Combining genes and memes to speed up evolution

Diego Federici
Norwegian University of Science and Technology
Department of computer and information science
N-7491 Trondheim, Norway
federici@idi.ntnu.no

Abstract- It is recognized that the combination of genetic and local search can have strong synergistic effects. In same cases though, the local search mechanism can be too aggressive, mislead the evolutionary search and produce premature convergence.

We set up a population of evolving agents also capable of learning by operant conditioning and communicating acquired behaviors (memes). The diffusion and discovery of memes gives rise to a second process of evolution atop of the genetic one. Memes are shown to have both guiding and hiding effects on baldwinian and lamarckian evolution.

In contraposition to previous models, simulations show that back-coding of acquired behaviors is highly beneficial only at the beginning of the evolutionary search. This result arises because of the different nature of the guiding provided by memes and the hiding effect that they generate.

To minimize the negative influence of the hiding effect but still benefit from the memetic guidance, we decrease the maximum number of memes that an agent can acquire as evolution proceeds.

Agents can then develop the optimal harvesting strategy in incremental steps with a great performance advantage.

1 Introduction

Even though Lamarck's theory of evolution [1] has been disproved there has been quite an interest in its application in artificial evolution.

As local search strategies can be more directed than genetic, back-coding of acquired characteristics can operate as a smart mutation operator yielding faster convergence to optima.

Since local search focuses on the most promising parts of the search space, it can increase the evolutionary speed in two ways: at the beginning by overlooking low fitness zones and at the end by climbing local optima. Of course, this more aggressive strategy can often produce premature convergence [3, 4].

The type of local search modulates the pro and cons of back-coding. This can be summarized by the *guide or hide* dichotomy [9, 10]). If on one side, learning can guide the evolutionary process by smoothing the fitness landscape (Baldwin effect [2, 5]) or through back-coding (lamarckianism [1, 3, 4]), on the other it can mask the selection pressure for certain characteristics hiding genetic differences and slowing down the entire process (hiding effect [9, 10]).

In this paper we adopt a learning mechanism based on acquisition through operant conditioning and communication. Individuals have a genetically encoded neuro-controller that outputs the expected reinforcements for the different possible actions. Individuals that experience an unexpected reinforcement build a meme (a reminder) that will allow them to avoid the same error in the future. At the same time, when two individuals fall into the communication range, they can exchange memes.

As memes are acquired, exchanged and dropped during a fitness evaluation, they give rise to a second evolutionary process atop of the genetic one. We will refer to this process as cultural evolution.

Previous work has already introduced models of cultural and memetic evolution. In [11, 12] culture is a population shared memory that acts as a global blackboard that individuals can read and write. The model of social exchange used in [13] is implemented by a crossover operator that combines the candidate with an individual of high fitness.

In this model memes are stand alone behavioral entities that reside on a single host and can be acquired and transmitted. If an agent perceives two memes to be similar enough, it will be merge them, generating a more general variant that can eventually spread in the population. The set of memes available to an individual specifies its culture and modifies its instinctive behavior. Since behavior determines the fitness scored by individuals, there is an evolutionary advantage in the development of fit cultures and therefore fit memes.

In this framework, the lamarckian back-coding of an individual's culture is shown to have a positive guiding effect in a first phase of the evolutionary search but is also shown to mask refinements of instinctive behaviors that do not yield immediate reinforcements. Because of the way that they are built, memes can encode only sources of immediate reinforcement while optimal control policies should also consider long term effects. Acquired behaviors have precedence over instinctive ones resulting in a censorial action of culture.

To minimize the negative effects of this cultural masking, we show that it is possible to decrease the number of memes that an agent can possess as evolution proceeds.

Although not applicable in every context, a good feature of this hybrid evolutionary system, is that it does not require additional fitness evaluations since memes are acquired online during the single fitness test.

2 Background

Lamarck's theory of evolution states that adapted traits are inheritable [1]. The discovery of germ cells disproved Lamarck's theory, still Baldwin suggested that there could have been a "new factor" that might operate in a similar way.

The Baldwin effect [2] states that phenotypic plasticity would allow adaptation to partially successful mutations, smoothing the fitness landscape and increasing the efficency of the evolutionary process.

However, phenotypic plasticity has inherent costs associated with the training phase in terms of energy, time and eventual mistakes. For these reasons, in a second phase, evolution may find a way to achieve the same successful behaviors avoiding plasticity. Thus a behavior that was once learned may eventually become instinctive.

In computer science, the phenotypic plasticity is analog to a local search strategy. The evolutionary process and the local search may be used in combination, often achieving higher efficency than either of the methods alone [6, 4, 7].

