
Neural Systems and Artificial Life Group
Institute of Psychology
National Research Council, Rome

What does it take to evolve behaviorally complex organisms?

Raffaele Calabretta¹, Andrea Di Ferdinando^{1,3}, Günter P. Wagner² and Domenico Parisi¹

¹ Institute of Psychology
National Research Council, Rome, Italy
{rcalabretta, andread, parisi}@ip.rm.cnr.it

² Department of Ecology and Evolutionary Biology,
Yale University, New Haven, CT, U.S.A.

³ University of Rome "La Sapienza"
Rome, Italy

Technical Report NSAL 01-02

April 11, 2001

Submitted

Department of Neural Systems and Artificial Life
Institute of Psychology, Italian National Research Council
V.le Marx, 15 00137 Rome - Italy
Phone: +39-06-86090233, Fax: +39-06-824737
E-mail: {rcalabretta, andread, parisi}@ip.rm.cnr.it gunter.wagner@yale.edu

<http://gral.ip.rm.cnr.it>

What does it take to evolve behaviorally complex organisms?

Raffaele Calabretta¹, Andrea Di Ferdinando^{1,3}, Günter P. Wagner² and Domenico Parisi¹

¹ Institute of Psychology
National Research Council, Rome, Italy
{rcalabretta, andread, parisi}@ip.rm.cnr.it

² Department of Ecology and Evolutionary Biology,
Yale University, New Haven, CT, U.S.A.
gunter.wagner@yale.edu

³ 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 must 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 also genetically inherited, this raises the problem of genetic linkage, with favorable mutations falling on one genetic module and unfavorable mutations on another genetic module. In our simulations this effect can cause a genetic constraint by preventing the genotype from reaching an adaptive optimum. This effect is different from other linkage effects described in the literature and we argue that it is a new class of genetic constraints. We show that sexual reproduction can alleviate the problem of genetic linkage by recombining together separate modules which have either favorable or unfavorable mutations. We speculate that this effect may contribute to the evolutionary maintenance of sexual reproduction. However, it is probable that the real solution to the problem of genetic linkage for behaviorally complex organisms is entrusting evolution with the task of finding appropriate modular architectures and learning with the task of finding the appropriate connection weights for these architectures.

Keywords: genetic linkage, neural interference, modular neural networks, genetic algorithms.

1. Introduction

What does it take to evolve behaviorally complex organisms? A behaviorally complex organism can be defined as an organism which is capable of a variety of different behavioral tasks and which responds to sensory input by processing the input in a variety of different ways. 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 & McClelland, 1986) to model the nervous system that controls their behavior and genetic algorithms (Holland, 1992) to model the evolutionary process that changes their genotypes in a succession of generations.

In Section 2 we propose that to avoid neural interference due to one and the same connection being dedicated to different tasks the accomplishment of different tasks requires modular neural networks, that is, neural networks which are divided up into separate modules each with connections 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 the 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 evolve the more difficult tasks later. 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 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 is that while evolution is guided by a global evaluation signal (fitness), learning during life can use distinct evaluation signals (teaching inputs) for each separate neural module.

2. Modularity in network architectures solves the problem of neural interference

Imagine an organism that must accomplish two different tasks in response to the same sensory input. For example, the organism is visually perceiving an object and it must recognize both where the object is located (Where task) and what type of object it is (What task). If the organism's nervous system is modeled by a non-modular neural network, neural interference can arise that will affect the organism's performance. A non-modular neural network for the What and Where tasks is a network in which some 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 these 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 as is appropriate 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 9 geometrically different objects is located in which of 9 different spatial positions in a retina (Figure 1) is better able to learn the task using the backpropagation procedure if the network architecture is modular rather than nonmodular. The visual input is contained in a 5x5 cell retina (Figure 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, Figure 1b), and 9 encoding the 9 different objects (What output units, Figure 1c).

What distinguishes the modular from the non-modular architecture is the pattern of connections projecting from the 18 hidden units to the 18 output units (higher connections). In the non-modular architecture all 18 hidden units project to each of the 18 output units. In the modular architecture 14 hidden units project only to the 9 What output units and the remaining 4 hidden units project only to the 9 Where output units (Figure 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 Rueckl *et al.*, 1989. We will come back to this below.) Therefore, in the non-modular 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, Rueckl *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 non-modular.

In Rueckl *et al.*'s work 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 the 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 procedure, exactly as in Rueckl *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 procedure 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 20x5=100 new individuals constitute the next generation. Each offspring inherits the same network architecture of its single parent (reproduction is nonsexual) but not its parent's connection weights. The inherited network architecture is subject to random mutations that can modify the network's pattern of higher connections, i.e., the connections between the hidden units and the output units (mutation rate = 5%; i.e., 5% of the 18 architectural genes, one for each hidden unit, were randomly selected and

