

# What does it take to evolve behaviorally complex organisms?

Raffaele Calabretta<sup>a,\*</sup>, Andrea Di Ferdinando<sup>a,c</sup>,  
Günter P. Wagner<sup>b</sup>, Domenico Parisi<sup>a</sup>

<sup>a</sup> *Institute of Cognitive Sciences and Technologies, National Research Council, Rome, Italy*

<sup>b</sup> *Department of Ecology and Evolutionary Biology, Yale University, New Haven, CT, USA*

<sup>c</sup> *University of Rome "La Sapienza", Rome, Italy*

## Abstract

What genotypic features explain the evolvability of organisms that have to accomplish many different tasks? The genotype of behaviorally complex organisms may be more likely to encode modular neural architectures because neural modules dedicated to distinct tasks avoid neural interference, i.e. the arrival of conflicting messages for changing the value of connection weights during learning. However, if the connection weights for the various modules are genetically inherited, this raises the problem of genetic linkage: favorable mutations may fall on one portion of the genotype encoding one neural module and unfavorable mutations on another portion encoding another module. We show that this can prevent the genotype from reaching an adaptive optimum. This effect is different from other linkage effects described in the literature and we argue that it represents a new class of genetic constraints. Using simulations we show that sexual reproduction can alleviate the problem of genetic linkage by recombining separate modules all of which incorporate either favorable or unfavorable mutations. We speculate that this effect may contribute to the taxonomic prevalence of sexual reproduction among higher organisms. In addition to sexual recombination, the problem of genetic linkage for behaviorally complex organisms may be mitigated by entrusting evolution with the task of finding appropriate modular architectures and learning with the task of finding the appropriate connection weights for these architectures.

© 2002 Elsevier Science Ireland Ltd. All rights reserved.

*Keywords:* Genetic linkage; Neural interference; Modular neural networks; Genetic algorithms

## 1. Introduction

What does it take to evolve behaviorally complex organisms? We define a behaviorally complex organism as an organism which is capable of a variety of different behaviors or functions (“tasks”), i.e. input/output mappings, and in particular responds to one and the same sensory input by processing the input in a num-

ber of different ways. Of course, real animals do not have separate and well defined tasks and evolution is not concerned with optimizing specific tasks in real animals, but in our simulations we have adopted these simplifications to explore some hypotheses about the evolution of behaviorally complex organisms. What kind of genotypes explain the evolvability of such organisms? In this paper, we provide some answers to this question by simulating the evolution of behaviorally complex organisms using neural networks (Rumelhart and McClelland, 1986) to model the nervous system that controls the organisms’ behavior, and genetic algorithms (Holland, 1992) to model the

\* Corresponding author.

*E-mail addresses:* rcalabretta@ip.rm.cnr.it (R. Calabretta), andread@ip.rm.cnr.it (A.Di. Ferdinando), gunter.wagner@yale.edu (G.P. Wagner), parisi@ip.rm.cnr.it (D. Parisi).

evolutionary process that changes their genotypes in a succession of generations.

In [Section 2](#) we propose that, to avoid neural interference that may arise if the same connections are dedicated to different tasks, the accomplishment of different tasks may be easier with modular neural networks, that is, neural networks which are divided up into separate modules each dedicated to a particular task. In [Section 3](#) we introduce the notion of genetic linkage which explains why genotypes divided up into separate genetic modules encoding different neural modules may be inefficient because they may be subject to conflicting mutations falling on different but genetically linked modules. In [Section 4](#) we show that the problem of genetic linkage has a very general nature and it is not restricted to cases in which two or more tasks are of different difficulty and the evolution of the capacity to evolve the easier tasks first makes it impossible to subsequently evolve the more difficult tasks. In [Section 5](#) we consider the role of sexual reproduction in at least partially solving the problem of genetic linkage in that sexual recombination allows the selection process to more efficiently eliminate unfavorable mutations and retain favorable ones. In [Section 6](#) we provide a number of biological arguments for our interpretation of our results. In [Section 7](#) we draw some general conclusions and we advance the hypothesis that the evolution of behaviorally complex organisms requires genotypes that encode modular network architectures but entrust learning with the task of finding the appropriate connection weights for the different neural modules. The reason for these different roles of evolution and learning is that evolution is guided by a global evaluation signal (fitness) and it seems “not to care” about the specific capacities of individuals but only about their total performance, whereas learning during life can use distinct evaluation signals (teaching inputs) for each separate neural module.

## 2. Modularity in network architecture solves the problem of neural interference

Imagine an organism that must accomplish two different tasks in response to the same sensory input. The organism visually perceives an object and it must recognize both where the object is located (Where task) and what type of object it is (What task) ([Ungerleider](#)

and [Mishkin, 1982](#); notice, however, that subsequent work has contested [Ungerleider and Mishkin's](#) interpretation of the two tasks in terms of two separate pathways in the brain ([Milner and Goodale, 1995](#)). If the organism's nervous system is modeled by a non-modular neural network, neural interference can arise that will negatively affect the organism's performance. A nonmodular neural network is a network in which the same connections are involved in both the Where and What tasks. Neural interference can result from the fact that in learning the two tasks the weight of one of the connections may need to be increased to accomplish one task and decreased to accomplish the other task. If there is no way out of this conflicting situation, it may be impossible for the organism's neural network to acquire the appropriate connection weights for accomplishing both tasks adequately.

Neural modularity can be a solution to this problem. A modular network is a network in which some of the network's connections are dedicated to one task and play no role in the other task while some other connections are dedicated to the second task and are not used for the first task. This type of modularity solves the problem of neural interference because the weights of the connections that are proprietary for any single task can be increased or decreased to accomplish the task without interfering with the accomplishment of the other task.

[Rueckl et al. \(1989\)](#) have shown that a neural network that must recognize which of nine geometrically different objects is located in which of nine different spatial positions in a retina ([Fig. 1](#)) is better able to learn the task using the backpropagation algorithm if the network architecture is modular rather than non-modular. The visual input is contained in a  $5 \times 5$  cell retina ([Fig. 1a](#)) which maps into 25 input units. These 25 input units are all connected to each of 18 hidden units (lower connections). There are 18 output units, 9 encoding localistically the 9 different positions of the object in the retina (Where output units, [Fig. 1b](#)), and 9 encoding the 9 different objects (What output units, [Fig. 1c](#)).

What distinguishes the modular from the nonmodular architecture is the pattern of connections projecting from the 18 hidden units to the 18 output units (higher connections). In the nonmodular architecture all 18 hidden units project to each of the 18 output units. In the modular architecture 14 hidden units project only

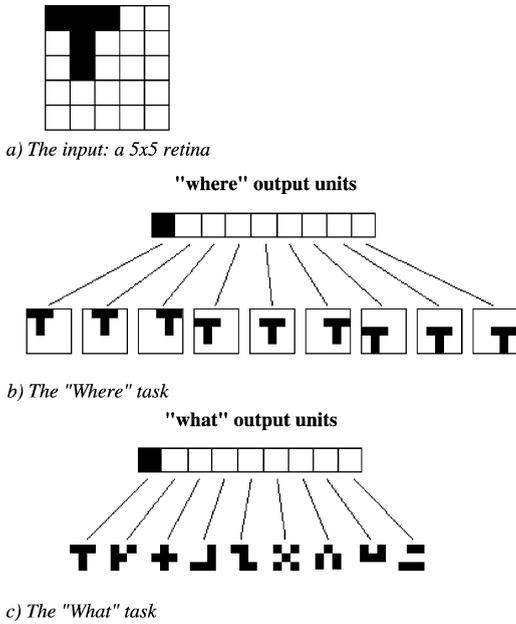


Fig. 1. The What and Where tasks. Retina (a), nine different locations (b), and nine different objects that can appear in each location (c).

to the 9 What output units and the remaining 4 hidden units project only to the 9 Where output units (Fig. 2). (More hidden units are dedicated to the What task than to the Where task because the What task is more complex than the Where task. See Ruckl et al., 1989. We will come back to this further.) Therefore, in the nonmodular architecture all the lower connections are involved in both the What and Where tasks, and this may result in neural interference, whereas in the modular architecture some of the lower connections are

responsible only for the What task and the remaining connections only for the Where task—and this rules out neural interference. In fact, Ruckl et al. have found that a neural network trained with the backpropagation procedure learns the two tasks much better if its architecture is modular rather than nonmodular.