Hinton and Nowlan [7] were the first to prove the benefits of the Baldwin effect in a computer simulation. In a *needle in the haystack* function optimization problem, they showed that a local search mechanism could speed up evolution. The difficulty of the fitness landscape is dampened by the local search strategy, but since each step of the local search requires an additional fitness evaluation, the speed up of the evolutionary search is paid by the increased time required for each generation.

A different picture appears when considering the evolution of systems that require long fitness tests, for example controllers for situated agents. To get a good evaluation of the agent's fitness, it is often necessary to run several hundreds activations of its controller, see [16, 14, 15] among others.

In this context it is possible to add learning without requiring additional fitness evaluations. For example we can suppress behaviors that lead the robot to immediate negative reinforcements, such as when it hits an obstacle during the fitness test.

3 The model

We set up a population of 30 learning individuals that move in a 20×20 toroidal world. The world contains 30 of each of the two different types of resources: food and poison (see figure 1). When an agent visits a tile containing a resource it consumes it, receiving an immediate reinforcement. When consumed, the resource is removed and regenerated at random in the world.

The fitness is defined as the sum of the accumulated reinforcements over 150 simulation steps.

Each simulation step, an agent's reinforcement is computed as the sum of any of the following:

+0.8	if	visiting a tile containing food
-0.8	if	visiting a tile containing poison
-0.1	if	colliding with another agent
-0.1	if	the agent did not move

Each individual/agent is equipped with two different controllers. The first, a single layer neural network (NN) with hyperbolic tangent transfer function, is subject to an evolutionary process. The second is a classifier-like system (memes) and models the individual culture.

Agents perceive resources and other bots from all the 13 tiles within a hamming distance of 2 (see figure 1), this constitutes the input vector for both controllers.

For each of the 13 tiles, the 39 element boolean input vector contains a triplet T_{1-3} such as

T_1	1 if the tile contains food,	0 otherwise
T_2	1 if the tile contains poison,	0 otherwise
T_3	1 if the tile contains an agent,	0 otherwise

The neuro-controller also receives an additional input, always set, to provide the network bias.

The action performed by an agent is computed as follows:

- The NN produces 5 outputs. Each output is interpreted as the anticipated reinforcement (R_A) for each of the possible actions: don't move, go north, west, south and east. This constitutes the agent's instinctive response.
- The memes produce a set of reminders. Each reminder contains an action a and an experienced reinforcement R_E. These tell the agent that it seems to recognize the current input and that if action a is performed it will yield a reinforcement equal to R_E. This set constitutes the acquired responses.
- Acquired responses R_E replace the corresponding instinctive ones R_A (see figure 2), this constitutes the vector of expected reinforcements. The action with the highest expected reinforcement is selected with .7 probability. Otherwise a random action is selected.

3.1 genetic evolution

Given that the NN receives 40 inputs (a triplet for each of the 13 tiles within an agent's vision range, plus a bias), and that it produces 5 outputs (R_A), the weight matrix $\in \Re^{\{(39+1)\times 5\}}$.

The neuro-controller genotype is a linear gray-coded representation of its weight matrix.

Each generation, the best scoring 25% of the population survives and reproduces. Three quarters of the offspring are generated by the crossover of two randomly selected reproducing individuals; the remaining are generated by mutation of a single parent.

Mutation modifies each weight with a .2 probability by adding to it Gaussian noise with .25 variance.

As each NN can be considered composed of 5 independent sub-nets, one for each output, crossover produces two new individuals shuffling the parents sub-nets.

3.2 memetic evolution

Memes ideally remind the agent of the reinforcement experienced in the past[1]. They consist of an input pattern P, an action a, a value V and an experienced reinforcement R_E. If the pattern P matches the present input vector, then