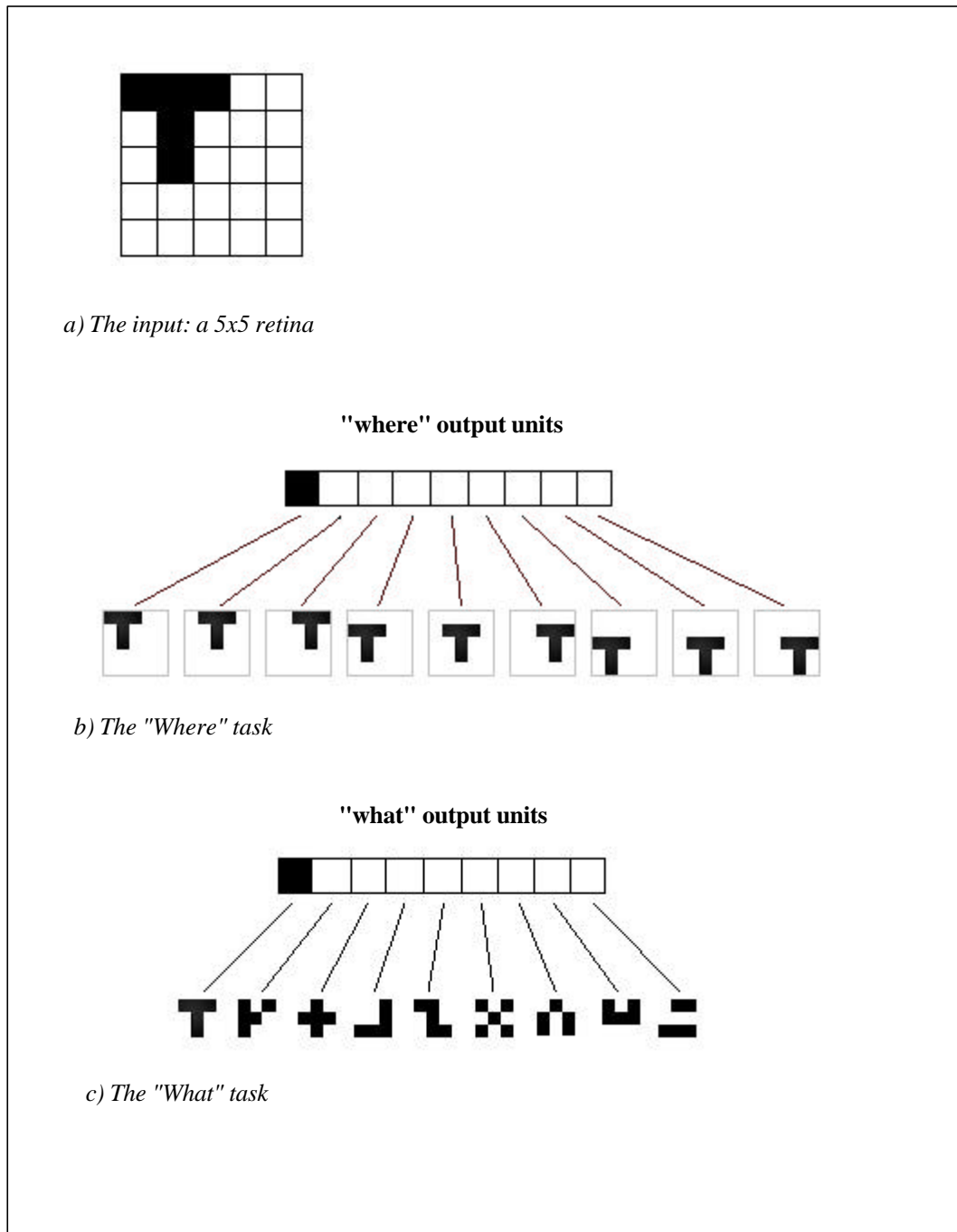


Figure 1. The What and Where tasks. Retina (a), 9 different locations (b), and 9 different objects that can appear in each location (c).

mutated). 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 network architecture that Rueckl *et al.* have

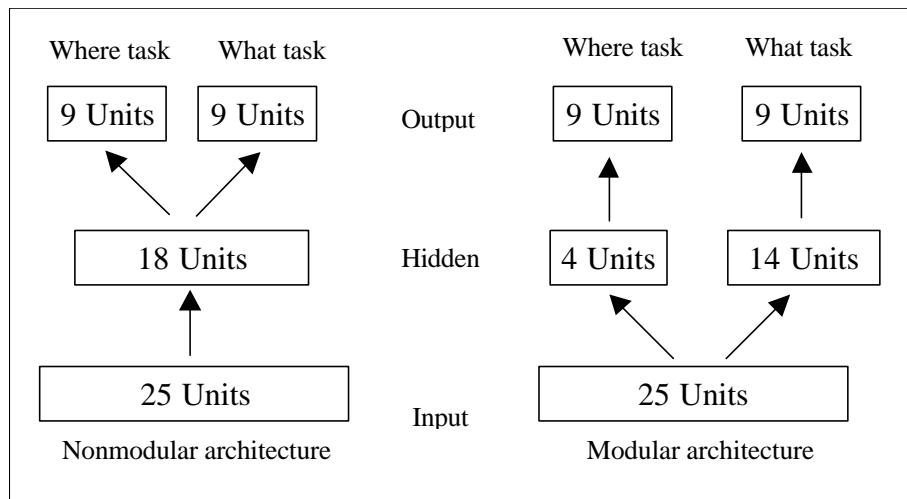


Figure 2. Modular and non-modular architectures for the What and Where tasks.

