

Evolving Childhood's Length and Learning Parameters in an Intrinsically Motivated Reinforcement Learning Robot

Massimiliano Schembri Marco Mirolli Gianluca Baldassarre

Laboratory of Autonomous Robotics and Artificial Life,
Istituto di Scienze e Tecnologie della Cognizione,
Consiglio Nazionale delle Ricerche (LARAL-ISTC-CNR),
Via San Martino della Battaglia 44, I-00185 Roma, Italy
{massimiliano.schembri, marco.mirolli, gianluca.baldassarre}@istc.cnr.it

Abstract

The capacity of re-using previously acquired skills can greatly enhance robots' learning speed and behavioral complexity. 'Intrinsically Motivated Reinforcement Learning (IMRL)' is a framework that exploits this idea and proposes to build agents capable of solving several specific tasks by assembling general-purpose building-block behaviors ('skills') previously acquired on the basis of 'intrinsic motivations'. This paper proposes a novel neural-network hierarchical reinforcement-learning architecture which exploits 'evolutionary robotics (ER)' techniques that not only allow tackling important limits of IMRL, as shown in previous papers, but they also allow investigating two other important issues, namely: (1) the optimization of the parameters that regulate the architecture's learning processes; (2) the optimization of the time the architecture dedicates to the acquisition of the skills' repertoire. These two issues are investigated here through a simulated robot engaged in solving compositional path-following navigation tasks. The main results obtained indicate that the proposed approach allows obtaining a remarkable improvement of performance of the architecture, while at the same time decreasing the time the system needs to learn the skills ('childhood'), with respect to cases where hand-tuned parameters are used.

1. Introduction

Current robots are typically directly programmed to solve just one task at a time in one environment. This makes them severely limited in that they cannot cope with any other task nor with other kinds of

environments. Recently, in both the machine learning and the developmental/epigenetic robotics communities, a number of proposals have been put forward for solving such limitations by relying on autonomous robot development (Kaplan and Oudeyer, 2003; Schmidhuber, 1991; Weng et al., 2001; Huang and Weng, 2002; Marshall et al., 2004; Oudeyer et al., 2007). The basic idea behind such proposals is to endow robots with developmental programs which allow them to learn, through an autonomous interaction with the environment, general-purpose building-block behaviors which might successively be 'assembled' to tackle several specific tasks.

One of the most promising frameworks that has been proposed to this purpose is 'Intrinsically Motivated Reinforcement Learning (IMRL)' (Barto et al., 2003; Stout et al., 2005) IMRL is based on the idea that natural organisms, especially the most sophisticated ones like humans and primates, are not driven only by basic *extrinsic* motivations directly related to survival (e.g. for eating, drinking, avoiding predation and mating), but also by *intrinsic* motivations which drive them to accomplish exploratory behaviors directed to acquire skills and knowledge (White, 1959; Berlyne, 1960). The adaptive value of these behaviors — and of the motivations behind them — resides in that they permit the acquisition of general-purpose skills which can be used, when needed, for accomplishing a number of different tasks directly related to survival and reproduction.

Notwithstanding its undeniable appeal, at present the IMRL framework has two important drawbacks. First, as they rely on the reinforcement learning framework of 'options' (Sutton and Singh, 1999), current implementations of IMRL assume high-level abstract representations of states and actions, and hence they can be applied only to agents acting in abstract simple grid-world environments. As also recognized by Barto and coworkers (Stout et al., 2005),

this is a serious limit and it is not clear whether and how IMRL might be used in robotic scenarios. Second, the ‘salient events’ which initiate and drive the development of basic skills must be explicitly specified by the programmer: this requires the introduction of a significant amount of assumptions about the tasks at hand and their possible solutions, thus considerably reducing both the generality of the approach (for each problem the appropriate salient events must be specified) and agent’s autonomy.

Recently (Schembri et al., 2007b) we have proposed an architecture, based on a hierarchical actor-critic reinforcement-learning model (Sutton and Barto, 1998), which overcomes both these limits of IMRL by integrating it with ‘Evolutionary Robotics (ER)’ framework (Nolfi and Floreano, 1999). The architecture, described in detail in Sec. 2.2, is formed by a number of ‘experts’, which learn basic skills, and a ‘selector’, which learns to select the expert which is most appropriate for the current situation (cf. Baldassarre, 2002). In contrast to the IMRL implementations proposed so far, the use of neural-networks allows the architecture to be applicable to continuous and noisy environments typical of robotic tasks. Furthermore, the architecture acquires basic skills on the basis of evolved ‘reinforcers’, that is neural networks that assign a reward value to explored states, instead of hardwired salient events, thus significantly enhancing both the generality of the approach and the overall autonomy of the system. The model is able to acquire general-purpose skills on the basis of intrinsic evolved motivations during a ‘childhood’ phase, and to solve several different robotic tasks by combining such skills during a successive ‘adulthood’ phase.