[1] a meme could also have been acquired by communicating with other agents

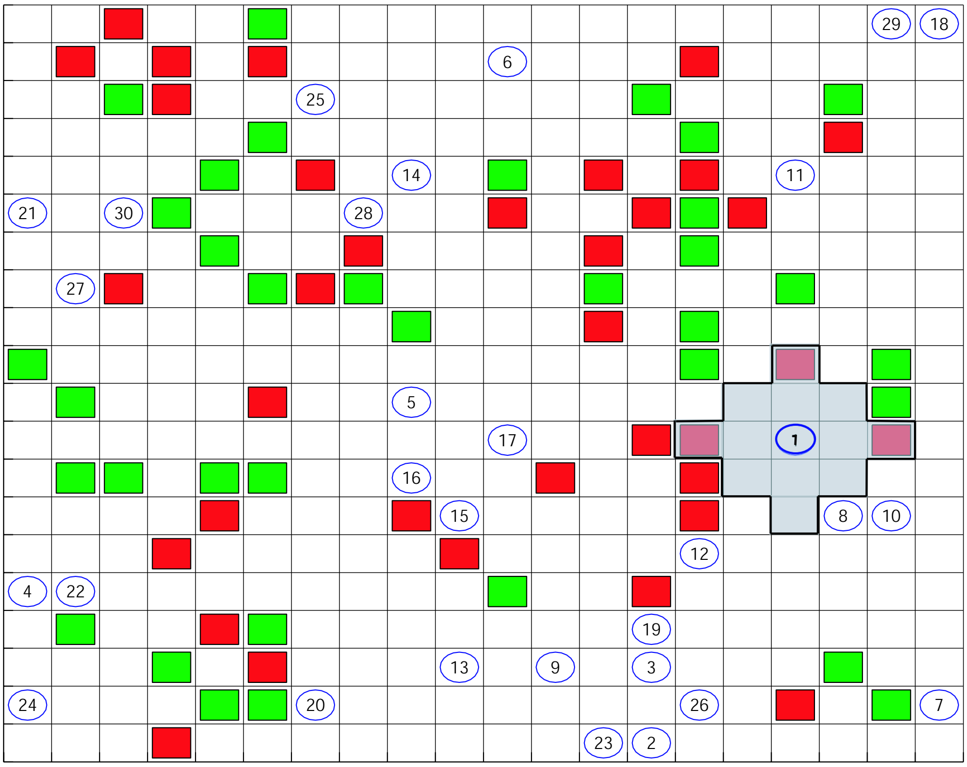


Figure 1: Simulated Environment. The vision range of agent 1 is shadowed and surrounded by a thick line. The two different types of resources are represented by squares of different colors. The resources represented by a darker color give a fixed negative reinforcement and fitness value (-0.8), while the others give a fixed positive value (+0.8). Resource types never change value and when consumed are regenerated on a random tile.

the meme replaces the output of the genetically evolved NN with R_E for the given action a. Basically the meme can recognize a particular sensory context (P), and it reminds the agent that in the past he had performed a certain action (a) and the action yielded a given reinforcement (R_E). If two memes match the current input, the one with highest value is used.

The input pattern P is a $\{-1, *, 1\}^{39}$ vector. Each element of P matches an element of the input vector. $*$ is a *don't care* symbol and matches any value of the input element.

An agent culture consists of up to 20 memes. Memes can be acquired either by transmission or by operant conditioning.

Transmission occurs whenever two individuals are next to each other. In such a way, the two agents can acquire each other's memes.

When an agent experiences an unexpected reinforcement a meme is generated through an operant conditioning mechanism. A reinforcement (R) is unexpected if the instinctive anticipation (R_A) is too different from the actual one:

$$R\ is\ unexpected\ if\ \mid R_A - R \mid \geq 0.075$$

The meme's pattern P is set to match the input vector proceeding the reinforcement, a is set to the performed action, R_E to the reinforcement and V equals $\mid R \mid$.

Memes variants are generated by merging, a stochastic generalization mechanism. Merging can occur if two memes code the same action and expected reinforcement. In this case, the merging probability (P_M) is inversely proportional to the hamming distance (d_H) between the memes input matching patterns:

$$P_M(meme_i, meme_j) = 1 - \frac{d_H(P_{meme_i}, P_{meme_j})}{39}$$

where the distance between elements containing a $*$ is 0.

Merging is seen as a weak simplification of a boolean functions:
given ($P_1 \wedge a \mapsto R$) and ($P_2 \wedge a \mapsto R$) then with a probability proportional to the similarity of P_1 and P_2 replace them with $((P_1 \circledast P_2) \wedge a \mapsto R)$; where $P_i \in$ pattern, $a \in$ action, $R \in$ reinforcement and $\circledast$ is a bitwise operator:

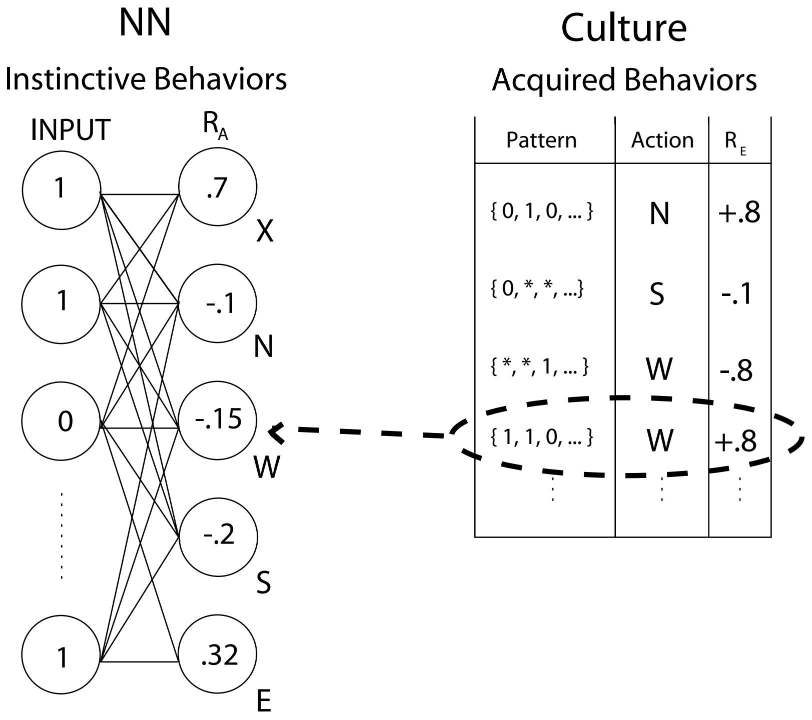
$$\circledast(b_i, b_j) = \begin{cases} b_i & if \quad b_i = b_j \\ * & if \quad b_i \neq b_j \end{cases}$$

The value of the new meme is set to $\mid R \mid \cdot N_*$ where N_* is the number of *don't care* symbols in the new meme.

If it does not merge, a meme can be added only if the meme pool size does not exceed the maximum. If the maximum is exceeded a meme is dropped, the one with lower value being dropped with higher probability. Because merging of memes can sometimes produce unfit memes, if the expected reward does not match the one experienced, the meme responsible for the error is instantly removed.

3.3 Lamarckianism

Since memes are generated when the neuro-controller makes a prediction error, the situations that their patterns represents are a source for possible improvements of the genotype. A meme can then be used as a training example with which increase the neuro-controller performance.



Pattern	Action	R_E
{ 0, 1, 0, ... }	N	+.8
{ 0, *, *, ...}	S	-.1
{ *, *, 1, ... }	W	-.8
{ 1, 1, 0, ... }	W	+.8
⋮	⋮	⋮

Figure 2: Agent controller. The genetically evolved NN and the acquired culture are activated in parallel. When a pattern matches the current input, the corresponding meme is activated (encircled in the figure). The meme's R_E value $(+.8)$ replaces the NN output $(-.15)$ for the corresponding action (go west). This modified output vector is interpreted as the agent's expected reinforcements when performing each of the possible actions. The action that is actually performed is selected stochastically giving a .7 probability to the one with highest value in the modified output vector. If, after performing the an action, the actual reinforcement received is too different from the expected one, the agent's culture is modified (refer to section 3.2).

In this way, acquired behaviors are coded back into the genotype in a lamarckian process.

At the beginning of each new generation, all the population undergoes a training phase based on its acquired behaviors. Each individual's culture is used to train its neural network in 10 steps of back-propagation. The gradient is computed in batch mode using each meme as a training exemplar. $*$'s are replaced by zeros for this purpose. A learning rate of .25 is used.

The offspring uses one of the parent's culture for training (but does not inherit the culture itself).

This method cannot compute the exact gradient as each $*$ symbol distorts the error back-propagation. In fact, patterns containing N_* *don't care* symbols, should have each of them replaced by $\{-1, 1\}$. But this would give rise to the expansion of 2^{N_*} training exemplars for each pattern, with the result being too computationally expensive.

Because of this distortion, the training mechanism is actually performing only an incomplete back-coding of the acquired behaviors.

4 Results

We have tried four different simulation settings:

- Genetic: standard genetic evolution without memes.
- Memetic: no evolution, individuals continue acquiring memes through all the simulation. All instinctive outputs are set to zero.
- Baldwinian: both memes and genes evolve. No back-coding takes place.
- Lamarckian: both memes and genes evolve. Memes are used to modify the genes.

Figure 3 shows the average over 10 runs of the average population fitness with and without the help of memes. Selection is performed on fitness values plotted in figure 3A, but since the task is to optimize the evolution of the genetically encoded NN, its performance is given in figure 3B.

Memetic performance increases more quickly than the genetic one. The average fitness scored at generation 25 by memetic populations is at least as high if not better than any of the others even after 500 generations.

But since memes can suggest only actions that yield an immediate reinforcement, the genetically evolved NN could take advantage of the two squares vision range, also approaching distal resources.