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, everything is genetically inherited, both the network architecture and the connection weights, and there is no learning during life. In these conditions (mutation rate = 5% for the network architecture; mutation rate = 10% for the connection weights) it turns out to be impossible to create neural networks that are able to solve both the What and Where tasks (cf. 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 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 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.

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 find also 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 can encode only the set of connection weights while the network architecture is fixed (it is the optimal network architecture of 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 of the simulations using the genetic algorithm to evolve the connection weights for fixed architectures, the organisms concerned have to solve only one task and therefore their network architecture tends to be nonmodular. 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 conducted a set of simulations in which the neural network architecture is fixed and is the optimal modular one for the What and Where task (14 hidden units for the What sub-task and 4 hidden units for the Where sub-task) and the genetic algorithm needs only to evolve the appropriate connection weights for this architecture. (Unless otherwise specified, all simulations reported in this chapter use a mutation rate of 10%, i.e., 10% of the connection weights are mutated by adding a quantity randomly chosen in the interval between -1 and +1 to the weight values, which are encoded as real numbers.) We have compared the results obtained with this architecture with those obtained with a fixed non-modular architecture.

As shown in Figure 3, for both the modular and the non-modular 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 connections in this architecture subserves both the Where and the What tasks, this 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 interference. Hence, neural interference cannot be invoked to explain the bad results obtained by the modular architecture.

We believe that these results can be explained by invoking 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).

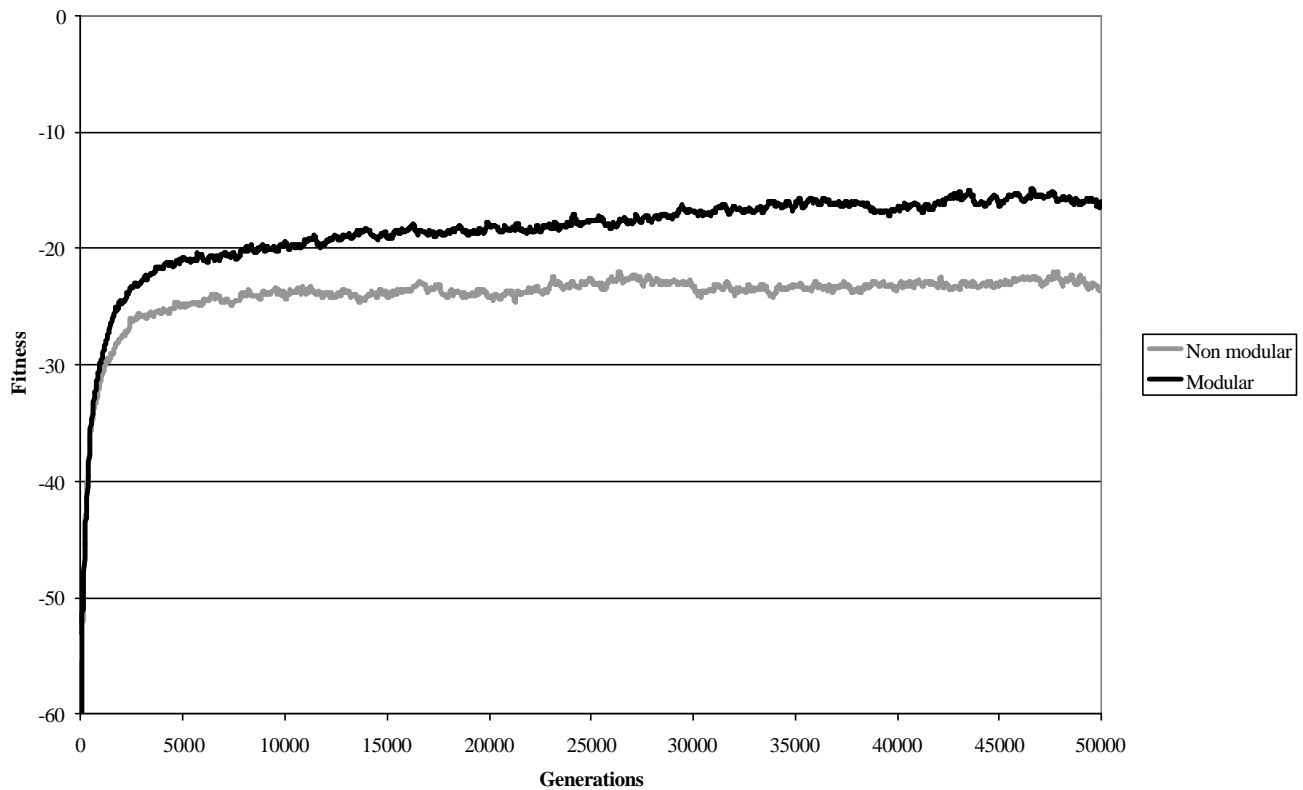


Figure 3. Average fitness for the What and Where tasks across 50,000 generations for modular and non-modular architectures when network architecture is fixed and what evolves is only the set of connection weights for the architecture. (All the figures in this paper show the average of 10 replications of each simulation.)

