Learning and evolution in neural networks

Stefano Nolfi
Istituto di Psicologia
National Research Council (CNR)
Rome, Italy
stiva@irmkant.bitnet

Jeffrey L. Elman
Department of Cognitive Science
University of California, San Diego
La Jolla, California 92093-0515
elman@crl.ucsd.edu
619-534-1147

Domenico Parisi
Istituto di Psicologia
National Research Council (CNR)
Rome, Italy
domenico@irmkant.bitnet

(Correspondance should be addressed to the second author)

ABSTRACT

In this report we present the results of a series of simulations in which neural networks undergo change as a result of two forces: learning during the "lifetime" of a network, and evolutionary change over the course of several "generations" of networks. The results demonstrate how complex and apparently purposeful behavior can arise from random variation in networks. We believe that these results provide a good starting basis for modeling the more complex phenomena observed in biological systems. A more specific problem for which our results may be relevant is determining the role of behavior in evolution (Plotkin, 1988); that is, how learning at the individual level can have an influence on evolution at the population level within a strictly Darwinian — not Lamarckian — framework.

Keywords: learning, evolution, neural networks, artificial life

INTRODUCTION

Research in recent years has shown that artificial neural networks have many of the characteristics found in biological organisms, including humans (e.g., Hanson & Olson, 1990). There also remain a number of behaviors for which neural network models have yet to provide good accounts. At first glance, one might argue that the most striking deficiency is that the fundamentally statistical nature of network learning algorithms provides little insight into the seemingly purposeful nature of almost all biological systems. Networks are capable of learning very high-order statistical correlations that are present in a training environment; and very importantly, they provide a powerful and attractive mechanism for generalizing behavior to new environments. Yet in some sense they are essentially passive. This passivity and lack of purpose lie in sharp contrast with organisms for whom endogenous goals play an important role in determining behavior.

One can, of course, construct a network to simulate a set of goals. Miyata (1988) describes a problem in which constraint satisfaction can be used within a connectionist model to achieve certain prespecified goals. For certain research programs, this is a reasonable and useful strategy. But it begs the question: Where do these goals come from originally? Who does the original network construction?

When one considers this question in the context of biological organisms, the answer is not difficult to find. There is no design process. The biological construction is the result of evolutionary processes. Genetic variation and natural selection interact to produce organisms which display behaviors that are, on the whole, conducive to perpetuation of the species. The most basic goals — instincts — arise from essentially a random walk through what Dawkins (1987) has fancifully called 'biomorphland'. The infant that fails to suckle or the animal that fails to forage for food is quickly pruned from the family tree. By this account the advent of many behaviors is merely fortuitous, and reinforced by natural selection. This suggests that the appropriate mechanism for developing goals and purposeful behavior in neural networks should be an evolutionary process, rather than either *a priori* hard-wiring or explicitly training to produce goal-directed activity.

This is the approach we have taken in the work which we now describe. We provide a mechanism for networks to change from one generation to the next; and we structure the environment in such a way that there is pressure on the networks to solve a problem. We do not specify the form of the solution; that is to be worked out through genetic variation and natural selection.

THE PROBLEM

Let us begin by assuming that our ultimate goal is to create an organism (O) that is able to find and eat food in its environment. We imagine that O's environment is a two-dimensional grid-world. At any particular moment O occupies one of the

cells. A number of food units are randomly distributed in the environment with a food unit occupying a single cell. O has a "face". We shall imagine that it is has a rudimentary sensory system that allows it to receive as input from the environment the angle (relative to where it is currently facing) and the distance to the nearest food. We shall also equip O with a simple motor system that provides it with a repertory composed of four different actions. O may turn 90 degrees right, turn 90 degrees left, move to the next cell in the facing direction, or stay still. Finally, when O happens to step on a food cell, it eats the food contained in it (and the food disappears). O shall be dumb to the extent that it is not aware when it eats.

We are interested in the question of how it is that O might come to develop purposeful behavior. Once inserted into an environment such as that described above, how might O learn to actively seek out food in an efficient manner? (We will measure efficiency simply by measuring the number of food pellets eaten divided by the number of actions. Thus if an O eats, on average, every 10 actions, it is less efficient then another O that eats every 5 actions.)

