Proceedings of the
International Conference on Complex Systems
Nashua, NH, 21–26 Sept. 1997.

Y. Bar-Yam (ed.), Addison-Wesley (1998).
Also on-line in the *InterJournal*.

An Artificial Life Model for Investigating the Evolution of Modularity

Raffaele Calabretta*
Yale University, EE Biology Dept.
Institute of Psychology,C.N.R.,Rome
Stefano Nolfi
Institute of Psychology,C.N.R.,Rome
Domenico Parisi
Institute of Psychology,C.N.R.,Rome
Günter P. Wagner†
Yale University, EE Biology Dept.

To investigate the issue of how modularity emerges in nature, we present an Artificial Life model that allow us to reproduce on the computer both the organisms (i.e., robots that have a genotype, a nervous system, and sensory and motor organs) and the environment in which organisms live, behave and reproduce. In our simulations neural networks are evolutionarily trained to control a mobile robot designed to keep an arena clear by picking up trash objects and releasing them outside the arena. During the evolutionary process modular neural networks, which control the robot's behavior, emerge as a result of genetic duplications. Preliminary simulation results show that duplication-based modular architecture outperforms the nonmodular architecture, which represents the starting architecture in our simulations. Moreover, an interaction between mutation and duplication rate emerges from our results. Our future goal is to use this model in order to explore the relationship between the evolutionary emergence of modularity and the phenomenon of gene duplication.

1 Introduction

In evolutionary biology, the concept of modularity is used to capture the fact that the bodies of higher organisms appear to be composed of semi-autonomous units ([9]; [11]; [1]). It has been argued that modularity is a `prerequisite` for the adaptation of complex organisms: modularity would allow the adaptation of different functions with little or no interference from other functions ([1]). However, this explanation raises two important questions. First, to say that modularity is a prerequisite for the adaptation of complex organisms seems to imply the need to explain the origin of a 'trait' (module) potentially useful for the species but not for the organism at the time of its emergence. This question is related to the more general issue of the evolution of the boundary conditions of evolution, in particular of the genotype-phenotype mapping (see [12]). Second, to say that modularity would allow the adaptation of different functions with little or no interference from other functions implies that we should be able to find a class of selective forces that can shape the genotype-phenotype mapping to allow for the existence of selective pleiotropic effects between genes, complexes of characters, and functions (see [11]). (Pleiotropy is "the influence of the same genes on different characters", [3], p. 429).

In more general terms, modularity requires that we look at a complex system from the point of view of three different and chronologically successive phases: the phase of emergence, the phase of actual functioning, and the phase of maintenance. In fact, for modularity to exist it is necessary for many different 'elements' to interact locally and nonlinearly at a number of different levels: genetic, phenotypic, physiological, and behavioral level. This complexity makes it more difficult to choose the right model for study and, therefore, to find the answers to important questions. As a consequence, even if the fact and the importance of modularity seems to be widely appreciated, there is little understanding of how modularity originates, works, and remains incorporated in the genome.

To evolve a neural controller for a mobile robot, Nolfi ([7]) used a modular neural network architecture that clearly outperformed other architectures in performing a task of garbage collecting (see below). To investigate the issue of how modularity can emerge in nature, we present a modification of Nolfi's model ([7]) in which gene duplication is also included as part of the evolutionary process and, therefore, modular neural networks can evolve starting from a population of non-modular ones as a result of gene duplication. Our future goal is to use the model to explore the relationship between the evolutionary emergence of modularity and the phenomenon of gene duplication.

Our preliminary simulation results show that duplication-based modular architecture outperforms non-modular architecture, which represents the starting architecture in our simulations. Moreover, an interaction between mutation

*This work has been made possible thanks to a fellowship from the Italian National Research Council (National Advisory Committee for Biological and Medical Sciences), which financially supported the one-year sabbatical stay of Raffaele Calabretta at Yale University in 1997. RC would also like to acknowledge the support of Aurelio Romeo, Jeffrey R. Powell, Valerio Sbordoni and Riccardo Galbiati, the assistance of Ginger Booth and Jan Taschner, and the useful discussions with the members of the GPW's lab at Yale University during weekly meetings.

†The financial support by NSF grant BIR 9400642 and the Yale Institute of Biospheric Studies is gratefully acknowledged.

and duplication rate emerges from our results.

2 The Model

We ran a set of simulations in which neural networks ([10]) are evolutionarily trained to control a mobile robot designed to keep an arena clear by picking up trash objects and releasing them outside the arena. The robot has to look for 'garbage', somehow grasp it, and take it out of the arena (see [8]).