Tests performed on a population with an optimal one-square vision controller gave a score of ~ 14, while with an optimal two-square vision controller the score was ~ 20. This means that a genetically evolved NN can score 40% higher fitness.

The baldwinian simulations are the most penalized from the use of memes. The acquired behaviors appear to mask the pressure towards the evolution of the appropriate neuro-controller.

An interesting picture emerges in the comparison between lamarckian and genetic simulations. During the first 250 generations, lamarckian runs outperform standard genetic evolution. After that the *hiding effect* takes over and while the standard genetic evolution keeps on improving the neuro-controller performance, lamarckian populations

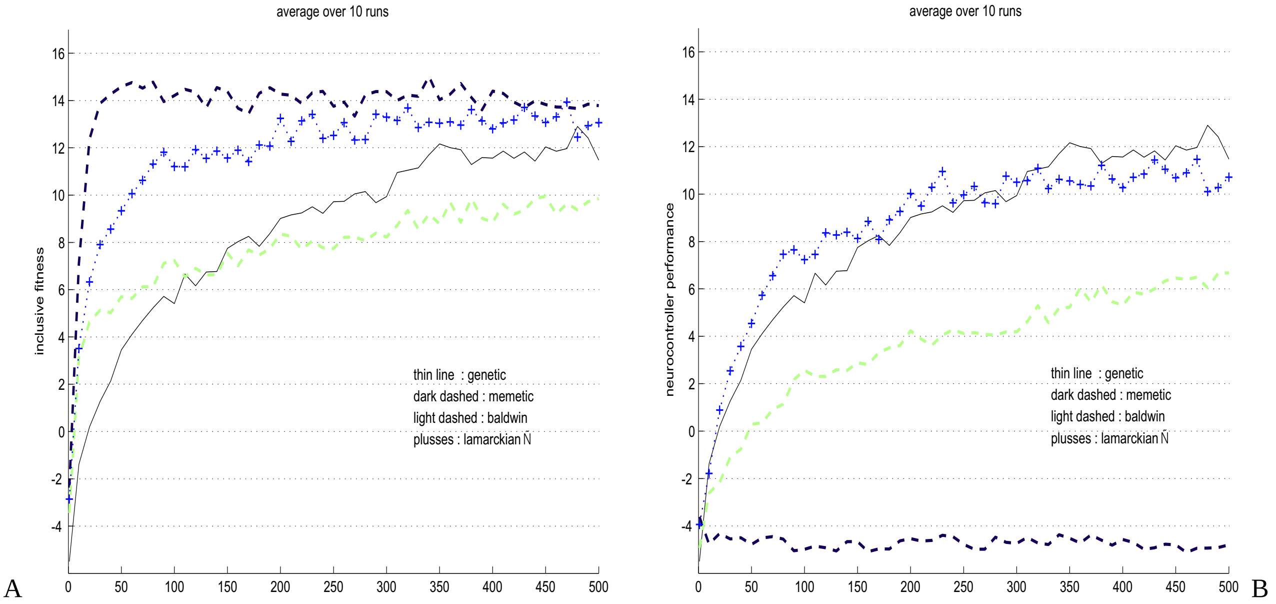


Figure 3: Inclusive Fitness and neuro-controller performance plots. A: inclusive fitness. B: neuro-controller performance. The performance of the neuro-controller is computed without the mediation of agent's cultures, in this way only the evolved NN is responsible of the agent's fitness. Population averages over 10 runs.

show a vary slow increase.

The problem is that their behavior is still very dependent on culture. Genetic assimilation and back-coding of memes, should make the genotype less prone to mistakes and hence reduce cultural acquisition (see section 3.2). This effect takes place but is shown to be slow, see figure 4.

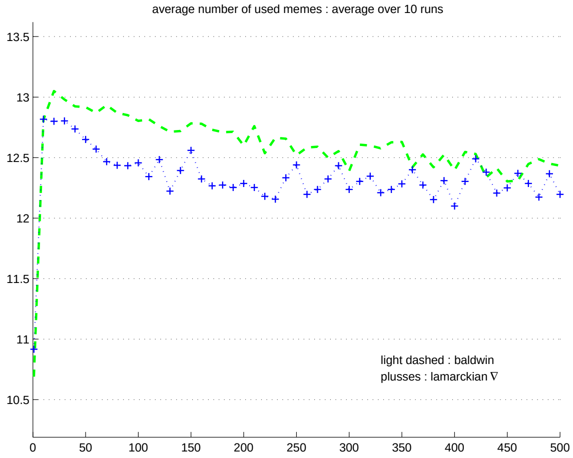


Figure 4: Population average of the number of memes used calculated over 10 runs. Both genetic assimilation and back-coding are slowly reducing the need of memes.