In a second work, (Schembri et al., 2007a) we compared the performance of the architecture with other systems in which various components of the architecture are either trained during lifetime or evolved through a genetic algorithm. The results were quite encouraging: the versions of the architecture using both evolution and learning significantly outperformed the versions using either one of the two. Furthermore, among the systems using both evolution and learning, the one evolving internal reinforcers driving the acquisition of building-block skills had a higher evolvability than those directly evolving the related behaviors.

The present work tries to push the idea of exploiting ER techniques for optimizing the learning capabilities of an intrinsically motivated robot even further. Any reinforcement learning architecture has a number of parameters that regulate its learning processes. Typically, the values of these parameters are decided by the programmer according to intuitive heuristics and non-systematic trial-and-error optimization processes. The use of ER opens up the

possibility of using a genetic algorithm for finding optimal sets of the parameters regulating reinforcement learning processes (to the best of the authors’ knowledge, the only work that exploited this idea is Eriksson et al., 2003).

Furthermore, as the architecture studied here assumes that the robot’s life is divided in a childhood and an adulthood phase, there is another fundamental parameter which in the previous two works (Schembri et al., 2007a,b) was set by trial-and-error processes and hence which could be optimized through the genetic algorithm: the length of childhood, that is the number of steps during which the robot trains its experts on the basis of intrinsic motivations. Both for robots and for real organisms, there is clearly a trade-off between short and long childhood phases. If childhood is too short, an agent cannot learn enough, and all of its basic abilities can only be genetically inherited. On the other hand, childhood has clear costs: for an organism, it is time during which the organism is not autonomous and must be fed and protected by its parents; for a robot, it is time which is not spent for solving the tasks the robot has been designed for. In order to test whether our system could be further optimized, in this paper we use the genetic algorithm not only to evolve the reinforcers driving the acquisition of basic skills, but also the length of the childhood length and the learning parameters of the reinforcement learning algorithm.

The rest of the paper is organized as follows. Sec. 2.1 describes the robotic setup and the simulated experiment used to test the model. Sec. 2.2 contains a detailed description of the model. Sec. 2.3 describes the used genetic algorithm. Sec. 3. reports the main results. Finally, Sec. 4. concludes the paper.

2. The setup

2.1 The simulated environment and robot

The simulated robot is a mobile robot with a 30 cm diameter equipped with a camera assumed to look at a portion of the ground measuring 24×8 cm located just in front of the robot. In each cycle the robot’s input is furnished by a vector \mathbf{x} of $12 \times 3 = 36$ binary values that corresponds to the activation of the RGB receptors sampling the camera’s image on the vertex of a 6×2 regular grid (see Fig. 2). The robot’s motor system is a ‘wheelchair’ and is driven by setting the orientation variation within $[-30, +30]$ deg, and the translation speed within $[0, 2]$ cm.

The environment is a square walled arena with a regularly textured floor (Fig. 1). As mentioned above, the robot’s life is divided into two phases: ‘childhood’ and ‘adulthood’. During childhood the robot learns a set of basic sensory-motor skills based on its intrinsic motivational system. Childhood’s

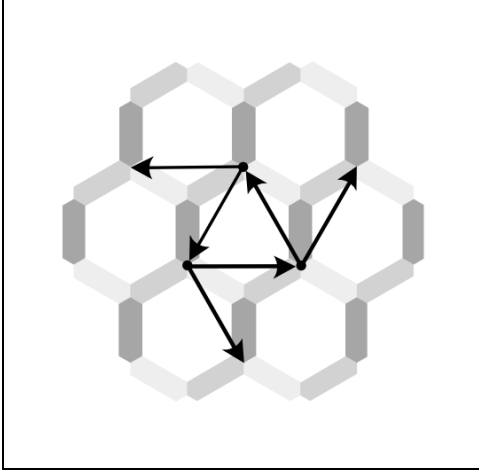


Figure 1: The environment and the six ‘adulthood’ tasks. The sides of the hexagons are colored with blue (dark gray), red (gray) and green (light gray). Arrows represent the different tasks: each arrow’s tail and head indicate, respectively, the starting and the target position of a task.

length is particularly important in this work as in some experiments its length was evolved and how this affected the parameters was studied (see Sec. 2.3). During adulthood, the robot learns to combine the acquired skills in order to accomplish six different tasks. In each of these tasks the robot has to reach a given target location (having a 26cm diameter) starting from a particular starting position (see Fig. 1). During each task, every time the robot reaches the target it receives a reward and is placed back at the starting position.