Why genetic linkage can have negative consequences for evolution? As we know, when the connection weights of a modular architecture are adjusted using the backpropagation procedure 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 procedure it is as if one were using two separate learning algorithms, one for the Where task and the other for the What task. In each cycle the backpropagation procedure 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 procedure and the connection weights are evolved rather than learned. But there is a critical difference between the two procedures. When one uses the backpropagation learning procedure each separate neural module is separately informed of the network's performance on each distinct task. 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 that 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 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 be improved, 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 can not "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 (Figure 4).

A direct test of our interpretation of our results in terms of genetic linkage consists in eliminating genetic linkage and by evaluating the performance of the two task 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 procedure 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, in this set of simulations the genetic algorithm is separately applied to two distinct populations. One 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 the other 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 can be 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. In both populations the final error on their respective task is near zero (results not shown).

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 should expect that increasing the mutation rate should lead to greater unfavorable consequences. A higher mutation rate should result in a higher

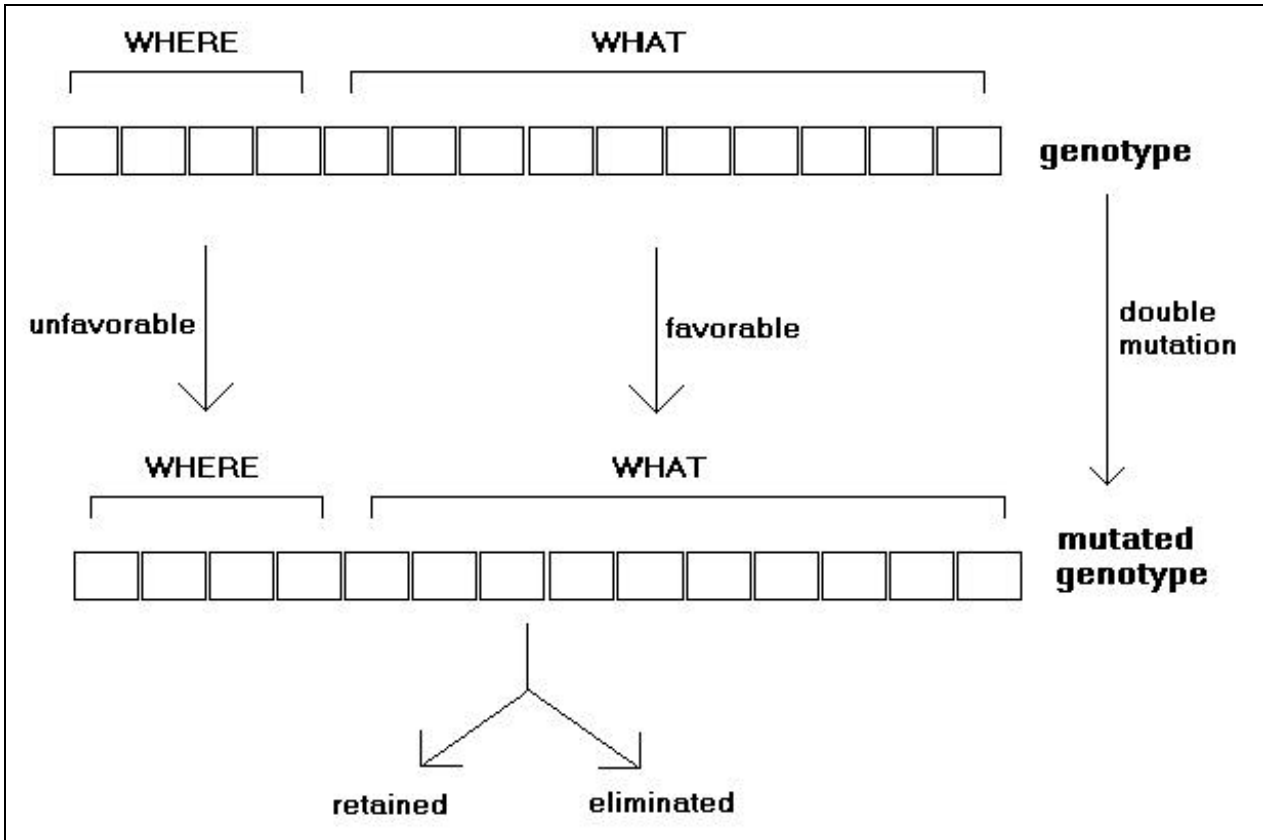


Figure 4. Genetic interference. Conflicting mutations (one favorable and the other one unfavorable) can fall on separate portions of the genotype encoding distinct neural modules, thereby hindering the efficiency of the selection process.

probability that a favorable mutation in one of the two genetic modules is accompanied by an unfavorable mutation in the other module. 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 an individual performs only one of the two tasks.