— Insert Figure 1 about here —

Let us begin by simulating O's nervous system with a feedforward network consisting of three layers (Figure 1). The input layer contains 2 nodes which receive the sensory information from the environment and another 2 nodes that receive proprioceptive information indicating O's current movement. These 4 nodes are fully connected to an intermediate ("hidden") layer of 7 nodes; the 7 hidden units are connected to 2 output units which will indicate O's next movement. Sensory input is encoded by 2 units representing the angle and distance to the nearest food (values scaled from 0.0 to 1.0). Movement is encoded in a binary fashion, such that 2 units represent the four possible actions (00 = halt; 01 = right; 10 = left; 11 = forward; the actual analog values produced by the network will be thresholded to the nearest binary value) The network is initially assigned random weights on its connections.

Thus if O is placed in the sort of environment described above, a sequence of events will occur. Sensory input is received on two of the input nodes, and the previous action (initially chosen at random) is received by the other two input nodes. Activation flows up through the hidden units, and from the hidden units to the two output nodes. The (thresholded) values on the output layer are then used to move O in the indicated direction, and the next cycle begins.

Clearly, given these starting conditions, O's eating ability will be random and, assuming food is not plentiful in the environment, not very great. O will move haphazardly through its environment, eating food once in a while when it happens, by chance, to step on a food cell. In no reasonable sense could we say that O's behavior was intelligent or purposeful.

One way of having O develop a better eating ability would be to use a connectionist learning algorithm (such as backpropagation of error, Rumelhart, Hinton, & Williams, 1986) to teach the network to select the appropriate actions and sequences of actions when O finds itself in a specific position relative to a food unit. When O selects a particular action with the nearest food in a particular position (i.e. angle + angular distance) with respect to O, and this action is wrong, the teaching input signals the correct action so that the network can use the error to gradually change the connection weights and improve performance.

However, this solution is the one we rejected at the beginning of this paper. We do not want to explicitly teach the network how to eat, since it seems highly unlikely that nature provides most species with a "teacher" that gives individuals feedback at each step and leads them to correct performance. Furthermore, the optimal strategies may be unclear, especially if the environment is complex. It would be preferable to make it possible for the network itself to develop useful eating strategies, rather than seeing if it could learn a particular strategy chosen by us.

Another way of teaching a network to approach food which is perhaps more plausible has been used by Cecconi and Parisi (1989). O's are allowed to roam at will in the environment. Whenever an O happens to step on a food unit its spontaneous behavior ceases. The O is re-positioned in the cell where it was 8 time steps previously. The O repeats its actions, but this time it receives feedback which teaches it how to get to the food cell.

However, in the work that follows we took a different approach. We wanted to develop O's with purposeful behavior by using natural selection — in a sense, by "breeding" networks. The idea was to see whether a population of randomly varying O's that was allowed to reproduce, with a selection agency and random variation, would result in a final population whose behavior was non-random and, to all intents and appearances, purposeful.

Simulation 1

In our first simulation we began with 100 O's, each with the architecture described above (i.e., 4 inputs, 7 hidden units, 2 outputs). Each of the 100 networks was assigned a different random set of weights. We called this Generation 0 (G0).

G0 networks were allowed to "live" for 20 epochs, where an epoch consisted of 250 actions in 5 different environments (50 actions in each) for a total of 5000 actions. The environments consisted of a grid of 10 x 10 cells; 10 pieces of food were randomly distributed throughout the grid. The O's were placed in individual copies of these worlds (that is, they developed in isolation).

At each time step, the input layer was activated in the manner described earlier, with 2 units receiving sensory information regarding the location of food, and 2 units receiving proprioceptive information indicating the already planned movement.

The input units activated the hidden units, which in turn drove the motor response in the manner described above, with the result that each O moved at random through the world. (O's were allowed go outside the 10 x 10 cells; however, there was no food outside this space.) In this first simulation, there were no changes to the connection weights during the lifetime of an O.