In Ruckl et al.'s simulation the network architecture is fixed and hardwired by the researcher. Di Ferdinando et al. (2001) have done a series of simulations in which the network architecture for the What and Where tasks is not decided by the researcher but it evolves in a population of networks using a genetic algorithm. In some of these simulations the connection weights for the particular genetically inherited architecture are learned during each individual network's "life" using the backpropagation algorithm, exactly as in Ruckl et al.'s simulations. The simulation begins with a population of 100 networks each with a randomly generated pattern of higher connections. Starting with random connection weights which are assigned at birth to each of the 100 individuals, each individual learns the connection weights for the What and Where tasks using the backpropagation algorithm and is assigned a fitness value (sum of the two "errors" in the two tasks, with a minus sign) at the end of life/training. The 20 individuals with highest fitness (smallest total error) are selected for reproduction and each of them generates 5 offspring. The  $20 \times 5 = 100$  new individuals constitute the next generation. Each offspring inherits the same network architecture of its single parent (reproduction is asexual) but not its parent's connection weights. The inherited network architecture is subject to random mutations that can modify the network's pattern

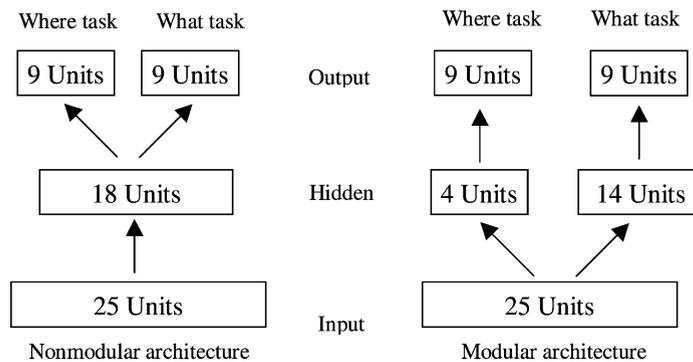


Fig. 2. Modular and nonmodular architectures for the What and Where tasks.

of higher connections, i.e. the connections between the hidden units and the output units. After a certain number of generations the average fitness is very good and comparable to Rueckl et al.'s results. What is of interest from our present perspective is that the evolved architecture that emerges at the end of evolution approaches Rueckl et al.'s modular architecture with 14 hidden units dedicated to the What task and 4 hidden units dedicated to the Where task. In other words, evolution spontaneously finds the same modular network architecture that Rueckl et al. have found to be the optimal one for the What and Where tasks.

### 3. Genetic interference

Imagine now we entrust evolution with both the task of finding the best network architecture for the What and Where tasks and the task of finding the appropriate connection weights for this architecture. In other words, unlike the preceding simulation everything is genetically inherited, both the network architecture and the connection weights, and there is no learning during life. Under these conditions it turns out to be impossible to create neural networks that are able to solve both the What and Where tasks (Di Ferdinando et al., 2001). The error on the two tasks at the end of the simulation remains substantial. This result can be explained by the fact that some particular set of connection weights is an appropriate set of connection weights allowing the network to solve both tasks only with respect to some particular network architecture, not with respect to any network architecture. In some of Di Ferdinando et al.'s simulations the inherited genotype encodes both the network architecture and the connection weights for the architecture and, when an individual inherits a particular genotype, this implies that the network architecture and the connection weight encoded in the genotype are well adapted to each other, otherwise the individual's parent would have not been selected for reproduction. However, if a mutation changes the inherited network architecture, the associated set of connection weights encoded in the same genotype may suddenly cease to be appropriate for the changed architecture. This may explain why the genetic algorithm is unable to find a solution for both the What and Where tasks if everything

is genetically inherited and evolution must identify both the appropriate network architecture and the appropriate connection weights for this architecture.

Another explanation of this result might be that "in the initial generations the algorithm concentrates on the easier task, the Where task, and dedicates many computational resources (hidden units) to this task. When the performance on this task is almost perfect, however, the algorithm is unable to shift computational resources from the Where task to the more difficult What task" (Di Ferdinando et al., 2001).

However, even in the few cases in which evolution is able to find the appropriate network architecture for the What and Where tasks, evolution is unable to also find the appropriate connection weights for this architecture. In the few replications of the simulation in which the evolved architecture approaches the optimal one (14 hidden units for the What module and 4 hidden units for the Where module), the terminal error does not approach zero (Di Ferdinando et al., 2001). This may indicate that the problem goes beyond the fact that mutations can disrupt an evolved co-adaptation of network architecture and connection weights for this architecture. Evolution may be more generally unable to find the appropriate connection weights for modular architectures.

We can lighten evolution's burden if we entrust evolution only with the task of finding the appropriate connection weights for a fixed, and optimal, network architecture. In other words, inherited genotypes encode only the network's connection weights while the network architecture is fixed (it is the optimal network architecture with 14 hidden units for the What task and 4 hidden units for the Where task) and therefore is not encoded in the genotype and does not evolve. This approach to constructing artificial organisms that are capable to solve particular tasks has been used successfully in many simulations (see, e.g. Yao, 1999). In these simulations evolution has been repeatedly shown to be able to find the appropriate connection weights for a fixed network architecture as an alternative to the backpropagation procedure. However, in most published simulations that have used the genetic algorithm to evolve the connection weights for fixed architectures, the organisms have to solve only one task and therefore their network architecture tends to be non-modular. Will evolution be able to find the appropriate connection weights for a fixed modular architecture

controlling the behavior of organisms that must solve both the What and the Where tasks?

We have realized a set of simulations in which the neural network architecture is fixed and is the optimal modular one for the What and Where tasks (14 hidden units for the What task and 4 hidden units for the Where task) and the genetic algorithm needs only to evolve the appropriate connection weights for this architecture. (Unless otherwise specified, all simulations reported in this paper use a mutation rate of 10%, i.e. there is a probability of 10% for each connection weight to be mutated by adding a quantity randomly chosen in the interval between  $-1$  and  $+1$  to the weight value, which is encoded as a real number. Moreover, all the results represent the average of 10 replications of each simulation.) We have compared the results obtained using this architecture with those obtained with a fixed nonmodular architecture.

As shown in Fig. 3, for both the modular and the nonmodular network architecture the genetic algorithm is unable to find the appropriate weights for

the two tasks. It is important to notice that while what we have called neural interference is present in the nonmodular architecture because all the lower connections in this architecture serve both the Where and the What tasks, this is not true for the modular architecture. In the modular architecture distinct sets of connections are used for the two tasks and they can be separately adjusted by the evolutionary process without reciprocal neural interference. Hence, neural interference cannot be invoked to explain the bad results obtained by the genetic algorithm.

We believe that these results may be caused by another type of interference, this time not at the neural but at the genetic level. This type of genetic interference occurs because different portions of the genotype, encoding different neural modules, are linked together in the same genome and are inherited together (i.e. the phenomenon of genetic linkage). Why do we invoke genetic linkage as an explanation for our negative results with modular architectures? We know that when the connection weights of a modular architecture are

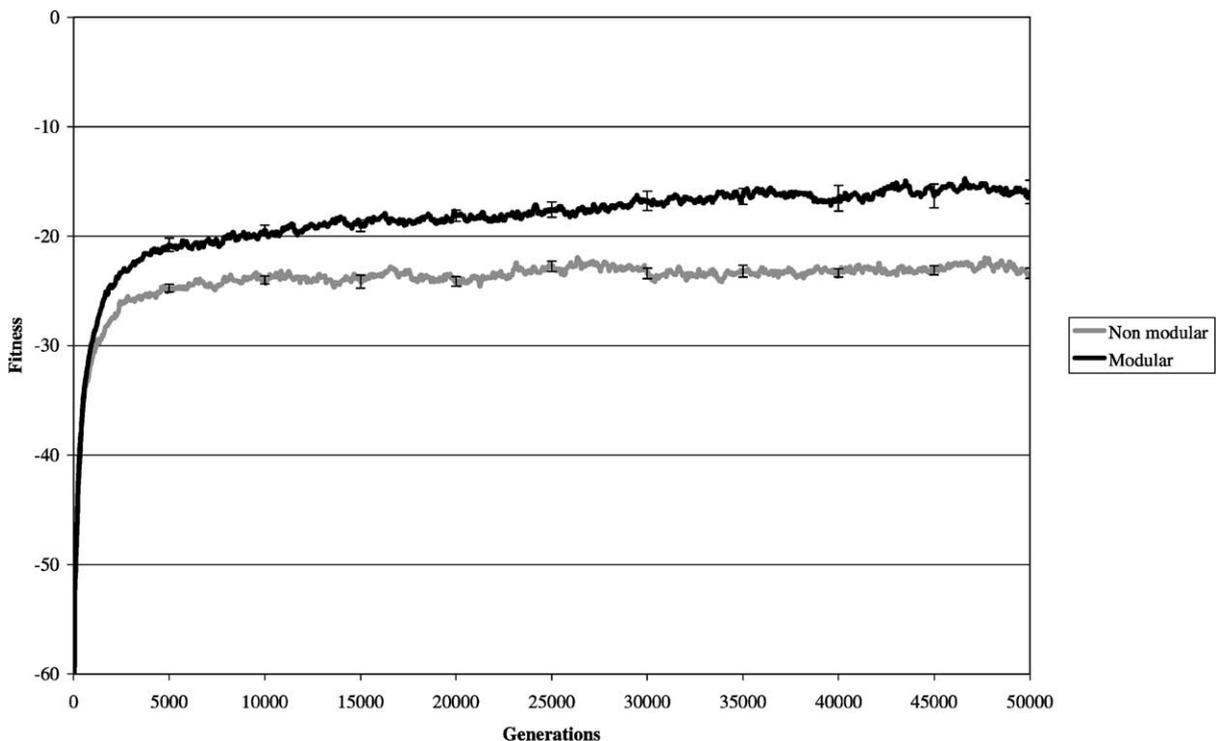


Fig. 3. Average fitness for the What and Where tasks across 50,000 generations for modular and nonmodular architectures when network architecture is fixed and what evolves is only the set of connection weights for the architecture. (In all figures S.E. are also shown.)

adjusted using the backpropagation algorithm this type of learning procedure is able to find the appropriate connection weights for both the Where and What tasks and the total error approximates zero (Rueckl et al., 1989). This appears to be due to the fact that when one is using the backpropagation algorithm it is as if one were using two separate learning processes, one for the Where task and the other for the What task. In each cycle the backpropagation algorithm computes two separate errors, one for each neural module, and it modifies the connection weights of each module separately on the basis of the appropriate error. The total error is simply the sum of the two separate errors for the two tasks.