The organism is a miniature mobile robot called Khepera, developed at E.P.F.L. in Lausanne ([6]). The robot is supported by two wheels that allow it to move in various directions by regulating the speed of each wheel. In addition, the robot is provided with a gripper module with two degrees of freedom. The two arms of the gripper can move in parallel through any angle from vertical to horizontal while the gripper can assume only the open or closed position. The robot is also provided with eight infrared proximity sensors (six sensors are positioned on the front of the robot and two on the back) and an optical barrier sensor on the gripper capable of detecting the presence of an object between the two arms of the gripper. The infrared sensors allow the robot to detect obstacles to a distance of about 4 cm. The environment is a rectangular arena 60x35 cm surrounded by walls containing 5 target objects. The walls are 3 cm in height and target objects are cylindrical boxes with a diameter of 2.3 cm and a height of 3 cm. The targets are positioned randomly inside the arena. To speed-up the evolutionary process a simulator was used (see [8]).

In the present work we compare the results obtained with three different neural network architectures (see Figure 1). In all cases the robot has 7 sensor neurons and 4 motor neurons. The first 6 sensory neurons are used to encode the activation level of the corresponding 6 frontal sensors of Khepera (the two back sensors are ignored) and the seventh sensory neuron is used to encode the light sensor on the gripper. On the motor side the 4 neurons respectively codify for the speed of the left and right wheels and for the triggering of the 'object pick up' and 'object release' procedures.

The activation values of the infrared sensors (which have 1024 different values ranging from 0 to 1023) and of the activation of the light-barrier sensor (which can have two values, 0 or 1023) were encoded in sensory neurons as floating point values between 0.0 and 1.0. The logistical function was used to determine the activation of the motor neurons. The activation of the first two motor neurons controlling the left and right wheels was transformed into 21 different integer values ranging from -10 to +10 (maximum speed backward and forward, respectively). The activation of the third and fourth motor neurons controlling the picking-up and releasing procedures, respectively, were thresholded into two values (1 = trigger the corresponding procedure, 0 = do not trigger the corresponding procedure).

The first architecture (a) is a simple feedforward network with 7 input units encoding the state of the 7 sensors and four output units encoding the state of the four effectors.

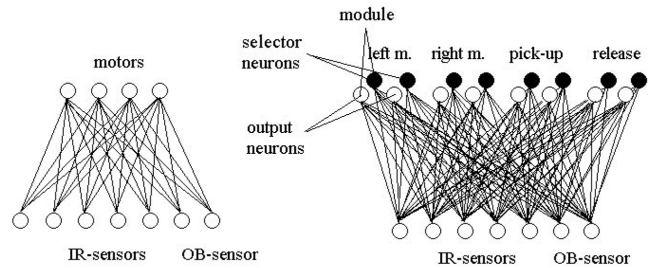

Figure 1: Figure 1 Architectures (a) and (b-c) are shown on the left and right side, respectively. Architectures (b) and (c) are structurally identical. However, in architecture (b) two modules compete to gain control of each of the four actuators. Individuals of the initial population with architecture (c) have only one module for each motor. However, a second competing module may be added in individuals of subsequent generations as a result of the duplication operator. Another difference is that in case (b) competing modules start with different random weights while in case (c), when a second competing module is introduced, the two competing modules have identical weights.

The input units are directly connected to the output units through 28 connection weights (plus 4 biases). This architecture is not divided into modules.

The second architecture (b) is a modular one and it has been called emergent modular architecture ([7]) because it allows the required behavior to be broken down into sub-components controlled by different neural modules, although it does not require the designer to do such a partition in advance. (Notice that in this paper the emergent architecture is referred to as `hardwired modular architecture`). There are two modules for each of the four outputs (the two wheels, the object pick up procedure, and the object release procedure). In any particular input/output cycle only one of the two competing modules can control the corresponding output.

Each module includes two output units: a motor output unit and a selector unit. The motor output unit determines the speed of the corresponding wheel or whether or not the two procedures are executed. The selector unit determines the probability that the module will control the corresponding output. In other words, which of the two competing modules determines the output depends on which of the two competing selector units is more activated. Both the motor output unit and the selector unit of each module receives 7 connections (plus one bias) from the 7 sensory neurons.

The third architecture (c) is also modular and is denoted as `duplication-based modular architecture` because, in this case, the modules are not hardwired in the architecture from the beginning of evolution but they can be added during the evolutionary process. Each module, as in the case of architecture (b), consists of two output units (one motor output unit and one selector unit) which receive connections from the 7 sensors. At the beginning of the evolutionary process there is only one module for each of the four outputs, i.e., always the same module controls the corresponding output. However, at reproduction, modules may be duplicated (see below). Duplicated modules, which are exactly the same when duplication takes place, can differentiate across generations because of genetic mutations.