2.2 The model

The controller of the robot is a hierarchical modular neural network (Fig. 2) formed by a selector and a number of experts. The selector and each expert are formed by a neural-network implementation of the actor-critic model (Sutton and Barto, 1998). This model is composed of two neural components, an ‘actor’ and a ‘critic’, and is capable of learning to select appropriate actions in order to maximize the sum of the future discounted rewards (‘discounted’ means that the same reward is given less importance if received later in time, see below). The actor learns to associate suitable actions with the perceived states of the environment on the basis of the critic’s evaluation. The critic learns to associate evaluations with single visited states on the basis of the rewards experienced after these visits, and produces a one-step judgment of the actor’s actions on the basis of the evaluations of couples of states visited in sequence.

The experts are now described in detail. Each expert e is formed by three components: a ‘reinforcer’, an ‘actor’ and a ‘critic’. The reinforcer is a 2-layer

neural network that with its 1×36 vector of weights \mathbf{w}_e^r maps the retina activation \mathbf{x}_t at time t to the activation of a sigmoid unit that ranges in $[-1, +1]$ and encodes the expert’s reward r_{et} (note that in the paper the symbols at the exponent do not represent indexes but qualify the main symbol):

$$r_{et} = 2 \cdot \sigma[\mathbf{w}_e^r \mathbf{x}_t] - 1 \quad (1)$$

where $\sigma[\cdot]$ is the sigmoid function. Note that \mathbf{w}_e^r are evolved, as illustrated in Sec. 2.3.

An expert’s actor is a 2-layer neural network that with its 2×36 matrix of weights \mathbf{w}_e^a maps the retina activation to two sigmoid units \mathbf{m}_e :

$$\mathbf{m}_{et} = \sigma[\mathbf{w}_e^a \mathbf{x}_t] \quad (2)$$

In order to obtain the performed actions (cf. Mannella and Baldassarre, 2007), the activation of the two units is added a Gaussian noise to obtain two values \mathbf{a}_{et} ranging within $[0, +1]$ (noise values are redrawn until the values respect this range):

$$\mathbf{a}_{et} = \mathbf{m}_{et} + \epsilon[0, \rho] \quad (3)$$

where $\epsilon[0, \rho]$ is a Gaussian noise with zero mean and standard deviation ρ initially set to 0.3 and linearly reduced to zero during childhood. The values \mathbf{a}_{et} are then mapped onto the orientation-variation and translation commands issued to the motor system.

The weights of the actor are updated using the following formula:

$$\Delta \mathbf{w}_e^a = \eta^{ae} \cdot s_{et} (\mathbf{a}_{et-1} - \mathbf{m}_{et-1}) \cdot \sigma'[\mathbf{w}_e^a \mathbf{x}_{t-1}] \mathbf{x}_{t-1} \quad (4)$$

where η^{ae} is the learning rate of the experts’ actors, s_{et} is the expert’s critic surprise (see below) and $\sigma'[\cdot]$ is the derivative of the sigmoid function. The effect of this learning rule is to lead the means \mathbf{m}_{et-1} of actions toward their noisy values \mathbf{a}_{et-1} if $s_{et} > 0$ and away from them if $s_{et} < 0$.

Note that the experts’ actor and critic are trained only during childhood, while in adulthood the experts skills are fixed and are recombined by the selector in order to achieve ‘externally’ rewarded goals.

The experts’ critic is mainly formed by an ‘evaluator’ which is a 2-layer neural network that with its 1×36 vector of weights \mathbf{w}_e^v maps the retina activation to a linear output unit encoding the expert’s evaluation v_{et} of the perceived state:

$$v_{et} = \mathbf{w}_e^v \mathbf{x}_t \quad (5)$$

The critic uses the evaluator’s evaluations, together with the reward provided by the expert’s reinforcer, to compute the expert’s surprise s_{et} as follows:

$$s_{et} = (r_{et} + \gamma^e \cdot v_{et}) - v_{et-1} \quad (6)$$

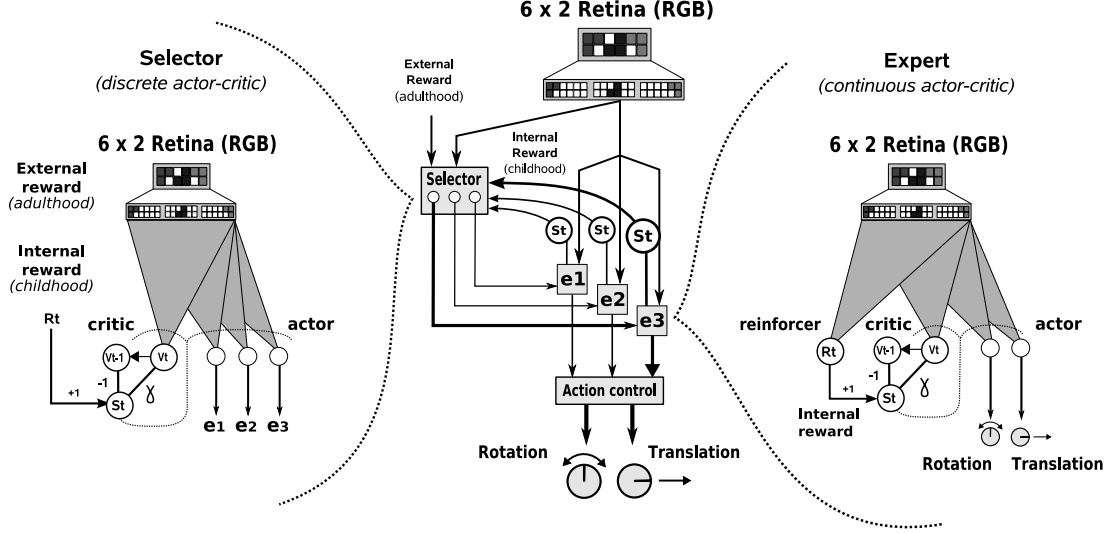


Figure 2: Center: the whole architecture. Left: the selector’s architecture. Right: one expert’s architecture (see text for details).

where γ^e is the experts’ discount coefficient used to weight the future rewards.

The weights of the evaluator are updated, on the basis of the surprise signal, using a Temporal Difference (TD) learning rule (Sutton and Barto, 1998):

$$\Delta \mathbf{w}_e^v = \eta^{ve} \cdot s_{et} \cdot \mathbf{x}_{t-1} \quad (7)$$

where η^{ve} is the learning rate of the experts’ evaluators.

The selector is now described in detail. The selector is formed by two components, the ‘actor’ and the ‘critic’. At each time the actor selects the expert that has the control and trains its actor and evaluator (only during childhood). The actor is a 2-layer neural network that with its 3×36 matrix of weights \mathbf{w}^a maps the retina activation to three (as many as the number of experts) sigmoid output units \mathbf{m} :

$$\mathbf{m}_t = \sigma[\mathbf{w}^a \mathbf{x}_t] \quad (8)$$

These activations are used as pseudo-probabilities to compute the probabilities \mathbf{p}_t used to randomly select the expert that takes control:

$$\mathbf{p}_t = \mathbf{m}_t / (\mathbf{u}^T \mathbf{m}_t) \quad (9)$$

where \mathbf{u} is a 3-element unit vector.

The weights of the actor, in particular only those related to the ‘winning’ (selected) expert, denoted with the 1×36 vector \mathbf{w}^{aw} , are updated as follows:

$$\Delta \mathbf{w}^{aw} = \eta^{as} \cdot s_t^s \cdot \sigma'[\mathbf{w}^{aw} \mathbf{x}_{t-1}] \cdot \mathbf{x}_{t-1} \quad (10)$$

where η^{as} is the learning rate of the selector’s actor and s_t is the selector’s critic surprise (see below). The effect of this learning rule is to increase

the m_{wt-1} of the selected expert if $s_t > 0$ and to decrease it if $s_t < 0$.

The selector’s critic is mainly formed by an ‘evaluator’ which is a 2-layer neural network that with its 1×36 vector of weights \mathbf{w}^v maps the retina activation to a linear output unit encoding the selector’s evaluation v_t of the perceived state:

$$v_t = \mathbf{w}^v \mathbf{x}_t \quad (11)$$

The critic uses the evaluator’s evaluations, together with the reward r_t (given its importance, this is discussed below), to compute the selector’s surprise s_t as follows:

$$s_t = (r_t + \gamma^s \cdot v_t) - v_{t-1} \quad (12)$$

where γ^s is the selector’s discount coefficient.