Partial lamarckianism, i.e. only a fraction of the population undergoes the back-coding phase, cannot solve this problem, see figure 5. The stagnation of the evolutionary process is not caused by convergence to a local optima, but by the hiding effect cause by the individuals' cultures.

To reduce the hiding effect, it is possible to progressively decrease the maximum number of memes. This will unmask the advantage for the refinement of the neuro-controller in a

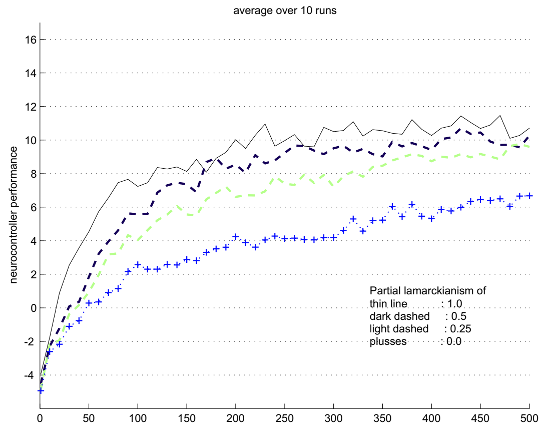


Figure 5: Neuro-controller performance plots for populations with partial lamarckianism. The fraction of the population to which back-coding is applied is varied from 0 to 1 (lamarck share).

second phase of the evolutionary process.

The maximum number of memes is reduced every 50 generations from 20 to 0, and at the same time, each individual culture is reset.

As the inclusive fitness must increasingly rely on the genetically evolved NN, pressure is gradually shifted from memes to genes. Results show that populations using lamarckianism have both a higher average and steeper increase in performance, see figure 6.

Figure 7 presents a summary of the performance for the best populations with different experimental settings. Every population with a neuro-controller capable of scoring a fitness greater than 15 is represented by a dot. Populations scoring more than 19 contain almost only optimal

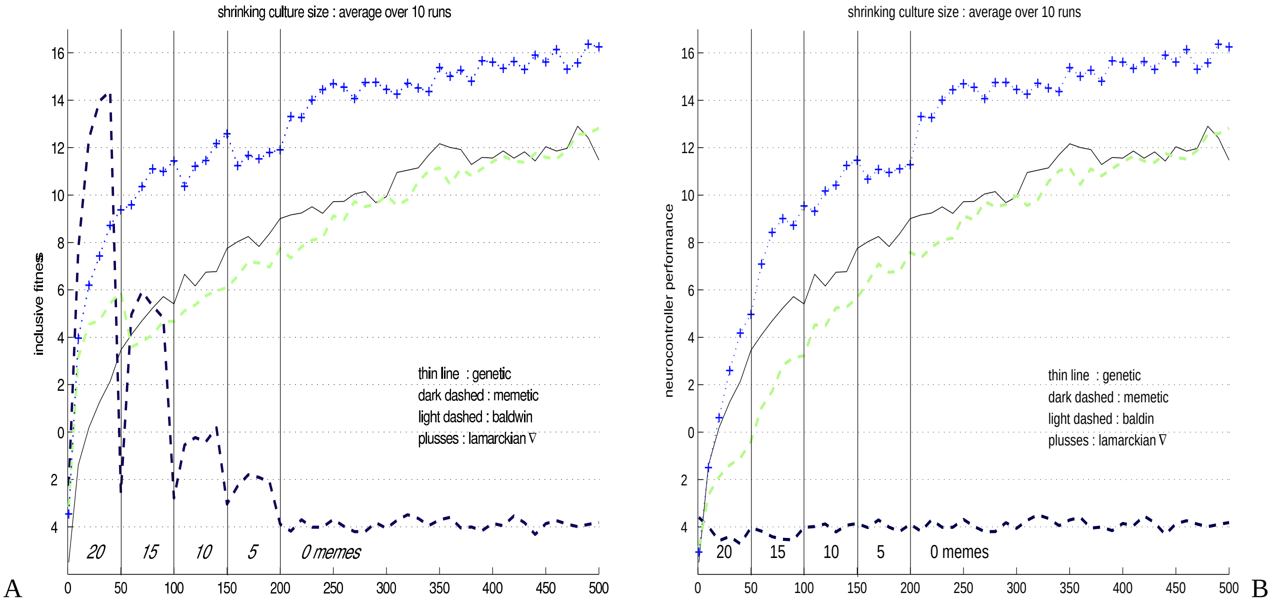


Figure 6: Inclusive Fitness and neuro-controller performance plots with a decreasing maximum culture size. Vertical lines indicate when the meme pool is reset and shrunk. The average performance of the genetic simulations is plotted for comparison. A: inclusive fitness. B: neuro-controller performance. The performance of the neuro-controller is computed without the mediation of agent's cultures, in this way only the evolved NN is responsible of the agent's fitness. Population averages over 10 runs.

controllers. The plot shows the performance enhancement given by lamarckianism with a decreasing culture size.