A genetic algorithm ([4]) was used to evolve the connection weights of our neural networks. Each connection weight or bias is encoded as a sequence of 8 bits in the genotype. We begin with 100 randomly generated genotypes each representing a network with the same architecture and a different set of random connection weights. Each individual is allowed to 'live' for 15 epochs, each epoch consisting of 200 input-output cycles or actions. At the beginning of each epoch the robot and the target objects are randomly positioned in the arena. An epoch is terminated either after 200 actions or after the first object had been correctly released. At the end of life, the best individuals are selected for reproduction. The 20 individuals that have accumulated the highest 'score' (i.e., performance measure; see below) during their lives generate 5 copies each of their neural networks. These 20x5=100 new robots constitute the next generation. The process is repeated for 1000 generations.

Reproduction consists in generating copies of an individual's genotype encoding the network's connection weights (we are assuming non-sexual reproduction in haploid populations) with the addition of random changes to some of the bits of the genotype sequence (genetic mutations) and, in the case of architecture (c), the duplication of a random selected neural module. The fitness formula is the way in which individuals are evaluated in order to decide who is allowed to reproduce. Individuals were scored for their ability to perform the complete sequence of correct behaviors, i.e., for their ability to release objects correctly outside the arena. However, in order to facilitate the emergence of this ability individuals were also scored (even if with a much lower reward) for their ability to pick up targets. More precisely, individuals were scored with 5 for each cycle they had an object in the gripper and with 10000 for each object correctly released outside the arena.

In the present preliminary model the maximum number of duplicated modules allowed in the case of architecture (c) is one for each motor output and no module-deletion operator was used. As a result, the architecture (b), already described in Nolfi ([7]), is the more complex architecture that can possibly evolve starting from architecture (a). However, the addition of competing modules during the course of evolution (instead than right from its beginning) that are initially identical to their competing module (instead of being completely unrelated) may produce qualitatively different results in the case of architecture (b) and (c), respectively.

3 Preliminary Results

We present the results of several simulations in which we compare a simple feedforward neural network, the hardwired modular architecture, and the duplication-based modular architecture (see Figure 1). In all simulations a mutation rate of 1% was used, i.e., 2% of the bits of the genotype randomly selected were replaced with a new randomly selected value. For the duplication-based modular architecture we investigated the performance obtained with a duplication rate of 0.02%, 0.03% and 0.04%, i.e., 0.02%, 0.03% and 0.04% of the modules were duplicated in each replication. We ran 10 simulations for each of the 3 different architectures described above. Each simulation started with a population of 100 networks with random connection weights and lasted 1000 generations.

Figure 2 shows the average and peak performance for non-modular robots and for duplication-based modular robots with a duplication rate of 0.04%. In both conditions the performance level increases until a plateau is reached. However, modular robots achieve a higher terminal performance level and need less time (generations) to reach this level. This result confirms Nolfi's observation ([7]) that a modular architecture is useful in accomplishing a complex task.

Let us consider the individual results obtained in the 10 different repetitions of the simulation. In the case of duplication-based modular architecture, the ability to accomplish the desired task rapidly evolves in all replications. If we examine the genotypes of the best individuals across generations, we see that they incorporate at least one duplicated module at the time the performance level increases significantly (results not showed). The picture is different in the case of the non-modular architecture. In some of the replications it takes many generations for the performance level to reach the plateau; moreover, in one replication performance does not increase at all (results not showed).

Both populations with modules reach a higher fitness level than a population with only the basic architecture and no modules (see Figures 2 and 3). However, the two populations with modules do not differ in terms of overall fitness except that fitness growth is slightly slower in the population with duplication-based modules (see Figure 4).

In order to demonstrate that modularity plays a critical role, we varied the duplication rate in the population with duplication-based modules, with the result that both average and peak performance decreased linearly with a decreased duplication rate (0.04%, 0.03%, and 0.02%; results not shown). Figure 5 shows the results obtained with a duplication rate of 0.02% and compares these results with those obtained with a non-modular architecture: the advantage of modular design is lost. This result underscores the importance of the interaction between mutation and duplication rate.