The results of these simulations are shown in Figure 5. 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 this value of 0.3% both tasks are solved in both conditions. Beyond this value, however, the equilibrium fitness decreases. What is interesting for us 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

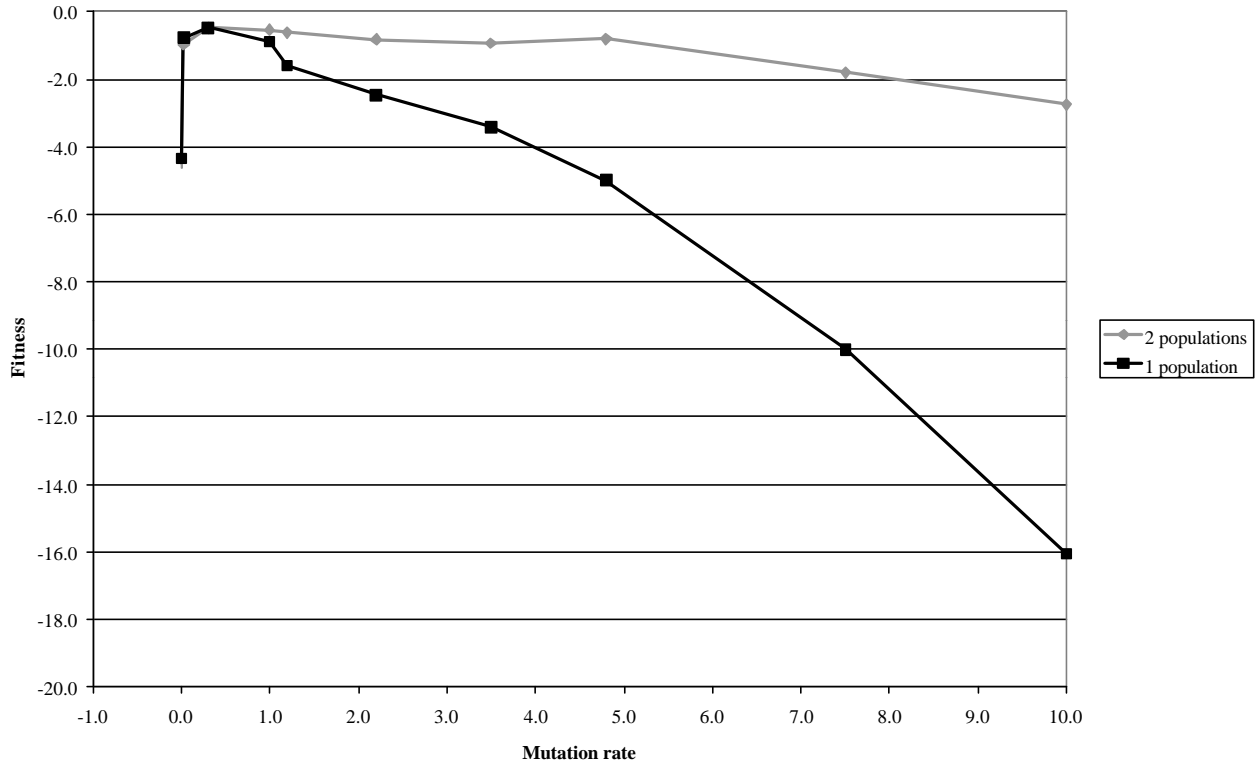


Figure 5. Average fitness at the end of evolution for populations with increasing mutations rates when the genotype encodes both the Where and What tasks (1 population) and when it encodes either one or the other of the two tasks (2 populations). (The fitness for the “2 populations” condition is the sum of the fitness of the What population and the fitness of the Where population.)

neural modules and behavioral 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.

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 (non-modular 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 reaches almost perfect performance. When it is time to acquire the more difficult What task, it turns out to be impossible to reach perfect performance in the later acquired What task. Many simulations using the backpropagation procedure have shown that neural networks exhibit what is 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 & 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. The Where task is acquired earlier than the What task in these evolving populations in the sense that the error on the Where task goes down more rapidly (in the early generations) than the error on the What task. Only when the error on the Where task has reached a sufficiently low level, the error on the What task begins to decrease significantly. However, this timing of the acquisition of the two tasks results in a lower terminal error in the Where task than in 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 reject 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 value 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:

Total Fitness = 5 x What Fitness + 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. Hence, evolution concentrates on the What task until high levels of performance in the What task are reached. But when, later on evolution, evolution turns to the Where task it is too late (results not shown). This "loss of plasticity" of the neural networks prevents evolution from reaching good levels of performance in the Where task. In other words, the situation is similar to that of our simulations so far, but reversed. In our simulations so far the two tasks were equivalent in terms of importance for reproduction but, since the Where task is 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 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.

This simulation shows that "age of acquisition" is a critical factor in evolution and not only in learning. A more balanced solution to the problem of solving 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:

Total Fitness = 3.5 x What Fitness + Where Fitness

show that we can obtain a parallel increase in fitness for both the What and Where tasks from the beginning of evolution. (The value of 3.5 reflects the ratio between 14 What hidden units and 4 Where hidden units.) However, even if the What and Where tasks are acquired simultaneously, genetic interference prevents the reaching of a satisfactory terminal error on both tasks (Figure 6).