5 Conclusions

This paper introduces a hybrid model of genetic and cultural evolution. The objective is to develop a neuro-controller for situated agents performing a harvesting task.

Cultural evolution is based on a simple mechanism of operant conditioning and memetic transmission. Given that there are 2^{39} possible inputs and that there are 3^{39} different matching patterns, developing general and fit memes without a priori knowledge is not a trivial task. Cultural evolution is shown to quickly develop behaviors that constitute an incomplete but fit strategy.

The incompleteness of the memetic strategy arises from its innovation process, which considers only immediate sources of reinforcements. Nevertheless, back-coding of memes allows individuals to quickly assimilate behaviors, speeding up the development of an optimal controller.

After this phase of increased evolutionary speed, culture appears to mask any further development. Being based on expectation, the acquisition mechanism was designed to allow un-masking of the complete strategy. This effect takes place but is very slow.

To accelerate it, the maximum number of memes is externally reduced, thus forcing individuals to rely more and more on instinctive behaviors. The method is shown to increase the performance of baldwinian and lamarckian simulations, with the latter capable of outperforming the other evolutionary strategies both in convergence speed and fitness score.

Lamarckian populations seem to benefit from the incremental refinement of the harvesting strategy. Mediated by cultural evolution, at first individuals develop a one-square vision optimal strategy and only afterwards, with the disappearance of memes, the two-square vision strategy is obtained.

These results suggest a methodology for an incremental development of control strategies. By assigning a reinforcement to causes of immediate fitness change (i.e. hitting a wall, entering a target zone or moving to full speed) and with culture back-coding, individuals will quickly learn to perform well in the most trivial cases. Individuals can thereafter discover the complete control strategy building upon the incomplete memetic one, thus saving evolutionary time.

As a final remark, the benefits of the lamarckian process must be inclusive of the overhead introduced by the local search mechanism. As in Houck et al. [4] it is necessary to compare performance taking as a reference the number of function evaluations and not generations alone. To this respect, cultural evolution does not require additional effort. In spite of this fact, populations with a maximum of 20 memes ran in simulation up to 4 times slower than those that did not use memes.

Acknowledgments

The simulations presented in this paper are very computational expensive. Results have been produced by the use of the (inexpensive) ClustIS cluster [20]. All the code has been written in Matlab.