The weights of the evaluator are updated, on the basis of the surprise signal, using a TD learning rule (Sutton and Barto, 1998):

$$\Delta \mathbf{w}^v = \eta^{vs} \cdot s_t \cdot \mathbf{x}_{t-1} \quad (13)$$

where η^{vs} is the selector evaluator’s learning rate.

The reinforcement signal r_t used by the selector is particularly important and is computed in different ways during childhood and adulthood. During childhood $r_t = s_{wt}$, that is the reward is equal to the surprise of the selected expert. This implies that the selector uses an *intrinsic* reward not directly related to the achievement of specific *pragmatic goals* but to the acquisition of knowledge and skills, that is to *epistemic goals*. Indeed, as the *surprise of an actor-critic system is good indicator of the learning progress*, it can be used to train the selector to give control to the expert which is expected to learn at

the maximum rate in a certain state. During adulthood r_t is set to 1 when the robot achieves the target location of the task and to 0 otherwise. This implies that this is a standard goal-related *extrinsic* reward.

2.3 The genetic algorithm

The genetic algorithm uses a population of 50 individuals, each encoding the connection weights of the three experts’ reinforcers as real variables evolved for 100 generations (the initial values are randomly drawn in $[-1.0, +1.0]$). In a first condition of the experiment, the parameters are set as indicated in the last column of Tab. 1 (cl denotes the childhood’s length). This parameters have been ‘manually optimized’ by running some pilot experiments, and were also used in Schembri et al. (2007a) to compare the version of the architecture presented here with other versions of the architecture in which some other components were evolved together with the reinforcers.

In a second condition of the experiment such parameters are evolved as values ranging in $[0, 1]$ (cl was then mapped onto 600,000). For each of these two conditions the experiment is run 20 times with different seeds of the random number generator. In a third condition the same parameters of the second conditions are evolved with the exception of cl that is set at 14 different fixed values, namely $100 \cdot 2^i, i = 0, 1, 2, \dots, 13$; five runs using different ‘seeds’ are run for each of these values. The adulthood’s length al is set to 600,000 in all conditions.

The fitness f is computed as the number of times that the robot reaches the target divided by the theoretical maximum achievable if the robot followed the straight lines indicated in Fig. 1 at maximum speed. In the second condition a cost linearly related to the childhood’s length is introduced to induce the algorithm to optimize cl , so giving a ‘penalized’ fitness pf :

$$pf = f - (cl/al) \quad (14)$$

At the end of each generation the best 10 individuals are selected and generate 5 offspring each. Each weight of the five offspring of each parent (with the exception of the first one to have ‘elitism’) is mutated with a probability of 10% by adding to it a random value uniformly drawn in $[-1.0, +1.0]$. In the second and third condition also the aforementioned evolved parameters are mutated, with a probability of 10%, by substituting to them a random value uniformly drawn in $[0, +1.0]$.

3. Results

Fig. 3 reports the fitness along the generations of the evolution of the best individual in each generation and the average fitness of whole population, for both the first and second condition of the experiment. The figure shows that the best individuals of

the condition with evolved parameters reach a level (about 0.78) that is remarkably higher than the level of the condition with hand-tuned parameters (about 0.55), that is about 40% higher. On the other side the average fitness of such condition is only slightly higher than the other condition. The reason of this is that the mutations of the parameters can easily have catastrophic effects. Since the mutation of each of the seven parameters is 10%, and so the chances that an individual is mutated is quite high, this have a strong effect on the average fitness. Another interesting fact emerging from these simulations is that, as indicated in Schembri et al. (2007a), they confirm that the architecture has a high evolvability when the parameters are set to fixed values (notice how in this condition the ‘best’ and ‘average’ fitness increase in few generations). On the contrary evolution is rather slower when the parameters are evolved, indicating that the search in their space is not easy.

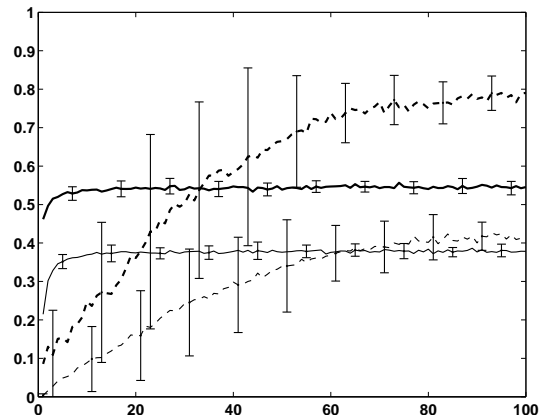


Figure 3: Fitness curves (y-axis) related to the best individuals (bold lines) and the average of the populations (thin lines) during evolution (x-axis reports the generations) of the simulations run with evolved parameters (dashed lines) and hand-tuned parameters (continuous lines). Each curve is the average of 20 different simulation runs. Note that the fitness measures of the condition with evolved parameters are related to f and not pf to ease comparisons.

