changed three times, and it is therefore unknown to the agent on four occasions: at the beginning of a lifetime, and and on three subsequent occasions. Given the four maze-ends, the average optimal number of trials to identify the

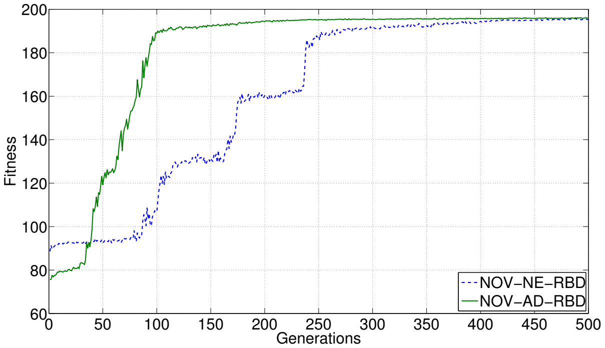

Figure 9: Evolution of operant reward learning in the RBD and NOV-AD-RBD environments when modulatory dynamics are employed. With the enhanced model, a better evolution can be observed with respect to the previous experiments. Despite the better evolution with the enhanced model, the NOV-AD-RBD environment—where reward depletion elicit novelty of behaviour—appeared to facilitate the evolution of operant reward learning.

high reward is 2.5: accordingly, an optimal behaviour, given 4 conditions of unknown reward, spends on average 10 trials out of the 200 to explore, and 190 to exploit the discovered location of the high reward. It would appear then that exploitation is a considerably more important feature than exploration because an optimal agent deploys exploration for a much smaller percentage of its lifetime. Nevertheless, during the evolutionary phases, the NOV-AD-RBD environment appeared to be more favourable for the evolution of fully-fledged operant reward learning.

4.4 Evolution of Learning with Neuromodulation

The evolutionary runs depicted in Figures 7 and 8 showed that optimal controllers able to reach the maximum fitness of 195.2[3] are difficult to evolve. Despite the introduction of the reward depletion mechanism, and the consequent improvement in speed of evolution, the problem proves difficult to solve with the given neural model and search algorithm. A further evolutionary experiment was performed here with the introduction of modulatory neural dynamics as in [14]. The enhanced algorithm has the possibility of inserting into the neural networks modulatory neurons that have the function of modulating the intensity of the plasticity of Equation 3. The weight-change is the value given by Equation 3 times the hyperbolic tangent of the modulatory signals received by a neuron, i.e. $\Delta w = \delta \cdot mod$, where δ is the weight-change from Equation 3 and mod is the modulatory signal received by the postsynaptic neuron. Modulatory neurons can be inserted randomly into evolving networks, as standard neurons do, and can connect to any neuron, therefore affecting the plasticity rates of the target neurons. Such model proved to be remarkably effective in the solution of reward-based dynamic problems [14]. Thus, it is possible to hypothesise that the effectiveness of neuromodulation could cancel or decrease the advantage of novelty of behaviour in the evolution of operant reward learning. To cast light on this point, we reproduced the experiments of the previous section with the environments RBD and NOV-AD-RBD, and evolved networks enhanced with modulatory dynamics. The median fitness over 30 runs for the two cases is shown in Figure 9. From the fitness progress it is evident that the enhanced model allows for a much faster and better evolution of control networks. Nevertheless, the presence of reward-depletion in the NOV-AD-RBD environment bestows the evolving agents with a significant advantage over those agents evolving in the neutral environment.

5. CONCLUSION

Behaviour-maintaining and behaviour-changing capabilities of an agent, here named conservative and novel behaviours, have been suggested to be predecessors of the more advanced exploiting and exploratory behaviours. Whereas both conservative and novel behaviours are seen as non-learning and open-loop behaviours, exploitation and exploration demand the presence of evaluative feedback. It was outlined that neural controllers, in order to perform operant reward learning, need non-trivial mechanisms to maintain a certain sequence of actions during exploitation, and to change that during exploration. Traditionally, these neural mechanisms are not a major subject of investigation in reinforcement learning algorithms where neural implementation of exploration and exploitation are not often discussed.

This paper introduced environments that favour novelty of behaviour in order to investigate its effect in the evolution of reward learning, or operant reward learning. Interestingly, the notion of behavioural novelty has been introduced recently in [9], where novelty of behaviour was used as a surrogate of fitness, and was applied to a population of solutions to evolve diverse behaviour: diversity of behaviour resulted in better performance than a fitness-driven evolutionary search in ill-behaved fitness landscapes. That study outlined that the search for novelty resulted in a better coverage of a number of possible behaviours, some of which were indeed good under the fitness criterion. Here novelty of behaviour was measured not among the members of a population, but for one agent capable of varying its behaviour over time. The hypothesis was that an agent capable of novelty of behaviour was more likely to evolve sensitivity to reward and achieve operant reward learning on a faster evolutionary scale.

An initial experiment outlined that novelty of behaviour is less immediate to evolve than unchanging behaviour because it requires more complex dynamics. Starting from those solutions, the evolutionary process continued in a reward-based environment: here the networks that were capable of novelty of behaviour had an advantage in evolving evaluative reward learning, or operant reward learning. This indicated that the capability of varying behaviour, although initially performed without feedback from the environment, is a beneficial skill to evolve learning and acquire the meaning and use of reward stimuli.