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. During backpropagation learning the neural network is given the same Where teaching input twice for the two modules. 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

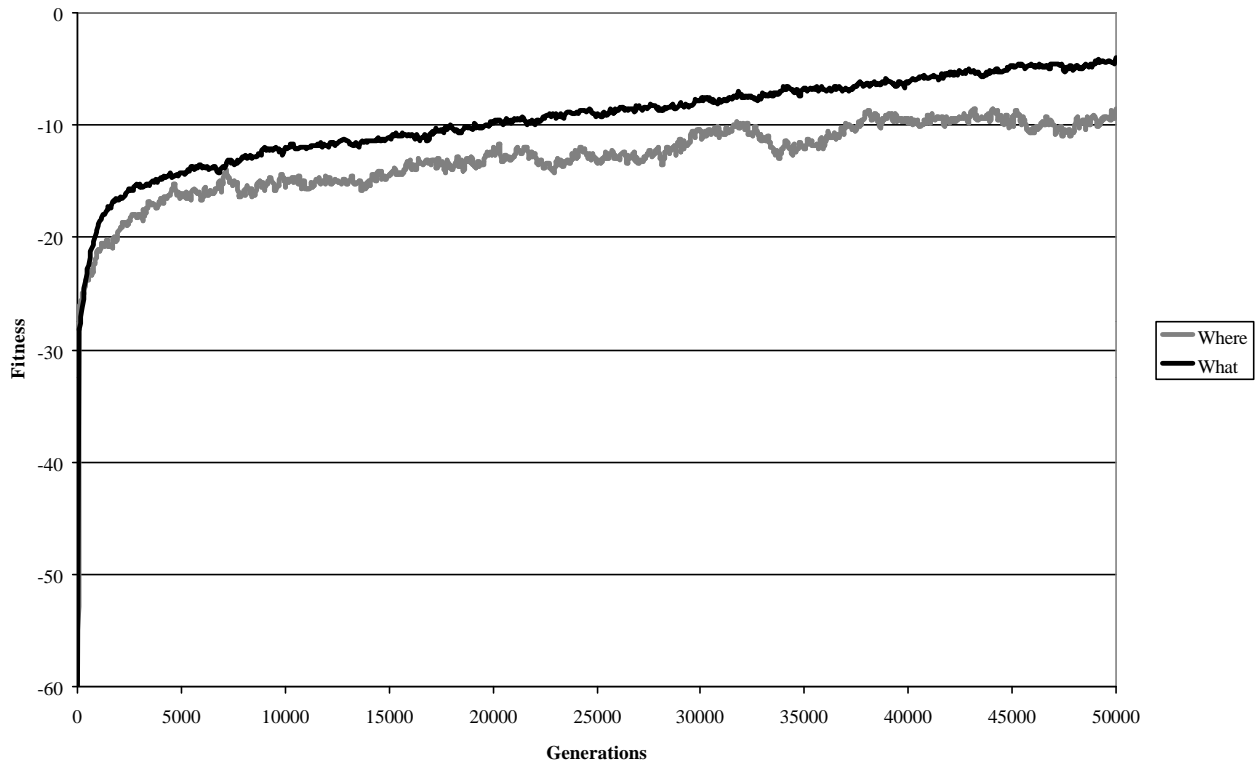


Figure 6. Average fitness across 50,000 generations separately for the What and Where tasks when both tasks evolve together because, by manipulating the fitness formula, the greater adaptive value of the What task compensates for the greater difficulty of the What task compared to the Where task.

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 (Figures 7 and 8).

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

In all simulations described so far reproduction is nonsexual: an offspring inherits the genotype of its single parent. If reproduction is nonsexual, 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