Since the 100 networks had different sets of random connection weights, it happened out that some O's ate more and some O's ate less during their lifetime. At the end of the period assigned to G0 (that is after 5000 actions), O's were allowed to reproduce. However, not all O's reproduced. The 20 O's which had accumulated the most food in the course of their random movements were allowed to reproduce; each was permitted 5 offspring. These 100 (20 x 5) new O's constituted the next generation, G1.

Reproduction was accomplished by using the set of network weights from a parent as the template for the child (since these weights did not change during the lifetime of the parent, the weights were the same those that the parent had started its own life with). In addition, mutations were introduced into the copying process. A mutation was obtained by selecting 5 weights at random and perturbing the weight's value by some random value between +/-1.0.

After the G1 O's were created they were allowed to live for 5000 time periods, following the same regimen as the preceding generation. Behavior in G1 differed slightly from that in G0 as a result of two factors. First, the 100 O's in G1 were offspring of a subset of O's from G0. Second, the offspring themselves differed slightly from their parents because of the small random mutations in network weights.

These differences led to small differences in mean food eaten by the O's in G1. At the end of their lifetime it was possible again to permit the 20 most successful individuals in G1 each to reproduce 5 times, forming G2.

This process continued for 50 generations. If one looks at the eating ability (i.e., mean food eaten) of successive generations of O's, one finds a continuous increase from G0 to G49. This increase actually conflates two different measures. There is an increase in the mean eating ability of entire generations (100 individuals), and there is also an increase in the peak eating ability of each generation, i.e. the eating ability of the best individual of that generation. Figure 2 shows the change in the two measures from G0 to G49.

— Insert Figures 2 and 3 about here —

The changes in behaviors which are summarized in Figure 2 can are also apparent through visual inspection of the movements of individual O's in different generations. Figure 3 shows the trajectory of a randomly selected O from G0. The movement pattern is typical. O's wander through the grid without any apparent goals. Oc-

casionally a food cell is traversed, but on average the amount of food collected is low.

— Insert Figure 4 about here —

By G49 the pattern is quite different. Figure 4 shows the path of a randomly selected O from this final generation. Movement is almost always toward the nearest food, and the O frequently succeeds in consuming all the food in a given environment.

In this simulation we have seen that differential reproduction combined with random mutation can led to networks which "learn" (over many generations) to solve a problem. Subjectively, one has the impression that behavior during initial generations is random and undirected, whereas O's in later generations exhibit what appears to be deliberate and purposeful behavior. This behavior is not taught explicitly, although it is a natural consequence of the selection mechanism. This mechanism depends on both differential reproduction and also random mutation. Differential reproduction alone would result in an increase in mean eating ability of successive generations but no increase in peak ability. Mutation is important because it introduces new variability into the pool for networks, and makes it possible to explore different strategies. Of course, some of these strategies will be inferior to existing ones; it is then the differential reproduction mechanism that retains the improvements and eliminates the bad strategies.

In this simulation there was no role for individual learning; change could only occur through evolution. But individuals do undergo change during their lifetime as a result of learning. In the next simulation we explore the interaction of individual learning with evolution.

Simulation 2

In the first simulation, an evolutionary mechanism was used to develop networks which exhibited a certain behavior. That behavior was innate in the sense that it was displayed from birth and did not require any learning. But in real animals, except for the simplest and most rigid behaviors or animals, behavior is not determined by innate knowledge alone but results from the interaction of innate knowledge (in the current scenario, the weight matrix at birth) and learning (changes of weights through experience). Several researchers have suggest that there is an interaction between innate knowledge and learning (Waddington, 1942; Baldwin, 1896; Hinton, 1987; Belew, 1989). In many cases, what is learned may be only indirectly relevant to the pursuit of some endogenously determined behavior. What is learned may be not the behavior itself, but may support the behavior.

We are thus interested in whether there may be an useful interaction between the evolution of the eating behavior in O's, and other, ostensibly unrelated, information about the environment that is learned during life.

In the current situation, even if there is not an explicit teacher which instructs

the O's on the desirability of finding food, it is also true that the perceived environment does change as a consequence of the O's actions, and it is not implausible that the networks might learn about such changes.