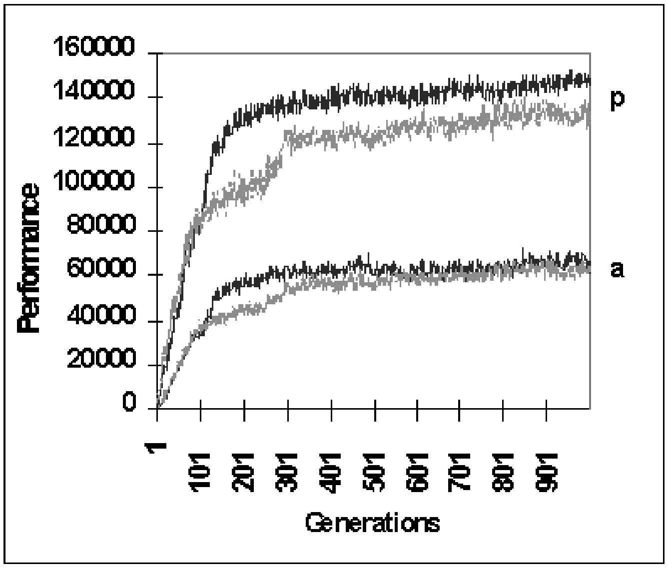

Figure 2: Average (a) and peak (p) performance of non-modular robots (grey curve) and duplication-based modular robots (black curve) with a duplication rate of 0.04%. Average of 10 different replications of the simulation.

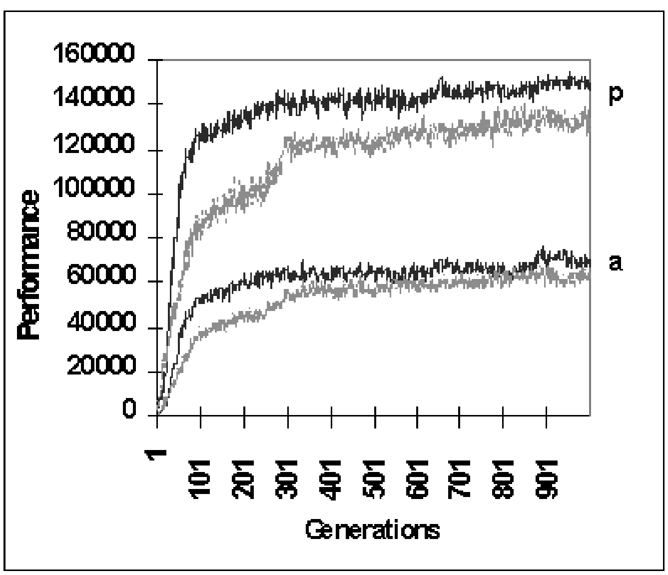

Figure 3: Average (a) and peak (p) performance of non-modular robots (grey curve) and hadwired-modular robots (black curve). Average of 10 different replications of the simulation.

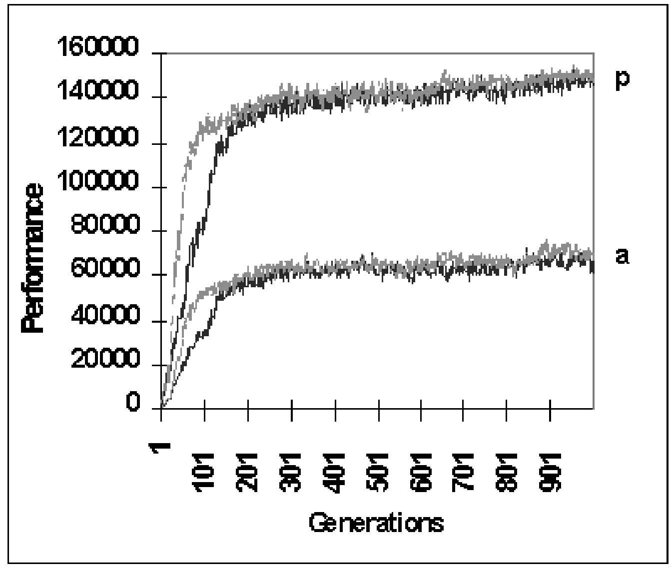

Figure 4: Average (a) and peak (p) performance of hadwired-modular robots (grey curve) and duplication-based modular robots (black curve) with a duplication rate of 0.04%. Average of 10 different replications of the simulation.