Two separate errors are computed also if one uses the genetic algorithm instead of the backpropagation algorithm and the connection weights are evolved rather than learned. But there is a critical difference between the two algorithms. When one uses the backpropagation learning algorithm each separate neural module is separately informed of the network's performance on each distinct task. On the contrary, when one uses the genetic algorithm the fitness formula which is used to determine which individuals reproduce and which individuals do not reproduce represents a global evaluation of each individual, summing together the error in the What task and the error in the Where task. This creates the conditions for a form of genetic interference caused by genetic linkage.

Consider what may happen when a genotype which is divided up into two separate portions, one encoding the connection weights for the What neural module and the other one the connection weights for the Where module, is reproduced and therefore is subject to random genetic mutations. Imagine that a favorable genetic mutation changes the What portion of the genotype resulting in an improved performance of the What neural module. If this is the only genetic mutation affecting the particular genotype, the global performance of the individual will improve, the individual will be more likely to reproduce, and the favorable mutation will be retained in the population. However, consider the case in which a favorable mutation falls on the What portion of the individual's genotype and at the same time an unfavorable mutation falls on the Where portion of the same genotype. Now there is a problem. The selection process may be unable to "see" the advantageous mutation. If the

overall fitness effect of both mutations is deleterious, the chromosome with the beneficial mutation will be eliminated by natural selection. Hence, if the rate and magnitude of deleterious mutations is high enough it may prevent the adaptation of a character, in this case the What task (Fig. 4).

A direct test of our interpretation of our results in terms of genetic linkage consists in eliminating genetic linkage and evaluating the performance of the two tasks separately. We have conducted a new set of simulations in which the genetic algorithm is used to evolve two separate populations of individuals, one in which each individual has to solve only the What task and the other one with individuals that have to solve only the Where task. As we have already noted above, when one uses the backpropagation learning algorithm there is no interference between the two tasks (in modular architectures) because the connection weights of the What neural module are adjusted only on the basis of the error on the What task, and similarly for the Where task. We have recreated these conditions with the genetic algorithm by setting up two distinct populations. The genotype of one population encodes the connection weights only for the What neural module and the individuals in this population reproduce only on the basis of their performance on the What task, and the same is true for the other population with respect to the Where task. Since each population is evaluated only for one task genetic interference within the same genotype is impossible.

In other words, we have realized two distinct simulations. In one simulation the population has the neural architecture which has been shown to be appropriate for the What task (25 input units, 14 hidden units, 9 output units) and the fitness formula reflects an individual's performance only on the What task, while in the other simulation the population has a neural architecture which is appropriate for the Where task (25 inputs units, 4 hidden units, 9 output units) and the fitness formula captures only an individual's performance on this task. In these circumstances there is no genetic interference between the two tasks and the genetic algorithm is able to evolve the connection weights appropriate for both the Where and the What tasks. The fitness for the "two populations" condition, that is, the sum of the fitness of the What population and the fitness of the Where population, is near zero (zero error in both tasks) (Fig. 5).

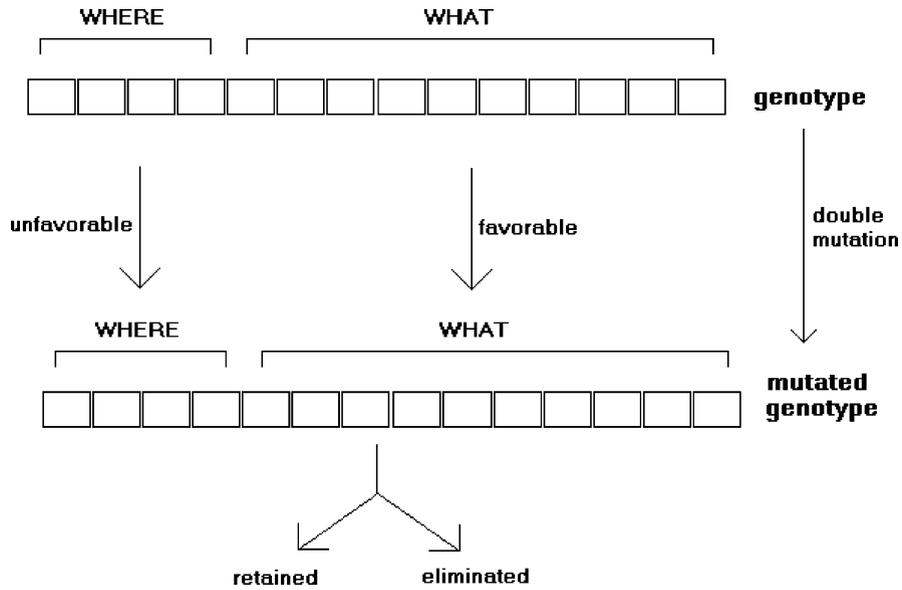


Fig. 4. Genetic interference. Conflicting mutations (one favorable and the other one unfavorable) can fall on separate portions of the genotype encoding distinct neural modules, thereby hindering the efficiency of the selection process.

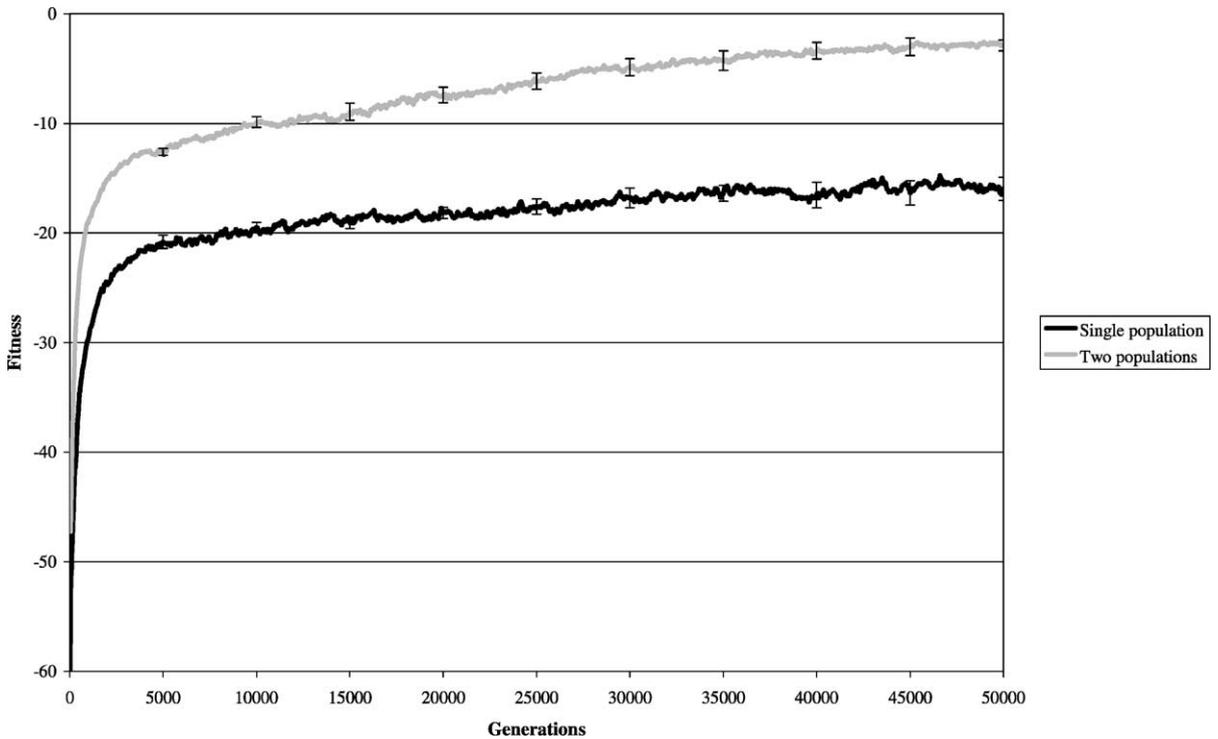


Fig. 5. Average fitness (error) across 50,000 generations when the genotype encodes both the Where and What tasks (single population) and when it encodes either one or the other of the two tasks (two populations). (The fitness for the “two populations” condition is the sum of the fitness of the What population and the fitness of the Where population.)