The parameters evolved are indicated in Tab. 1. The most important fact is that life is sensibly shorter, (34,993 cycles) than the value (150,000) that was manually optimized. The reason why the system is capable of training the experts within such a short childhood’s length is likely that the combination of parameter found by the genetic algorithm are particularly well suited to allow a fast learning.

Passing to analyze the differences between the evolved and the hand-tuned parameters, the most interesting result is the imbalance between the learning rates of the actor and the critic of both the selector and the experts. These were manually set at the same values whereas the genetic algorithm

found quite different values, namely values from ten to twenty times lower for the critic than for the actor. This result was quite unexpected. The reason of this outcome is probably the fact that the evaluators' neural networks have linear output units whereas the actors' neural networks have sigmoidal output units. Since this implies that the derivative of the output units' used in the learning rules (cf. Eq.4 and Eq.10) is respectively equal to 1 or ranges within $[0, 0.25]$, the genetic algorithm found suitable learning rates to compensate this difference. This result is quite general as in neural-network implementations of actor-critic reinforcement learning system it is quite common to use neural networks with linear units to implement evaluators and neural networks with non-linear units to implement actors (included the popular 'soft-max function', see Sutton and Barto, 1998).

Another interesting fact is that the γ of the selector is higher than that of the experts, reflecting the intuition we had when we manually tuned the parameters, concerning the fact that the assemblage of experts takes place at a bigger spatial and temporal granularity with respect to the assemblage of primitive actions composing the experts' behaviors. However, also in this case the genetic algorithm found different (lower) and probably more effective levels of the parameters with respect to those we found by trial-and-error.

Table 1: The mean and standard deviation (Std) of evolved parameters.

Parameter	Mean	Std	Hand-tuned
cl	34,993	13,708	150,000
γ^s	0.9522	0.0630	0.99
η^{as}	0.7016	0.1811	0.05
η^{vs}	0.0351	0.0156	0.05
γ^e	0.6184	0.1945	0.90
η^{ae}	0.6214	0.1827	0.01
η^{ve}	0.0402	0.0296	0.01

Tab. 2 shows the correlations, measured with R^2 , existing between all the possible couples of parameters emerged in the 20 runs of the second condition of the experiment. Only two couples seem to have a high correlation. The first is the couple $\{cl, \eta^{ae}\}$. The reason of this correlation is likely that when the childhood is longer the genetic algorithm can find lower learning rates for training the experts' actors that allow the system to build a repertoire of more accurate behaviors. The other couple of highly correlated parameters is $\{\gamma^s, \eta^{vs}\}$. This correlation implies that different values of these two parameters can produce a similar performance to the extent that they are suitably balanced. In particular, the positive sign of the correlation (data not reported) implies that, within the selector, higher discount factors, which have the effect of slowing the propagation

of evaluations to states far from rewarded states, can be compensated for by a higher learning rate.

Table 2: R^2 between all couples of parameters.

	γ^s	η^{as}	η^{vs}	γ^e	η^{ae}	η^{ve}
cl	.0052	.0088	.0117	.0333	.3190	.1510
γ^s	—	.0370	.3469	.0175	.0576	.0002
η^{as}	—	—	.0007	.0006	.0024	.0065
η^{vs}	—	—	—	.0001	.0000	.0003
γ^e	—	—	—	—	.0064	.0004
η^{ae}	—	—	—	—	—	.0584

The length of childhood is particularly important as it constraints the possibilities of the system to acquire accurate building-block behaviors. Given a particular setup and typology of tasks as those considered here, it is useful to have a technique that allows drawing a quantitative picture of the relation existing between such length and accuracy of skills. In the third condition of the experiment the architecture's parameters were evolved while systematically setting the childhood's length to fixed values. Fig. 4, which reports the values of the fitness obtained at fixed childhood lengths, indicates that beyond a childhood's length of about 6,400 the system reaches a rather high level of fitness indicating that this is a minimal childhood's length beyond which the system succeed to develop a repertoire of quite reliable skills thanks to the evolved parameters (note that this implies that the system takes only about 1,050 cycles, on average, to train each of the six experts). The figure also shows that the system achieves a maximum fitness with a childhood's length ranging between 25,600 and 51,200. This is a notable result as the optimized childhood's length emerged in the second condition of the experiment is equal to 34,993 (see Tab. 1).