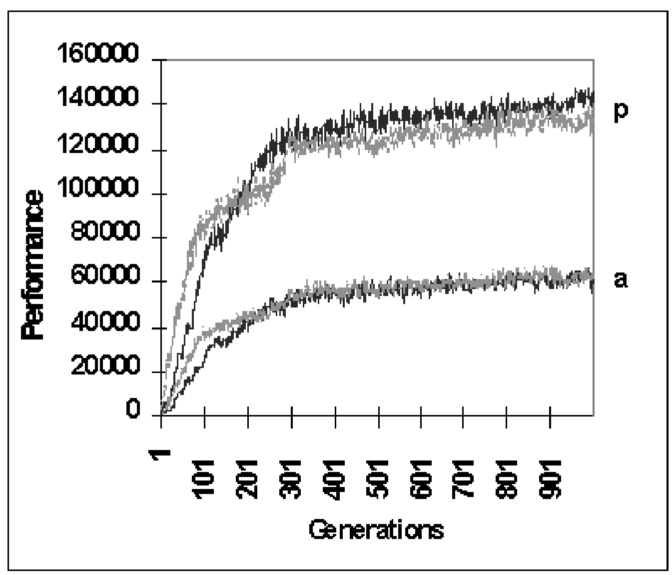

Figure 5: Average (a) and peak (p) performance of non-modular robots (grey curve) and duplication-based modular robots (black curve) with a duplication rate of 0.02%. Average of 10 different replications of the simulation.

4 Conclusions

In this paper we have described an Artificial Life model based on neural networks and genetic algorithms which can be used to understand the evolutionary mechanisms underlying the origin of modularity in nature, and we have presented some preliminary results obtained by comparing a nonmodular architecture, a hardwired modular architecture ([7]) and a duplication-based modular architecture. We used the same simulation scenario of Nolfi ([7]) but we added the genetic operator of gene duplication to explore the relationship between the evolutionary emergence of modularity and the phenomenon of gene duplication.

The cross-fertilization between Artificial Life and biology can take place since Artificial Life partially shares the theoretical apparatus and vocabulary of evolutionary biology and can offer additional methodological tools to biology. More specifically, our model allows us to reproduce in a computer both the organisms and the environment in which they live, behave and reproduce. An organism is simulated as having a body with a specific size, external shape, sensory and motor organs, etc., and an internal structure made up of a genotype, the nervous system, and other organs. Artificial organisms can be analyzed at the genetic level, at the level of the mapping from genotype to phenotype (development), at the neural and behavioral level, at the level of the effects of the network's output on the environment, at the level of the reproductive success of each individual (fitness), and at the level of populations of individuals and of entire ecosystems.

Examining organisms at various levels could be crucial for understanding their behavior, because often an explanation of what happens at one level can be found at another level (see, for example, [5]; [2]). In particular, one could hypothesize that the evolution of modularity results from the interaction among processes at different levels. In future work we will focus on the evolutionarily emergence of functionally different modules at the neural-behavioral level from gene duplication. We will try to test the hypothesis that the different origin and evolutionary history of modules that arise out of genetic duplication instead of being hardwired in the artificial organisms since the beginning of the evolutionary process results in modules endowed with a greater amount of functional meaning at the behavioral level.

References

[1] BONNER, John T., *The evolution of complexity*, Princeton University Press (1988).

[2] CALABRETTA, Raffaele, R. GALBIATI, S. NOLFI and D. PARISI, "Two is Better than One: a Diploid Genotype for Neural Networks", *Neural Processing Letters* **4** (1996), 149–155.

[3] FUTUYMA, Douglas, *Evolutionary biology*, Sinauer (1998).

[4] HOLLAND, John, *Adaptation in Natural and Artificial Systems*, University of Michigan Press (1975).

[5] MIGLINO, Orazio, S. NOLFI and D. PARISI, "Discontinuity in evolution: how different levels of organization imply pre-adaptation", *Adaptive Individuals in Evolving Populations* (Rick BELEW and Melanie MITCHELL eds.), Addison-Wesley (1996).

[6] MONDADA, Francesco, E. FRANZI and P. IENNE,"Mobile robot miniaturisation: a tool for investigation in control algorithms", *Experimental Robotics III, Lecture Notes in Control and Information Sciences* (T. YOSHIKAWA and F. MIYAZAKI eds.), Springer-Verlag (1994).

[7] NOLFI, Stefano, "Using emergent modularity to develop control systems for mobile robots", *Adaptive Behavior* **5** (1997), 343–363.

[8] NOLFI, Stefano, "Evolving non-trivial behaviors on real robots: a garbage collecting robot", *Robotics and Autonomous Systems* **22** (1997), 187–198.

[9] RAFF, Rudolph, *The shape of life. Genes, development, and the evolution of animal form*, University of Chicago Press (1996).

[10] RUMELHART, David and J. MCCLELLAND, *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1.*, MIT Press (1986).

[11] WAGNER, Günter, "Homologues, natural kinds and the evolution of modularity", *American Zoologist* **36** (1996), 36–43.

[12] WAGNER, Günter and L. ALTENBERG, "Complex adaptations and the evolution of evolvability", *Evolution* **50** (1996), 967–976.