Another test of our interpretation of our results in terms of genetic linkage consists in manipulating the mutation rate. If our interpretation is correct we expect that increasing the mutation rate should lead to greater unfavorable consequences. In fact, given that mutations tend to be generally unfavorable, increasing the mutation rate should lead to a decrease of the probability that their total effect in both modules will be positive. To test this prediction we have run another set of simulations in which we have systematically increased the mutation rate from a value of 0.0016% to a value of 10% in different simulations, both for populations in which an individual has to perform both the What and Where tasks and for separate populations in which individuals perform only one of the two tasks.

The results of these simulations are shown in Fig. 6. For both conditions there is an increase in fitness when mutation rates increase from a very low value of 0.0016% to a value of 0.3%. With a mutation rate

of 0.3% both tasks are solved in both conditions. Beyond this value, however, the equilibrium fitness decreases. What is interesting for our purposes is that, while the fitness is identical in the two conditions with very low mutation rates, the decrease in fitness beyond the threshold mutation rate of 0.3% is much greater for populations in which the What and Where neural modules are both encoded in the same genotype than in populations in which an individual's genotype encodes either only the What module or only the Where module. This seems to support our interpretation of our results in terms of genetic interference. When two distinct genetic modules, underlying two distinct neural modules and two distinct tasks, are present in the same genotype, genetic mutations can result in a more adapted module for one task but at the same time they can produce a less adapted module for the other task, and the probability that this will happen increases with increasing mutation rates.

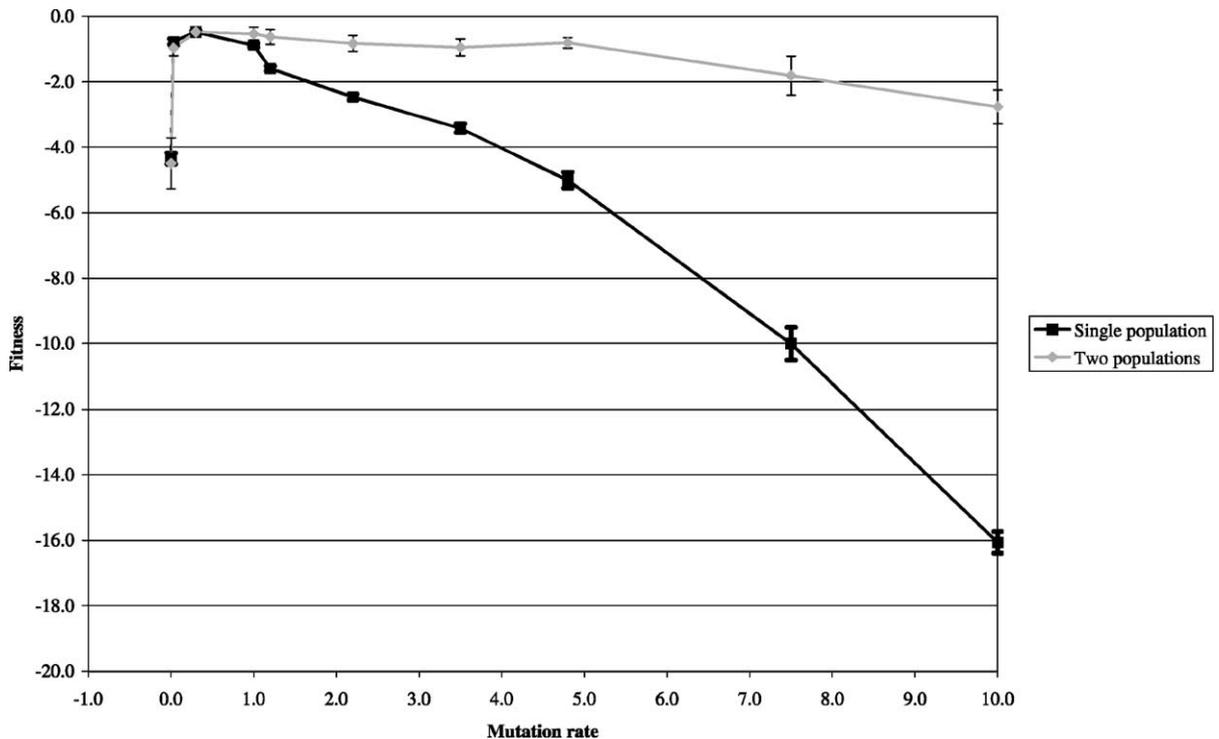


Fig. 6. Average fitness at the end of evolution for populations with increasing mutations rates when the genotype encodes both the Where and What tasks (single population) and when it encodes either one or the other of the two tasks (two populations). (The fitness for the “two populations” condition is the sum of the fitness of the What population and the fitness of the Where population.)

#### 4. Can genetic interference be explained by the fact that the what task is more difficult than the where task?

As Rueckl et al. (1989) note, and as is confirmed by the results of all our simulations, an individual that must solve both the What and Where tasks must not only solve two tasks simultaneously in response to the same input but it must solve two tasks that differ in complexity or difficulty. For reasons discussed in Rueckl et al., the What task is a more complex or difficult task than the Where task. A consequence of this fact is that, if the conditions allow for the acquisition of both tasks, both tasks are acquired equally well, but if neural interference (in nonmodular architectures) or genetic interference (the connection weights of the two neural modules are encoded in the same genotype) make it impossible to adequately solve both tasks, it is the more difficult What task which suffers. In other words, at the end of the simulation the neural networks are able to solve the Where task (the error for this task is near zero) but not

the What task (the error for the What task remains substantial).

The differential difficulty of the two tasks manifests itself in the fact that the easier Where task is acquired (evolved or learned) first and it rather quickly reaches almost perfect performance. When it is time to acquire the more difficult What task, however, it turns out to be impossible to reach perfect performance in the later acquired What task. Many simulations using the backpropagation algorithm have shown that neural networks exhibit what is sometimes called the “age of acquisition” effect: at the end of the simulation performance on tasks that are learned earlier than other tasks is better than performance on later acquired tasks (Ellis and Lambon Ralph, 2000; Smith et al., *in press*). This “age of acquisition” effect is observed also in our simulations in which populations of neural networks evolve the ability to solve the What and Where tasks, as can be seen by plotting separately the fitness curves for the Where and the What tasks in the modular architecture (Fig. 7). The Where task is acquired earlier than the What task in these evolving populations in

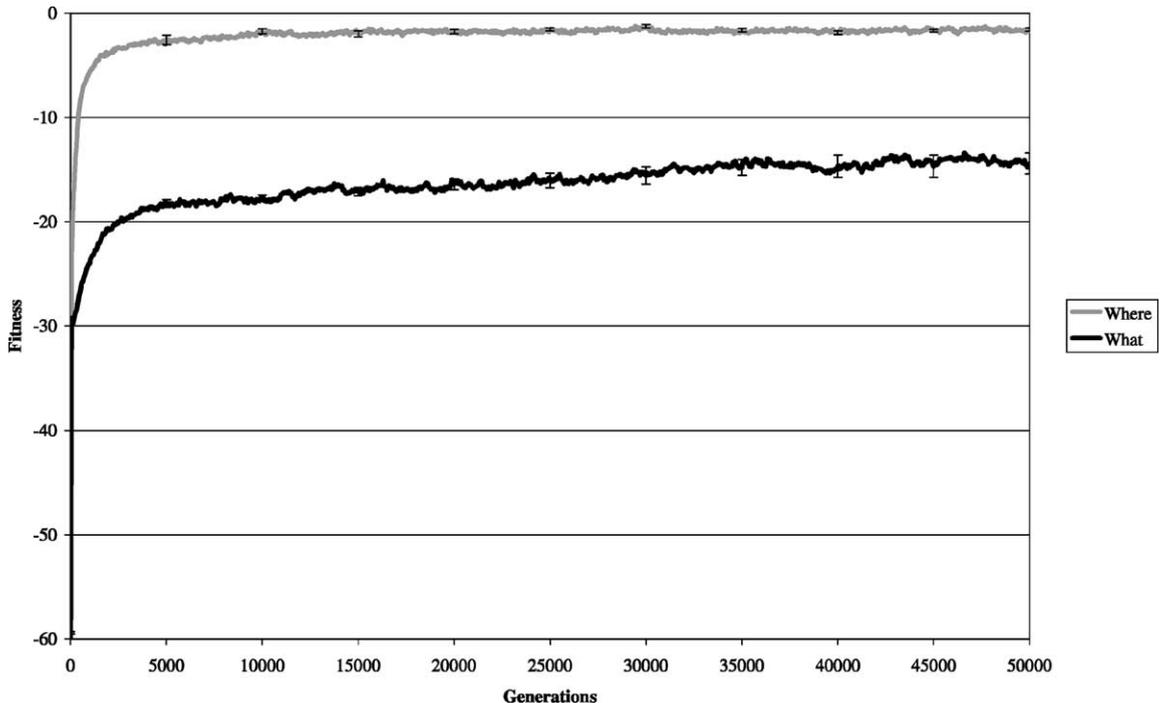


Fig. 7. Separate plotting of fitness curves for the Where and the What tasks in the modular architecture. (Same data of the higher curve in Fig. 3).

the sense that the error on the Where task goes down more rapidly (in the early generations) than the error on the What task. If we look at which individuals are selected for reproduction at various evolutionary stages, we see that in the early generations individuals are selected for reproduction in terms of their error in the Where task while their error in the What task is ignored. Only when, in the later generations, error in the Where task is so low that differences among individuals in this error become negligible, the error on the What task becomes the criterion on the basis of which individuals are selected for reproduction. But at this point it is too late for acquiring the What task sufficiently well and at the end of evolution the error on the What task remains higher than the error on the Where task.