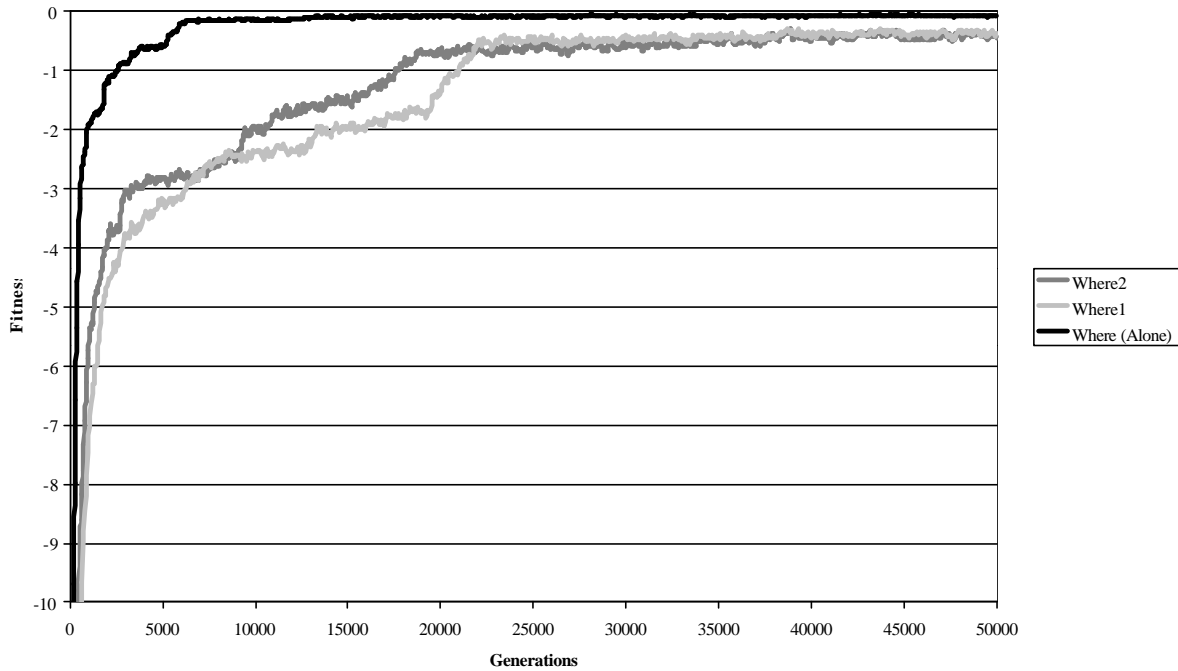


Figure 7. Average fitness across 50,000 generations in a population that has to solve two Where tasks compared with a population that has to solve a single Where task. (Notice that the fitness in this figure is scaled differently from the other figures to show the small differences among the curves.)

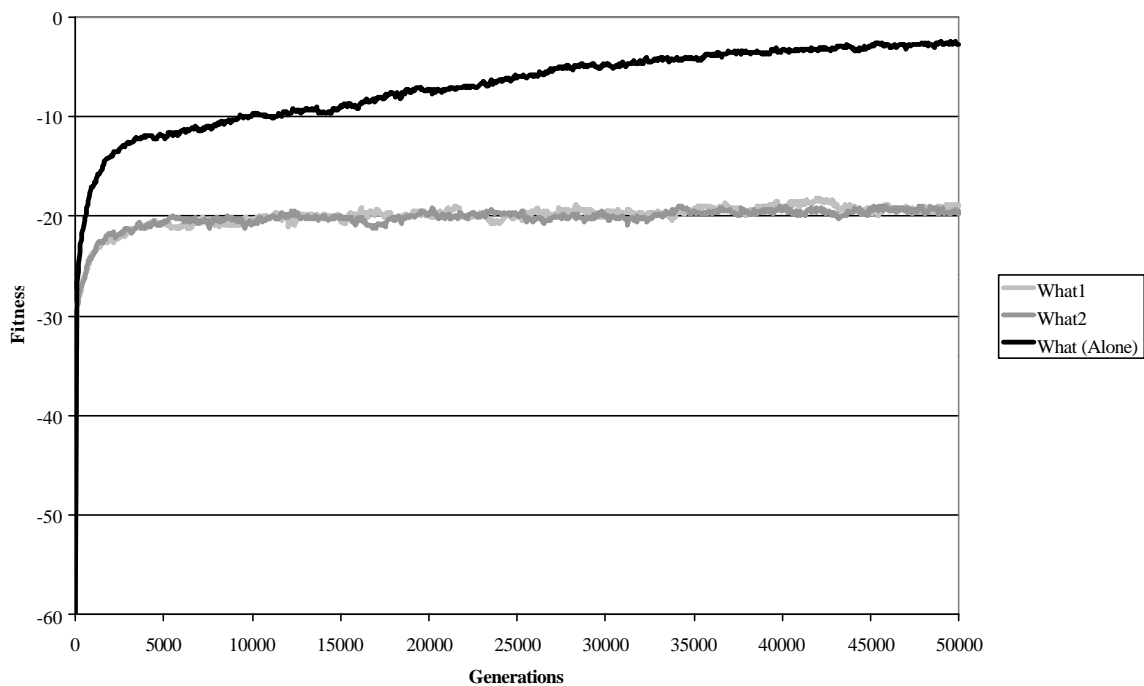


Figure 8. Average fitness across 50,000 generations in a population that has to solve two What tasks compared with a population that has to solve a single What task.

the negative consequences of genetic linkage. If this is true, this can contribute to explaining the evolutionary emergence of sexuality and sexual recombination since sexual recombination reduces the negative effects of linkage and make it possible the emergence of more complex organisms that are capable of multiple tasks.

To test this hypothesis we have changed the reproductive mechanism of our populations from nonsexual 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 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 mutation rate is 10% per connection weight.

The results of the simulations show that sexual reproduction limits the negative consequences of genetic linkage (Figure 9). (Consider that in some of the seeds both tasks are successfully solved.) If reproduction is nonsexual and conflicting mutations occur in the same genotype, favorable and unfavorable mutations are necessarily linked together and there is no way to retain the favorable mutations while throwing the unfavorable mutations away. 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).

5. Discussion

In the introduction of this paper we have asked the question "What it takes to evolve behaviorally complex organisms?", that is, organisms whose adaptive pattern requires the accomplishment of many different tasks. The simulations described in the paper allow us to propose some answers to