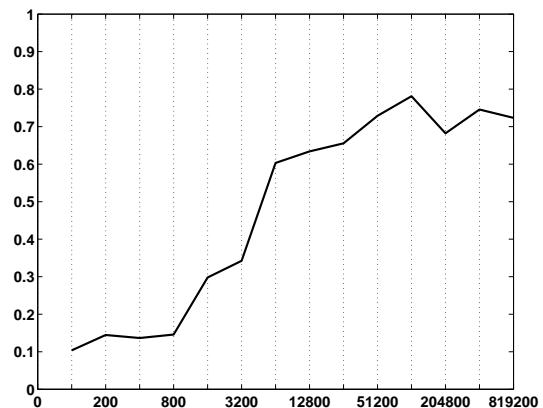


Figure 4: The fitness (y-axis) with different childhood lengths (x-axis). Each value of fitness is an average of 5 different simulation runs.

These results raise an interesting question: do the

parameters vary with increasing childhood’s lengths? Fig. 5 answers this question by reporting the values the genetic algorithm found with different childhood lengths. Limiting our analysis to values of childhood’s length that assured a high fitness (i.e. $> 6,400$), two facts are apparent from the graphs reported in the figure. The first is that with a longer childhood the experts’ actor and critic learning rates tend to decrease. This is in line with what found in terms of correlations between the parameters emerged in the second condition of the experiment: with a longer childhood the genetic algorithm can decrease the learning speed of the experts in order to increase accuracy. The second relevant fact highlighted by the figure is that all other parameters are quite stable with respect to the duration of childhood. This suggests that, given a setup and a set of tasks as those studied here, the parameters of the architecture that can be found with the genetic algorithm are quite reliable and robust.

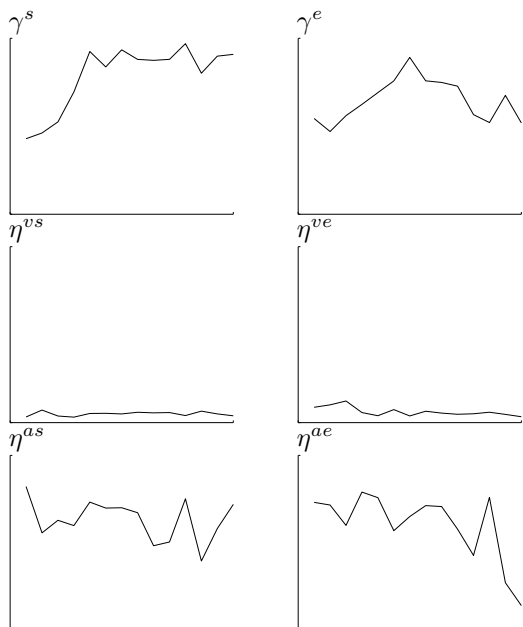


Figure 5: The evolved parameters (y-axis: this ranges in $[0, 1]$) with different childhood’s lengths (x-axis). Each value of fitness is an average of 5 different simulation runs. Above each graph the figure reports the R^2 related the correlation of the parameters with the childhood’s length.

4. Conclusions

This paper presented a neural-network two-level hierarchical reinforcement-learning architecture for Intrinsically Motivated Reinforcement Learning (IMRL) that exploits Evolutionary Robotic (ER) techniques to evolve various parameters it needs. Two previous works (Schembri et al., 2007a,b) showed that ER and the use of neural networks allow

the architecture to tackle two important limits of the models proposed so far within the Intrinsically Motivated Reinforcement Learning (IMRL) framework, namely (a) their applicability limited to problems with discrete abstract representations of states and actions, and (b) their need to be furnished ‘salient events’ by the programmer in order to be capable of learning the repertoire of skills. This work extended this research in two novel directions by further exploiting the ER framework, in particular it used a genetic algorithm to both optimize the learning parameters of the architecture and to optimize the time it spent in learning building-block behaviors. The viability of the proposed solutions was proved by using the architecture as controller of a simulated robot engaged in solving navigation path-finding tasks.

The results presented in the paper showed that the use of the genetic algorithm to evolve the learning parameters can lead to a notable increase of performance, about 40% here, with respect to the cases in which the parameters are tuned by hand. Remarkably, this increase of performance was obtained here while decreasing of about 75% the time (itself evolved) that the system dedicated to learn the skills.