One might argue that genetic interference is restricted to cases in which one task is acquired before another task because one task is easier than the other task. To test this hypothesis we have conducted two further sets of simulations.

One way to re-establish a parity between the What and Where tasks is to assign more adaptive value to the more difficult What task than to the easier Where task. The What task remains an intrinsically more difficult task than the Where task but if we arbitrarily decide that the What task is more important in terms of reproductive chances than the Where task, this greater adaptive importance of the What task may compensate for its greater difficulty and both tasks may be acquired together and equally well. In the new situation the tendency of evolution to concentrate on the easier Where task in the early evolutionary stages would be compensated by the need to pay attention from the beginning to the What task which is more difficult than the Where task but is also more important for reproduction.

To change the relative fitness contribution of the Where and What tasks we have modified the fitness formula used to select the individuals for reproduction. In the fitness formula used so far the error in the Where task is simply added to the error in the What task to yield the total error. This total error, with the minus sign, represents the fitness of an individual which decides whether or not the individual is one of the 20 reproducing individuals. In a new set of simulations we have changed the fitness formula to give more weight to the error in the What task compared to the weight of the error in the Where task. The new fitness formula

is the following:

$$\text{total fitness} = (5 \times \text{What fitness}) + (\text{Where fitness})$$

With the new fitness formula the What task, even if it is more difficult than the Where task, becomes more important (in fact, five times more important) in deciding which individuals are selected for reproduction. In fact, the results of our simulations show that evolution first concentrates on the What task until high levels of performance in the What task are reached. However, when, later on, evolution turns to the Where task it is too late to learn the Where task (Fig. 8).

It appears then that a sort of “loss of plasticity” prevents evolution from reaching good levels of performance in the Where task when the What task has been learned first. The situation is similar to that of the preceding simulations, but reversed. In the preceding simulations the two tasks were equivalent in terms of importance for reproduction but, since the Where task is intrinsically easier than the What task, the Where task was acquired first and then it was too late for acquiring the What task equally well. In the new simulations, since the What task has become much more important than the Where task from the point of view of reproductive chances, even if it is more difficult than the Where task it is acquired earlier than the Where task so that later in evolution it becomes impossible to reach adequate levels of performance in the Where task.

The simulation just described seems to show that “age of acquisition” is a critical factor in evolution and not only in learning. A more balanced solution for the problem of accomplishing both the What and Where tasks can be obtained if the more difficult What task is given more adaptive weight than the easier Where task but not too much more adaptive weight. In fact, the results of another simulation in which we adopt the following fitness formula:

$$\text{total fitness} = (3.5 \times \text{What fitness}) + (\text{Where fitness})$$

show that we can obtain a more balanced increase in fitness for both the What and Where tasks from the beginning of evolution. (The value of 3.5 was chosen because it reflects the ratio between 14 What hidden units and 4 Where hidden units.) However, even if the What and Where tasks are acquired almost together, genetic interference prevents the reaching of a satisfactory terminal error on both tasks (Fig. 9).

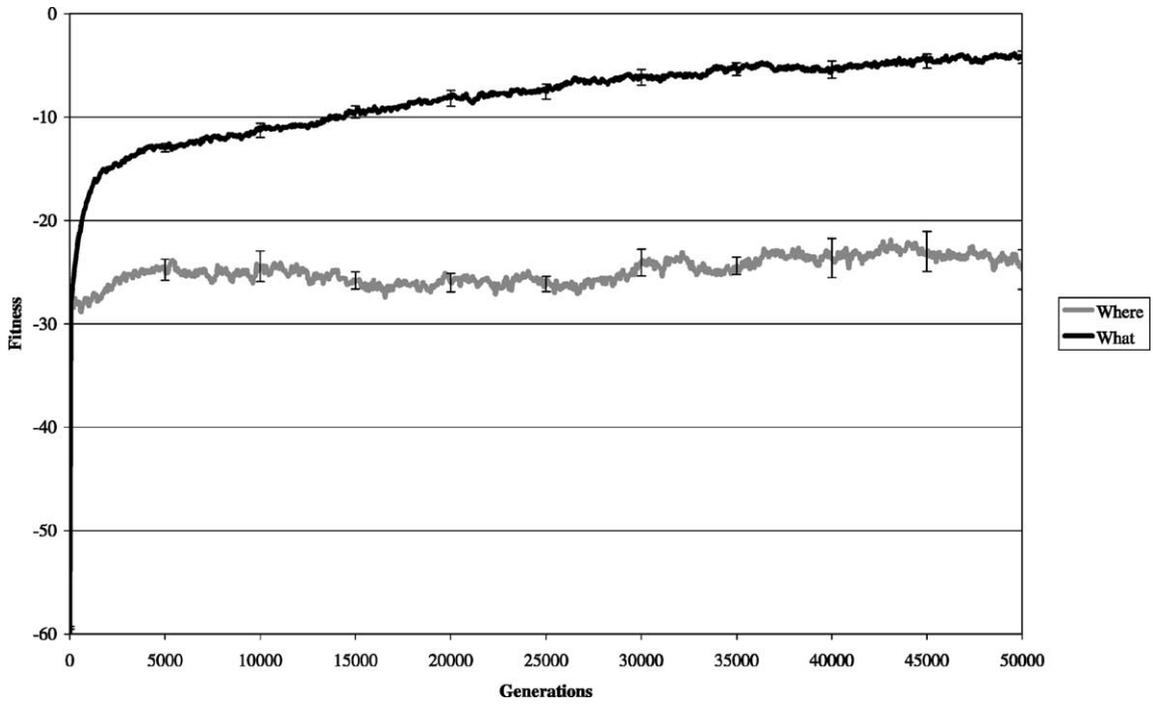


Fig. 8. Average fitness across 50,000 generations separately for the What and Where tasks when the What task is acquired earlier than the Where task because, by manipulating the fitness formula, the What task has become adaptively more important than the Where task.

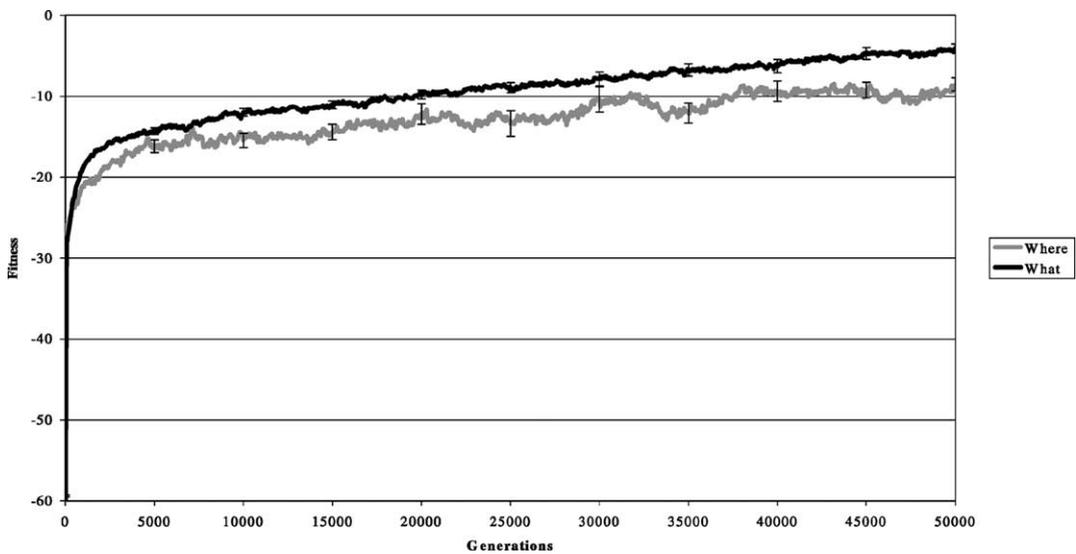


Fig. 9. Average fitness across 50,000 generations separately for the What and Where tasks when both tasks evolve almost together because, by manipulating the fitness formula, the greater adaptive value of the What task compensates for the greater difficulty of the What task compared to the Where task.