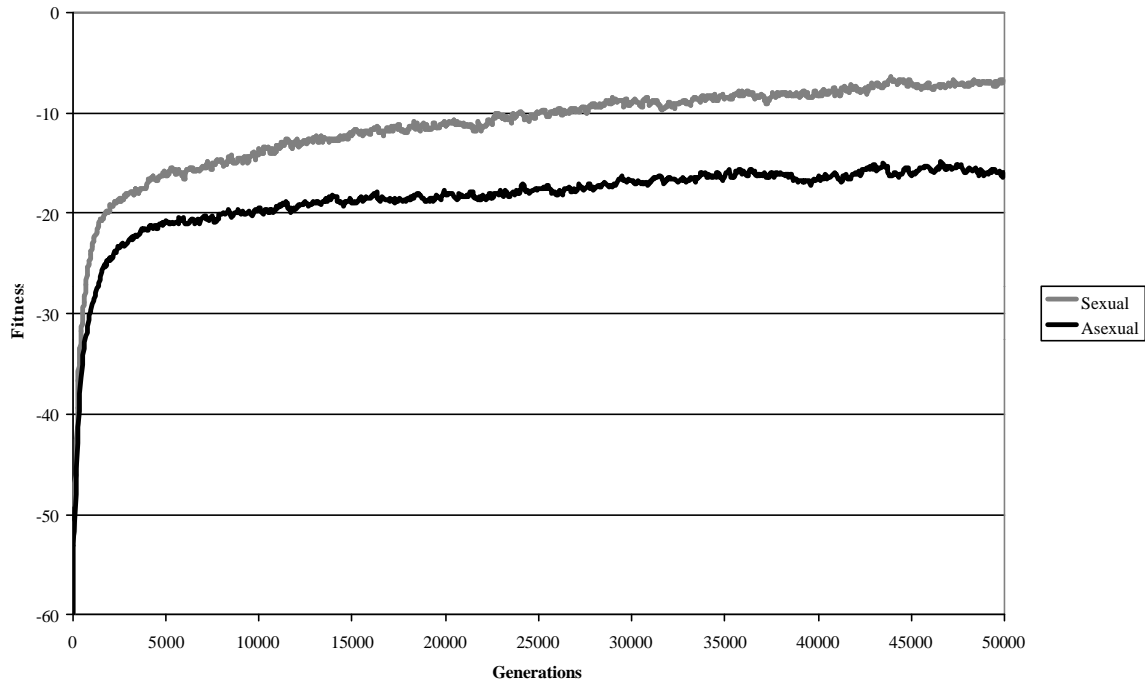


Figure 9. Sexual reproduction leads to a greater efficiency of the evolutionary process by limiting the negative consequences of genetic linkage.

this question. The evolvability of behaviorally complex organisms requires 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 & 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 very general phenomenon and 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.

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 process. Notice that this can contribute to explaining the evolutionary emergence of sexual reproduction in populations of initially nonsexually reproducing organisms and the observed fact that sexually reproducing populations tend to have higher mutation rates than nonsexually 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, 1976) or the slowdown of adaptive evolution because adaptive mutations that appear simultaneously in a population compete with each other (Crow and Kimura, 1965; *Arjan et al.*, 1999). Lenski and collaborators 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 is 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 which 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 larger 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 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.

In fact, we believe that the results of the simulations reported in this paper can 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 non-modular nervous systems, the necessity 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 various 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 (mostly) 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 is 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. A favorable mutation of the connection weights might be disadvantageous in another genetic architecture. The effects of mutations in connection weights and architecture can not be optimized simultaneously because they do not contribute additively to fitness. 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 backpropagation 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.

We have shown that sexual recombination can reduce genetic interference by combining together portions of the genomes of two distinct individuals (father and mother) which may both have been improved by favorable mutations. However, sexual recombination appears to be able to solve the problem of genetic linkage only for neural architectures that contain few separate modules. If the modules are more numerous, as they are in the nervous systems of really complex organisms, then sexual recombination may not be sufficient to make the strategy of entrusting evolution with the task of finding both the appropriate neural architecture and the appropriate weights a sensible strategy. A more sensible strategy is a "divide and conquer" strategy which relies on evolution for finding the appropriate modular architecture and on learning for finding the appropriate connection weights for that architecture. The reason is that while evolution is based on a global evaluation of individual organisms but not on a separate evaluation of their separate traits (in our case, behaviors exhibited in executing distinct tasks), learning relies on separate evaluations of separate traits (behaviors in distinct tasks) and therefore it can approximate the desired performance on one task without interfering with the performance on other tasks.

Two final comments. An important variable that may play a role in genetic interference is the type of genotype/phenotype mapping used in the simulations. In all our simulations the mapping is one-to-one: a particular segment of the genotype maps into a single phenotypical trait (connection weight) and each phenotypical trait is encoded in a single genotype segment. If the mapping is many-to-many (the same genotype segment enters into the encoding of many traits and each trait is the result of the interactions of many genotype segments), the problem of genetic interference might appear in a different light and it becomes more complex.

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 nonsexual reproduction, the mutation rate factor, the fitness formula factor, and the optimal assignment of acquisition processes to either evolution or learning.

References

- Arjan, J. J., Visser, 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, 404-406.
- 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. *American Naturalist* 99, 439-450.
- 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.
- 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. *Journal of Experimental Psychology: Learning, Memory & Cognition* 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.

- 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.
- Rueckl, J. G., Cave, K. R., Kosslyn, S. M., 1989. Why are “what” and “where” processed by separate cortical visual systems? A computational investigation. *Journal of Cognitive Neuroscience* 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.
- Wagner, G. P., Mezey, J., Calabretta, R., in press. 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.
- Yao, X., 1999. Evolving artificial neural networks. *Proceedings of the IEEE* 87, 1423-1447.