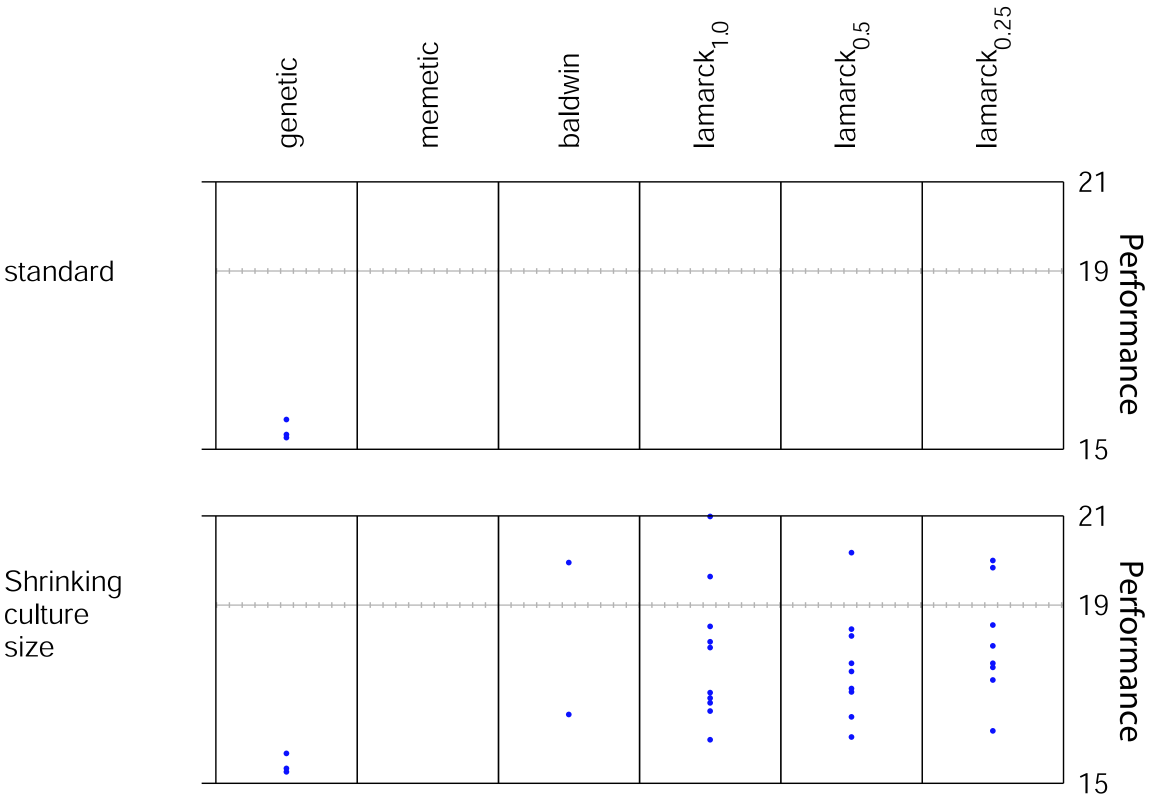


Figure 7: Neuro-controller performance plots for all the different simulations. Each dot represents the final average performance of one population (only those with a score greater than 15 are plotted).

Bibliography

[1] J.B. Lamarck: Philosophie zoologique, 1809

[2] J.M. Baldwin: A new factor in evolution. American Naturalist **30** 441–451 (1896)

[3] D. Whitley: Modeling Hybrid Genetic Algorithms. in Genetic Algorithms. Wiley and Sons, 191–201 (1995)

[4] C.R. Houck, J.A. Joines, M.G. Kay and J.R. Wilson: Empirical Investigation of the Benefits of Partial Lamarckianism. Evolutionary Computation 5(1), 31-60 (1997)

[5] P. Turney: Myths and legends of the Baldwin Effect. Proceedings of the ICML-96 (13th International Conference on Machine Learning) 135–142 (1996)

[6] R.K. Belew, M. Mitchell: Adaptive individuals in evolving populations: models and algorithms. Addison-Wesley (1996)

[7] G.E. Hinton, S.J. Nowlan: How learning can guide evolution. Complex Systems **1** 495–502 (1987)

[8] R. French, A. Messinger: Genes, phenes and the Baldwin effect. In Proceedings of Artificial Life IV 277–282 (1994)

[9] G. Mayley: Landscapes, Learning Costs and Genetic Assimilation. In Evolutionary Computation, Evolution, Learning and Instinct: 100 years of the Baldwin Effect (1996)

[10] G. Mayley: Guiding or hiding. In the Proceedings of the Fourth European Conference on Artificial Life (ecal97) (1997)

[11] L. Spector and S. Luke: Cultural Transmission of Information in Genetic Programming. in Genetic Programming 1996: Proceedings of the First Annual Conference, 209–214 (1996)

[12] L. Spector and S. Luke: Culture Enhances the Evolvability of Cognition. in proceedings of Cognitive Science 1996 (1996)

[13] C. Wellock and B.J. Ross: An Examination of Lamarckian Genetic Algorithms. 2001 Genetic and Evolutionary Computation Conference Late Breaking Papers, 9-11, 474–481 (2001)

[14] S. Nolfi, J.L. Elman and D. Parisi: Learning and Evolution in Neural Networks. Adaptive Behavior, Vol. 3, 1, 5–28 (1994)

[15] D. Floreano and F. Mondada: Evolutionary Neurocontrollers for Autonomous Mobile Robots. Neural Networks, 11, 1461–1478 (1998)

[16] J. R. Koza: Genetic Programming II. MIT Press, Cambridge (1994)

[17] J.H. Holland: Adaptation in natural and artificial systems. MIT press, Cambridge (1975)

[18] E.R. Kandel, J.S. Schwartz, T.M. Jessell: Principles of neural science. Elsevier Science Publishing Co., New York (2000) 4th edition

[19] H. Markram, J. L'bke, M. Frotscher, B. Sakmann: Regulation of Synaptic Efficacy by Coincidence of Post-synaptic APs and EPSPs. Science, **275** 213–215 (1997)

[20] J. Cassens and Z. Constantinescu Fülöp: It's Magic: SourceMage GNU/Linux as HPC Cluster OS. to appear in Proceedings Linuxtag 2003, Karlsruhe, Germany, July 2003. http//clustis.idi.ntnu.no/