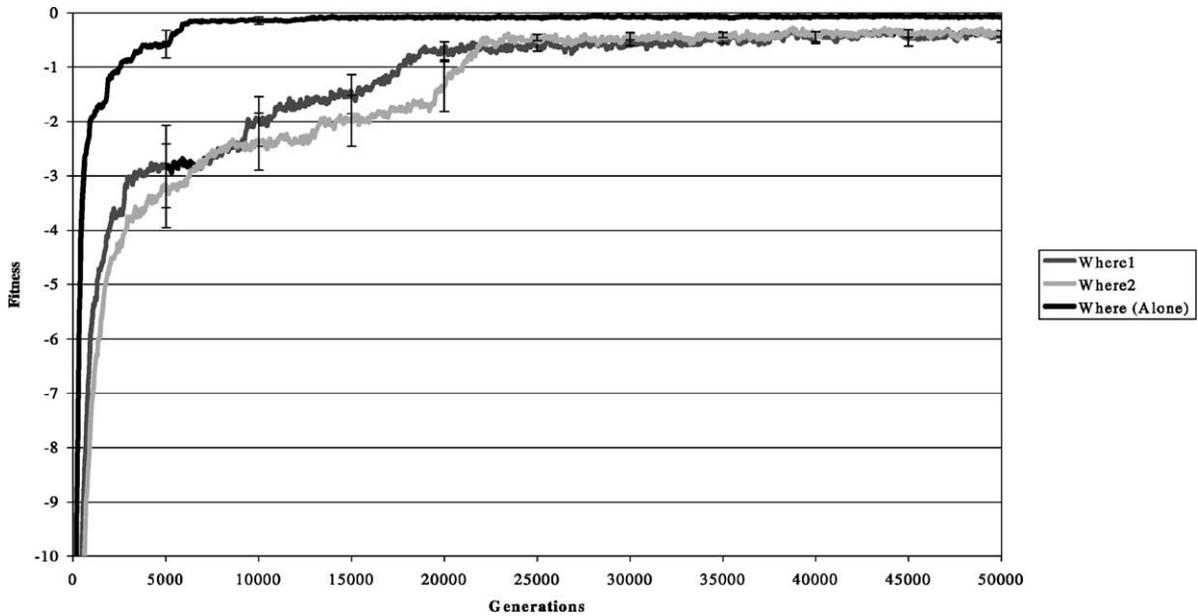


Fig. 10. Average fitness across 50,000 generations in a population that has to solve two Where tasks compared with a population that has to solve a single Where task. (Notice that the fitness in this figure is scaled differently from the other figures to show the small differences among the curves).

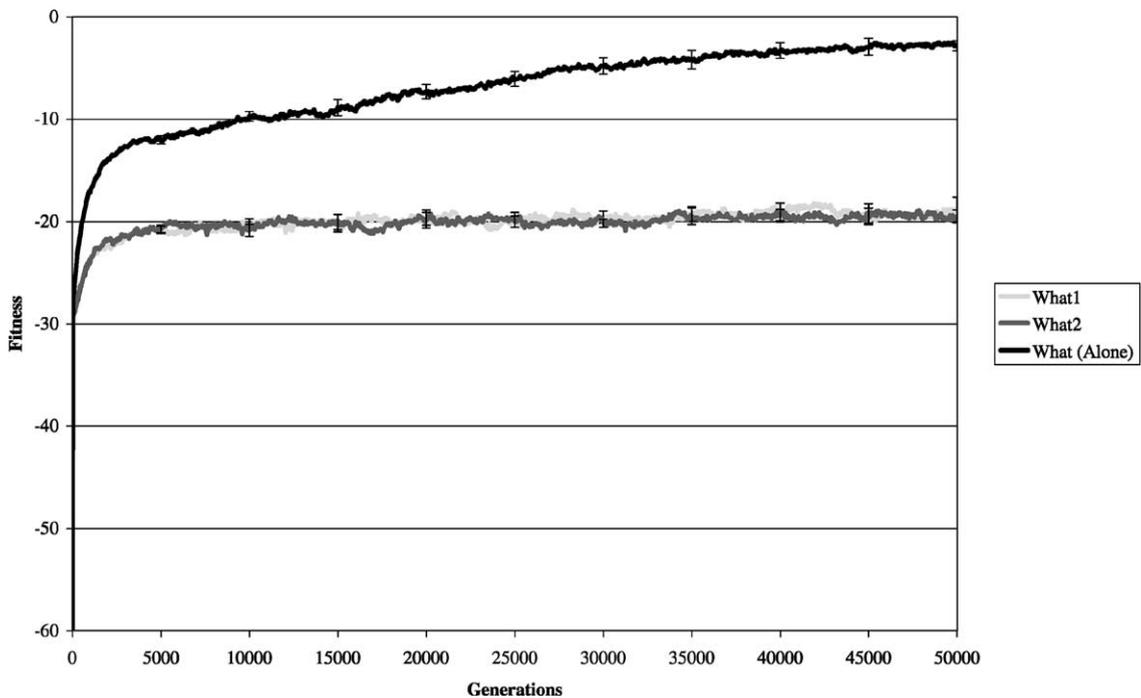


Fig. 11. Average fitness across 50,000 generations in a population that has to solve two What tasks compared with a population that has to solve a single What task.

A more direct demonstration that genetic interference is a general phenomenon and is not restricted to cases in which different tasks, because of their different difficulty or for other reasons, are acquired at different evolutionary stages, can be obtained with an even simpler manipulation of our simulation scenario. We have conducted another set of simulations in which an organism has to solve two tasks but the two tasks are identical. In other words, in one population organisms must solve the Where task twice in response to the same input and in another population the organisms must solve twice the What task. In the twice/Where population the network architecture includes two neural modules, each with 4 hidden units, and two separate sets of Where output units. The same for the twice/What population, with two identical What neural modules with 14 hidden units each. The fitness of an individual is based on the sum of the two Where errors or the two What errors.

Notice that in the new simulations genetic interference can occur because inherited genotypes encode two separate sets of connection weights, one for each of the two identical Where or What modules. However, since the two tasks that must be evolved in each population are by definition of identical difficulty, the different nature (difficulty) of the two tasks cannot be invoked to explain the effects of genetic interference. The results of these simulations show that, even when the two tasks that must be solved are of identical difficulty and both tasks evolve together, genetic interference hinders evolution (Figs. 10 and 11).

### 5. Can sexual reproduction solve the problem of genetic interference?

In all simulations described so far reproduction is asexual: an offspring inherits the genotype of its single parent. If reproduction is asexual, genetic interference can operate undisturbed. The offspring's genotype is the same genotype of its single parent, except for mutations. If conflicting mutations fall on distinct genetic modules, evolution is hindered. However, sexual reproduction can make a difference. In sexual reproduction portions of the genotype of one parent are recombined with portions of the genotype of the other parent. This can allow the recombination, in the offspring's genotype, of portions of genotypes

which have all been favorably mutated, and reduce the negative consequences of genetic linkage. If this is true, this can contribute to explaining the evolutionary advantage of recombination since recombination reduces the negative effects of linkage and make it possible to evolve more complex organisms that are capable of multiple tasks.

To test this hypothesis we have changed the reproductive mechanism of our populations from asexual to sexual. In all our simulations the encoding in the genotype of the connection weights for modular networks is as follows. The genotype is a sequence of 8 segments.

- Segment 1: bias weights for Where hidden units (4).
- Segment 2: bias weights for What hidden units (14).
- Segment 3: bias weights for Where output units (9).
- Segment 4: bias weights for What output units (9).
- Segment 5: connection weights between input units and Where hidden units (100).
- Segment 6: connection weights between input units and What hidden units (350).
- Segment 7: connection weights between Where hidden units and Where output units (36).
- Segment 8: connection weights between What hidden units and What output units (126).

In sexual reproduction the sequence of segments of the two genotypes selected for reproduction is cut at some randomly selected but corresponding place and two complementary pieces of the two genotypes are recombined together to produce the offspring's genotype. The 20 individuals with highest fitness in each generation are selected for reproduction. For each of these 20 individuals we randomly select 5 sexual partners among the remaining 19 individuals and the individual generates a single offspring with each partner. (Notice that during sexual recombination two genetic portions from the same parent can be recombined together.)

The sexually reproducing population performs significantly better than the asexual population (Fig. 12). The results of the simulations show that sexual reproduction limits, even if it does not completely eliminate (compare Figs. 5 and 12), the negative consequences of genetic linkage.