The evolution of the parameters also showed that there might be two couples of parameters that, within some extent, might compensate one each other while guaranteeing a high performance: (a) a longer duration of the phase dedicated to train the skills (childhood) can be compensated for by a lower learning rate of the same skills (experts’ actors); (b) a higher discount factor used by the component of the architecture that assembles the skills (selector) can be compensated for by a lower learning rate it uses to update the evaluations of states.

The study presented here, related to the evolution of the (costly) time spent by the system in acquiring the repertoire of skills (childhood’s length), indicates a technique that can be used to identify the minimum amount of such time beyond which the system is capable of developing a repertoire of skills with a *satisfying* accuracy (say about 75% of the maximum one) and the time beyond which it achieves the *maximum* possible accuracy. Moreover, the results of these experiments showed that the learning parameters evolved are quite stable with respect to the duration of such period with two notable exceptions, namely the learning rates of the skills (experts) that, in line with the result of previous point (a), tended to be lower when such period got longer. Incidentally note that this type of experiments might also be used to investigate the emergence of childhood’s length that real organisms invest in playing and in the acquisition of skills while relying on parents for protection and food (for lack of space this issue cannot be further expanded here). To the authors’ knowledge, this is the first time that this type

of study is conducted in a systematic fashion.

Overall these results confirm that IMRL architectures can greatly benefit if developed within a ER framework. In fact ER allows to evolve aspects of the architectures, such as the learning parameters, that might be very hard to be directly designed/set a-priori as they produce highly non-linear and unpredictable effects.

Acknowledgements

This research was supported by the EU Projects *ICEA*, contract no. FP6-IST-027819-IP, and *Min-dRACES*, contract no. FP6-511931-STREP.

References

- Baldassarre, G. (2002). A modular neural-network model of the basal ganglia's role in learning and selecting motor behaviours. *Journal of Cognitive Systems Research*, 3:5–13.
- Barto, A. G., Singh, S., and Chentanez, N. (2003). Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*.
- Berlyne, D. (1960). *Conflict, Arousal, and Curiosity*. McGraw Hill, New York.
- Eriksson, A., Capi, G., and Doya, K. (2003). Evolution of meta-parameters in reinforcement learning algorithm. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 412–417.
- Huang, X. and Weng, J. (2002). Novelty and reinforcement learning in the value system of developmental robots. In Prince, C. G., Demiris, Y., Marom, Y., Kozima, H., and Balkenius, C., (Eds.), *Proceedings Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 47–55, Edinburgh, Scotland.
- Kaplan, F. and Oudeyer, P.-Y. (2003). Motivational principles for visual know-how development. In Prince, C. G., Berthouze, L., Kozima, H., Bullock, D., Stojanov, G., and Balkenius, C., (Eds.), *Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems 101*, pages 73–80, Boston, MA, USA.
- Mannella, F. and Baldassarre, G. (2007). A neural-network reinforcement-learning model of domestic chicksthatlearn to localize the centre of closed arenas. *Philos Trans R Soc Lond B Biol Sci*, 362(1479):383–401.
- Marshall, J. B., Blank, D., and Meeden, L. (2004). An emergent framework for self-motivation in developmental robotics. In *Proceedings of the 3rd International Conference on Development and Learning (ICDL 2004)*, pages 104–111, Salk Institute, San Diego.
- Nolfi, S. and Floreano, D. (1999). Learning and evolution. *Autonomous Robots*, 7(1):89–113.
- Oudeyer, P.-Y., Kaplan, F., and Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(6).
- Schembri, M., Mirolli, M., and Baldassarre, G. (2007a). Evolution and learning in an intrinsically motivated reinforcement learning robot. In *Proceeding of the 9th European Conference on Artificial Life*. Springer.
- Schembri, M., Mirolli, M., and Baldassarre, G. (2007b). Evolving internal reinforcers for an intrinsically motivated reinforcement-learning robot. In *6th IEEE International Conference on Development and Learning (ICDL2007)*.
- Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers. In Meyer, J. A. and Wilson, S. W., (Eds.), *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227, Cambridge, Massachusetts/London, England. MIT Press/Bradford Books.
- Stout, A., Konidaris, G. D., and Barto, A. G. (2005). Intrinsically motivated reinforcement learning: a promising framework for developmental robot learning. In *Proceedings of the AAAI Spring Symposium on Developmental Robotics*, Stanford University, Stanford, CA.
- Sutton, R. S. Precup, D. and Singh, S. (1999). Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211.
- Sutton, R. and Barto, A. (1998). *Reinforcement learning an introduction*. MIT Press, Cambridge, MA.
- Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*, 291(5504):599–600.
- White, R. (1959). Motivation reconsidered: the concept of competence. *Psychological Review*, 66(5):297–333.