It is important to note that the two initial environments (NOV-NE and NOV-AD) do not have reward items: therefore both of them are equally similar to (or dissimilar from) the maze where the fitness is based on reward collection. Moreover, both conservative and novelty agents were exposed to reward stimuli in equal amount once immersed in

[3] According to the number of trials spent to find the high reward, the loss in reward is 0, 0.8, 1.6 or 2.4. On average the loss is 1.2 for each optimal exploration. Over 4 exploratory phases in a lifetime, the loss is 4.8.

the reward-based dynamic environment (RBD). If a comparison of similarity is attempted at the behavioural level, the RBD environment would possibly look more similar to the novelty neutral environment (NOV-NE) because exploration is required for very limited amount of time (about 5%). It is therefore surprising that the agents that change continuously their navigation patterns (those from NOV-AD) evolve back to mostly conservative behaviour (required in RBD) on a faster scale than the conservative agents from NOV-NE evolve to perform a small percentage of exploration. A possible explanation is that the RBD environment, although novelty-neutral, requires the capability of turning both to the left and to the right, as in the NOV-AD precursory environment, whereas agents in the NOV-NE can evolve to perform only one action. Therefore, one can conclude that the difficulty is to evolve the essential skill of performing different actions, even on a random basis, and that this skill must be encouraged to facilitate the evolution of reward learning.

A mechanism that favoured novelty of behaviour was then introduced into a reward-based environment. This was done by allowing a depletion of reward-items in proportion to the level at which they were exploited. It must be noted that the advantage that derived from changing reward-source was considerably less than the advantage that derived from exploiting the high-rewarding maze-end. In other words, reward learning was more advantageous than open-loop novel behaviour. Surprisingly, by introducing this contrasting objective in the fitness function, the experiments showed better performance on the evolutionary scale for those agents evolving in a novelty advantageous environment. Such advantage was observed in two cases with different neural models: a model with a traditional plasticity rule, and a more advanced model with neuromodulated plasticity, therefore supporting the generality of the conclusion. Both environments where novelty was advantageous appeared to increase the speed of evolution of learning.

The experimental findings in this study suggest that the neuro-evolution of reward-based learning benefits from environments where on average exploration has low risk, or it is advantageous. Before a reward-driven behaviour can proliferate in evolving solutions, the disposition to manifest varying behaviour in an open-loop, non-learning fashion, appeared a beneficial factor. This conclusion opens new perspectives on the role of random or novel behaviour in the neuro-evolution of reward-based learning skills.

6. ACKNOWLEDGMENTS

Special thanks to Joel Lehman and Kenneth Stanley for our discussions on the concept of Search for Novelty.

7. REFERENCES

[1] *Oxford English Dictionary.* Oxford : Oxford University Press, 1989.

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

[3] J. Blynel and D. Floreano. Exploring the T-Maze: Evolving Learning-Like Robot Behaviors Using CTRNNs. In *EvoWorkshops*, pages 593–604, 2003.

[4] R. Bogaz. Optimal decision-making theories: linking neurobiology with behaviour. *Trends in Cognitive Sciences*, 11:118–125, 2006.

[5] Britannica. Animal learning. Encyclopedia Britannica 2007 Ultimate Reference Suite, 2007.

[6] N. D. Daw, J. P. O'Doherty, P. Dayan, B. Seymour, and R. J. Dolan. Cortical substrates for exploratory decisions in humans. *Nature*, 441(15):876–879, June 2006.

[7] D. Floreano and J. Urzelai. Evolution of plastic control networks. *Auton. Robots*, 11(3):311–317, 2001.

[8] C. R. Gallistel. *The Organization of Learning.* MIT Press, 1993.

[9] J. Lehman and K. O. Stanley. Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In *Proceedings of the Eleventh International Conference on Artificial Life (ALIFE XI) Cambridge MA: MIT Press*, 2008.

[10] 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*, 10(1):5–24, 2002.

[11] D. T. Pham and X. Liu. *Neural Networks for Identification, Prediction and Control.* Springer, London, 1995.

[12] J. E. Rowe and D. Hidovic. An evolution strategy using a continuous version of the gray-code neighbourhood distribution. In *GECCO (1)*, pages 725–736, 2004.

[13] A. Soltoggio. Neural Plasticity and Minimal Topologies for Reward-based Learning Problems. In *Proceeding of the 8th International Conference on Hybrid Intelligent Systems (HIS2008), 10-12 September, Barcelona, Spain*, 2008.

[14] A. Soltoggio, J. A. Bullinaria, C. Mattiussi, P. Dürr, and D. Floreano. Evolutionary Advantages of Neuromodulated Plasticity in Dynamic, Reward-based Scenarios. In *Proceedings of the Artificial Life XI Conference 2008. MIT Press.*, 2008.

[15] K. O. Stanley and R. Miikkulainen. Evolving adaptive neural networks with and without adaptive synapses. In B. Rylander, editor, *Genetic and Evolutionary Computation Conference Late Breaking Papers*, pages 275–282, Chicago, USA, 12–16July 2003.

[16] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, USA, 1998.

[17] A. M. S. Zalzala and A. S. Morris, editors. *Neural Networks for Robotic Control: Theory and Applications.* Ellis Horwood, London, 1996.