If reproduction is asexual and conflicting mutations occur in the same genotype, favorable and unfavorable mutations are necessarily linked together and

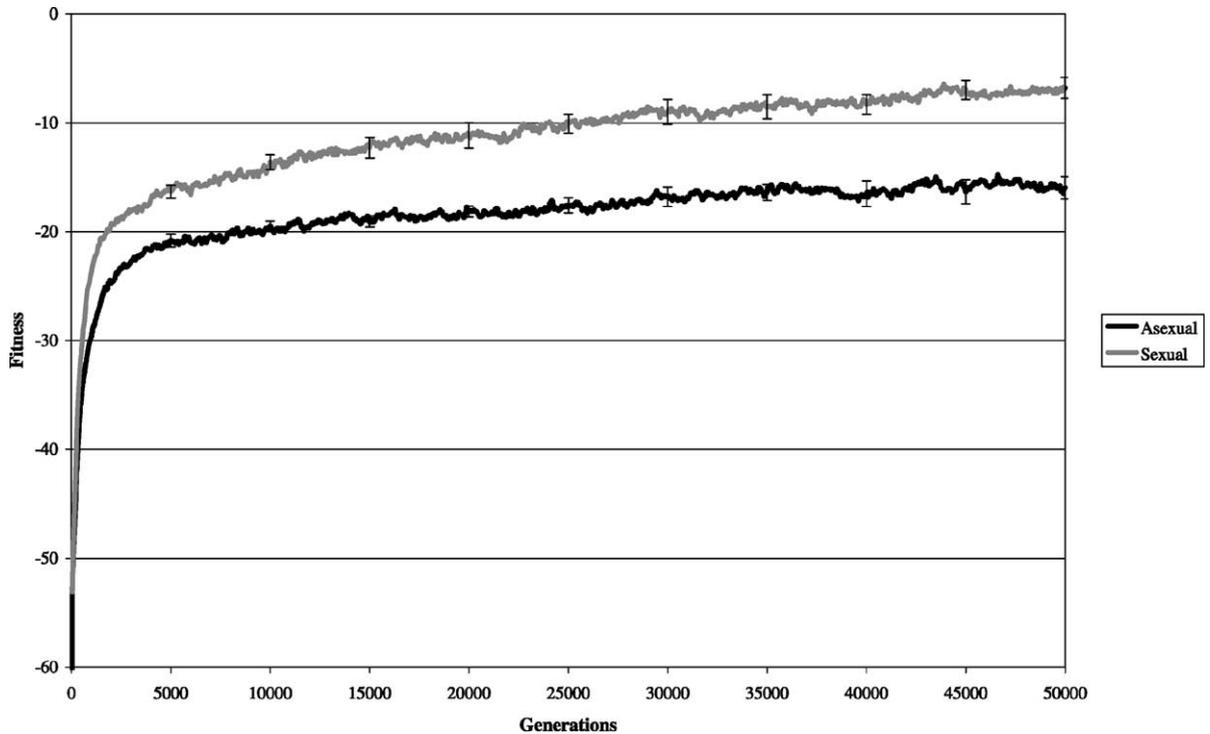


Fig. 12. Sexual reproduction leads to a greater efficiency of the evolutionary process by limiting the negative consequences of genetic linkage.

there is no way to retain the favorable mutations while eliminating the unfavorable mutations. Instead, this is exactly what sexual reproduction makes possible. In sexually reproducing organisms the sexual recombination of portions of different genotypes allows pulling apart portions of the same genotype on which conflicting mutations have operated and putting together, in the offspring's genotype, portions of different genotypes in which there are no conflicting mutations. Sexual reproduction can also facilitate the elimination of negative mutations. If two portions of genotypes that have both been changed by unfavorable mutations are recombined together in the same genotype, the new genotype will more likely be eliminated from the population (see Futuyma, 1998, p. 610).

## 6. Genetic interference, a new consequence of linkage?

We conclude from the results described above that genetic interference is caused by the linkage between

advantageous and deleterious mutations. There are a number of well understood negative consequences of genetic linkage believed to contribute to the prevalence of sexual species, especially among higher organisms. Examples are the degeneration of the genotype due to the accumulation of deleterious mutations, i.e. Mullers ratchet (Felsenstein, 1974; Haigh, 1978), and the interference between the selection of advantageous mutations ("speed limit" of adaptation; De Visser et al., 1999; Waxman and Peck, 1999). We thus have to ask whether the phenomenon described here is different from other consequences of genetic linkage.

There is one distinguishing feature of our model that differs from any other model of genetic linkage. In our model the two phenotypic characters measure the functional performance of two different tasks or functions. In all models we are aware of there is no distinction of different functions. Either the model only talks about genotypes and their fitness (Felsenstein, 1974; Haigh, 1978) or different phenotypic characters are modelled with their fitness (Wagner and Gabriel, 1990;

Waxman and Peck, 1999). The fact that our model explicitly refers to different functions rather than just fitness or quantitative characters in general is significant because interference may tend to occur when one of the tasks is already optimised (age of acquisition effect). Genetic interference is thus not a consequence of the co-occurrence of advantageous and disadvantageous mutations per se, but mutations affecting different functions. Below we propose a simple model of why we think this is important.

Most models of quantitative genetic characters do not impose limits to the possible character states. They are assumed to be points on the real number line between plus and minus infinity. In contrast, functional performance in the context of our model is measured on a closed interval. The task is either performed not at all, or perfectly, or to some degree in between. There is no further improvement possible beyond always correctly identifying the location or identity of the object in the Where or What tasks. The probability of advantageous and deleterious mutations, however, can not remain constant across the range of character values. If task performance is perfect, all mutations have to be either neutral or deleterious. Assuming that neutrality is not exceptionally high for genotypes with perfect performance of, say, task A, the probability of deleterious mutations to task A has to increase as task A approaches its optimum. From this it follows that as task A improves, the probability of an advantageous mutation for task B to co-occur with a deleterious mutation for task A is increasing. Depending on the distributions of deleterious mutations for the optimised task A and advantageous mutations for task B, the probability that a haplotype has an improved fitness may become very small. Once that happens, task B cannot be improved anymore, unless the linkage between the deleterious mutations for task A and the advantageous mutations for task B is broken, as for instance by recombination or in the simulations where two separate populations evolve the two tasks.

The form of genetic interference explained above requires that mutation frequency is high enough such that an advantageous mutation gets associated with a deleterious mutation in the other task before it reaches fixation. This is shown by the fact that genetic interference occurs only above a critical mutation rate (Fig. 6). In addition it seems to be necessary that the joint distribution of advantageous and deleterious mu-

tations is such that the combined effect of them is unlikely to lead to positive selection coefficients when one of the tasks is optimised. This requires information about the distribution of these effects, and thus is not predicable from first principles of population genetics alone. Consequently it is not possible to predict the critical mutation rate without extensive analysis of the mutational effect distributions.

One may argue that this explanation of genetic interference is suspect since it depends on the assumption of a closed interval of character values, rather than an open scale of possible character states. The reason is that any closed interval can be mapped onto the real number line with limits plus and minus infinity. This is true but does not affect the explanation for genetic interference put forward above. The reason is that any such transformation will also affect the mutational effect distribution and thus the rationale put forward above would not change. As the average genotype would assume larger character values on the new scale, the probability of deleterious mutations would still increase, since this property needs to be unaffected by the scale transformation. We thus think that the model sketched above is scale independent.

In summary, genetic interference as describe here is a phenomenon that requires well defined functional tasks with more or less independent contributions to fitness and it is thus a phenomenon different from the effect of deleterious pleiotropic effects. We conclude that genetic interference is a new population genetic mechanism that may contribute to the genetic and adaptive advantages of sexual reproduction.

## 7. Discussion

In the introduction of this paper we have asked the question “What does it take to evolve behaviorally complex organisms?”, that is, organisms whose adaptive pattern requires the accomplishment of many different tasks with independent contributions to fitness. If our simulation results have captured essential properties of biological evolution, then the following principles may have relevance to biological evolvability and they may allow us to propose some answers to this question. The evolvability of behaviorally complex organisms may favour genotypes with the following properties.

1. Since the accomplishment of many different tasks by the same organism requires modular neural networks to avoid neural interference, the genotype for complex organisms must encode modular neural networks. In our simulations the network architecture is entirely encoded in the genotype and it is already there at birth. In real organisms the modular network architecture can develop during an individual's lifetime as a result of an interaction between inherited genetic information and experience. (For a general discussion on the origin of evolutionary and developmental modules see [Wagner et al., in press](#); [Calabretta and Parisi, in press](#)).
2. Genotypes that encode modular neural networks are subject to genetic interference due to genetic linkage. We have defined genetic interference as the possibility that favorable mutations can fall on some genetic module encoded in the genotype while unfavorable mutations fall on another genetic module of the same genotype. These conflicting mutations can reduce the efficiency of the selection process.
3. Genetic interference is a phenomenon that is not restricted to multiple tasks which are of different difficulty and therefore evolve sequentially rather than simultaneously, with the easy tasks being solved first and the more difficult tasks later on in evolution. Genetic interference is a more general phenomenon which occurs when the genotype encodes multiple separate modules underlying tasks which can be either identical or different, with either the same or different difficulty. From this we conclude that this phenomenon is different from other effects of linkage on the evolution like Muller's ratchet or selective interference among various advantageous mutations.
4. Sexual reproduction can reduce the negative effects of genetic linkage by allowing the decoupling of portions of genotypes affected by favorable and unfavorable mutations and the recombining together of genetic segments in new genotypes. In this way sexual reproduction can find new genotypes that include only favorable mutations or only unfavorable mutations and this may increase the general efficiency of the evolutionary selection process. Notice that this can contribute to explaining the prevalence of sexual reproduction in higher organisms and the observed fact that sexually reproducing populations

tend to have higher mutation rates than asexually reproducing populations ([Maynard-Smith, 1978](#)).

A number of negative consequences of genetic linkage have been shown to exist. These include genetic deterioration due to the accumulation of mildly deleterious mutations, also called Muller's ratchet ([Felsenstein, 1974](#)), and the slowdown of adaptive evolution because adaptive mutations that appear simultaneously in a population compete with each other ([Crow and Kimura, 1965](#); [De Visser et al., 1999](#)). [Lenski and collaborators \(De Visser et al., 1999\)](#) called the latter phenomenon "adaptive speed limits". To our knowledge it has not been suggested that genetic linkage can completely prevent the adaptation of a character due to linkage with deleterious mutations affecting another character. We thus suggest that the form of genetic interference reported here may represent a new form of genetic constraint associated with genetic linkage. This effect is potentially more serious than either Muller's ratchet or "adaptive speed limits." Muller's ratchet is a slow process that only affects the long term stability of an asexual population. Adaptive speed limits are only relevant as long as the population is in adaptive non-equilibrium. Genetic interference as demonstrated here has immediate fitness consequences since it permanently prevents access to the fitness optimum. On the other hand, a sexually reproducing population would be able to approach the optimum to a higher degree and would thus experience a direct and permanent fitness advantage. One has to note, however, that this genetic interference effect needs both genetic linkage and high mutation rates.

One should also point out, however, that sexual reproduction may be unable to completely solve the problem of genetic interference. Behaviorally complex organisms such as human beings have to solve a large number of different tasks, not just two tasks as in the *What and Where* simulations. This implies that their genotypes will consist in a large number of genetic modules each encoding a different neural module. This may create so many possibilities of genetic interference with its negative consequences on evolution that sexual reproduction may be unable to block these negative consequences with its recombinatory power. A possible implication is that this may provide a possible explanation for the evolutionary advantage of learning.

Another implication of our simulations is that the results that have been obtained may contribute to explaining why the behavior of simple organisms tends to be genetically inherited whereas more complex organisms exhibit many behaviors that are learned during life. Simplicity/complexity in organisms can be defined in terms of number of different tasks. An organism is simple if its adaptive pattern requires the execution of a limited number of different tasks. An organism is complex if to survive and reproduce the organism must be able to execute many different tasks. While simple organisms tend to have nonmodular nervous systems, the need to avoid neural interference requires that the nervous system of a complex organism be modularized. The existence of separate modules for distinct tasks makes it possible to adjust the connection weights of each module without interfering with other modules and therefore with other tasks.

But why highly modular nervous systems tend to rely on learning during life as the mechanism for finding the appropriate weights for the different modules instead of entrusting evolution with the task of finding those weights? In other words, why are the connection weights of the highly modular nervous systems of complex organisms learned rather than genetically inherited, whereas in simpler organisms the connection weights of their nervous system can be genetically inherited? Elman et al. (1996) have suggested that the neural architecture of nervous systems is (largely) genetically inherited whereas the connection weights of the inherited architecture are (mostly) learned during life. The results of our simulations lend support to this proposal with respect to the highly modularized nervous system of complex organisms that must be able to execute many different tasks. In previous research (Di Ferdinando et al., 2001) we have shown that the best solution for neural networks that must be able to execute both the What and the Where tasks is to have evolution take care of finding the appropriate modular architecture for these two tasks and to have learning during life solve the problem of finding the appropriate connection weights for the inherited architecture. The results of the present simulations provide further arguments for Elman et al.'s proposal and explain why the solution found in Di Ferdinando et al.'s simulations for the What and Where tasks may be the most appropriate one.

If one were to entrust evolution with both the task of finding the appropriate network architecture and the task of finding the appropriate weights for the architecture, one would encounter the problem of genetic interference. Entrusting evolution with the task of finding the appropriate modular architecture and learning with the task of finding the appropriate weights for the inherited modular architecture solves the problem. Unlike evolution that relies on a single, global measure of an individual's performance, i.e. the individual's fitness, learning can rely on separate measures of the individual's distinct performances on different tasks. When the individual is executing some particular task, the individual is informed by its experience on how good is its performance on that particular task and this information (e.g. the teaching input in the back-propagation learning procedure) can be used by the individual's neural network to modify the connection weights of the particular module which is responsible for the current task, with no consequences for the connection weights of other modules.

The problems, hypotheses, and simulation results discussed in the paper can be relevant for understanding the evolvability of behaviorally complex organisms but also for designing artificial systems with practical uses using evolutionary principles. Practical systems such as autonomous robots may need to be "behaviorally complex", that is, they need to be able to process the input data in a number of different ways and to execute a variety of different tasks in response to the input. In such cases one has to confront all the problems discussed in this paper: modularity in the control mechanism, genetic interference, sexual or asexual reproduction, the mutation rate factor, the fitness formula factor, and the optimal assignment of acquisition processes to either evolution or learning.

## Acknowledgements

The financial support of the Italian National Research Council (Short Term Mobility Program 2000 to RC and Bilateral Project 1998–2000 to RC and GPW) and by the Yale Institute for Biospheric Studies to GPW is gratefully acknowledged. The authors thank the referees for valuable suggestions on the manuscript. RC would also like to acknowledge the useful discussions with Lee Altenberg, Paul Bloom,

Frank Keil, Stephen Kosslyn, Dan McShea, Brian Scholl, and the members of the GPW's lab at Yale University during weekly meetings and with the members of the Research Group on Artificial Life (GRAL) in Rome.

## References

- Calabretta, R., Parisi, D., in press. Evolutionary connectionism and mind/brain modularity. In: Callebaut, W., Rasskin-Gutman, D. (Eds.), *Modularity. Understanding the Development and Evolution of Complex Natural Systems*. The MIT Press, Cambridge, MA.
- Crow, J.F., Kimura, M., 1965. Evolution in sexual and asexual populations. *Am. Naturalist* 99, 439–450.
- De Visser, J.A.G.M., Zeyl, C.W., Gerrish, P.J., Blanchard, J.L., Lenski, R.E., 1999. Diminishing returns from mutation supply rate in asexual populations. *Science* 283 (5398), 404–406.
- Di Ferdinando, A., Calabretta, R., Parisi, D., 2001. Evolving modular architectures for neural networks. In: French, R. M., Sougné, J. P. (Eds.), *Connectionist Models of Learning, Development and evolution*, Springer-Verlag, London, pp. 253–262.
- Ellis, A.W., Lambon Ralph, M.A., 2000. Age of acquisition effects in adult lexical processing reflect loss of plasticity in maturing systems: insights from connectionist networks. *J. Exp. Psychol. Learning Memory Cogn.* 26, 1103–1123.
- Elman, J.L., Bates, E.A., Johnson, M.H., Karmiloff-Smith, A., Parisi, D., Plunkett, K., 1996. *Rethinking Innateness. A Connectionist Perspective on Development*. The MIT Press, Cambridge, MA.
- Felsenstein, J., 1974. The evolutionary advantage of recombination. *Genetics* 78, 737–756.
- Futuyma, D.J., 1998. *Evolutionary Biology*. Sinauer, Sunderland, MA.
- Haigh, J., 1978. The accumulation of deleterious genes in a population—Muller's Ratchet. *Theor. Pop. Biol.* 14, 251–267.
- Holland, J.H., 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, Cambridge, MA.
- Maynard-Smith, J., 1978. *The Evolution of Sex*. Cambridge University Press, Cambridge.
- Milner, D., Goodale, M., 1995. *The Visual Brain in Action*. Oxford University Press, Oxford.
- Rueckl, J.G., Cave, K.R., Kosslyn, S.M., 1989. Why are “what” and “where” processed by separate cortical visual systems? A computational investigation. *J. Cogn. Neurosci.* 1, 171–186.
- Rumelhart, D., McClelland, J., 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. The MIT Press, Cambridge, MA.
- Smith, M.A., Cottrell, G.W., Anderson, K.L., in press. The early word catches the weights. In: *Advances in Neural Information Processing Systems 12*. The MIT Press, Cambridge, MA.
- Ungerleider, L.G., Mishkin, M., 1982. Two cortical visual systems. In Ingle, D.J., Goodale, M.A. and Mansfield, R.J.W. (Eds.), *The Analysis of Visual Behavior*. The MIT Press, Cambridge, MA.
- Wagner, G.P., Mezey, J., Calabretta, R. Natural Selection and the origin of modules. In: Callebaut, W., Rasskin-Gutman, D. (Eds.), *Modularity. Understanding the Development and Evolution of Complex Natural Systems*. The MIT Press, Cambridge, MA, in press.
- Wagner, G.P., Gabriel, W., 1990. Quantitative variation in finite parthenogenetic populations: What stops Muller's ratchet in the absence of recombination? *Evolution* 44, 715–731.
- Waxman, D., Peck, J.R., 1999. Sex and adaptation in a changing environment. *Genetics* 153, 1041–1053.
- Yao, X., 1999. Evolving artificial neural networks. *Proc. IEEE* 87, 1423–1447.