More specifically, we can imagine that an O learns to predict what the sensory input will be at the next time step, given current sensory input from the world and a motor output on its part. The O does not learn that specific motor outputs are either good or bad, but simply that there are sensory consequences attendant upon specific motions in the context of specific environmental inputs. While this training involves supervised learning, the information required for the teacher is itself available in the environment; so one might think of this prediction task as an instance of self-supervised learning.

In order for the network to learn this task, it is necessary to modify the architecture slightly so that there are additional output units on which the sensory prediction can be made. As before, we have 4 input units: 2 input units which receive the current sensory information about the location of the nearest food (distance and angle) and another 2 which code the currently planned movement (identical to the motor output of the previous cycle). There are 7 hidden units, and now 4 output units. Two units represent the motor output of the network, and the 2 new units represent the O's prediction about what sensory input it will receive as a consequence of carrying out the currently planned movement. This second set of output units, but not the first, can receive teaching input from the environment (i.e., O's can see if their prediction is correct or not by comparing their expectations with the actual sensory input on the next time cycle). The network architecture is shown in Figure 5.

— Insert Figure 5 about here —

It is plausible to expect that the ability to predict how the input changes as a function of movement (an ability which is learned during an O's life) might be positively related to the ability to hunt for food (which is evolved over many generations). The interaction might go in both directions: Success at prediction would enhance the evolution of the hunting behavior, and at the same time those individuals who are successful hunters will also be successful at learning to predict changes in their environment.

To investigate this interaction, we ran a second experiment which was similar to the first, except that (a) we used the network architecture shown in Figure 5; and (b) on each time cycle, the networks were modified slightly in accord with teaching from the environment.

After each movement, the output on the prediction units (which are interpreted as the network's predictions of the angle and the distance of food on the next time cycle) were compared with the actual input from the world after the planned movement

was actually performed. The discrepancy constituted an error which was then used to change the weights of the network using the backpropagation of error learning algorithm (Rumelhart, Hinton, and Williams, 1986).

The network was not taught to make motor outputs. Because there was no error generated on the motor output units, only the weights connecting the hidden units to the output prediction units (i.e., those output units which did generate an error) were changed, along with the weights from the input units to the hidden units. Connections from the hidden units to the motor units were never changed during the course of a network's lifetime. (Connections subject to learning are shown in Figure 5 with dotted lines. Connections with solid lines were not modified through learning.)

All other parameters remained the same. Each O had a lifespan of 250 movements in 5 different worlds (50 movements in each world) for a total of 5000 movements. At the end of a generation, those O's which had gathered the most food were allowed to reproduce 5 offspring. As in the previous simulation, replicas were based on the parent's network at the time of the parent's own creation. This means that none of the changes which occurred in the parent's weight matrix as a consequence of learning during its lifetime were transmitted to offspring. What was transmitted was rather the potential for such changes.

The results of this simulation are shown in Figure 6 and 7, and are compared with the results of the first simulation. Figure 6 shows the peak performance and Figure 7 the average performance.

— Insert Figures 6 and 7 about here —

As one can see, the simulation with learning on the prediction task yields better performances on the eating task even though what is learned during the life of an O's is not inherited by its offspring.

If we now look at the eating performance during an O's life, we also find that, especially after a few generations, the O's eat more in the last part than in the first part of their life. Figure 8 shows the average food eaten as a function of epochs of life. Each curve shows the performance of a particular generation (the number on the right identifies the particular generation). The figure only shows the results every 5 generations.

— Insert Figure 8 about here —

From these results we can conclude that learning during life to predict how the input from the world changes with an O's movements helps in developing good eating strategies. But does the development of good eating strategies also help in the task of learning to predict? To obtain a direct answer to this question we could teach the O's of the first simulation (the one without learning) to predict and see if the O's of the

first generation learn to predict faster than the O's of the last generation. Unfortunately, we cannot do exactly this based on the behavior that the networks themselves generate because the O's of the first generation and those of the last generation do not have the same behavior (they execute different movements in the same context) and therefore the results of this test would not tell us much. More specifically, the O's of the last generations tend to have more complex behaviors. Completely random O's such as those in the first generation usually have limited action repertoires (e.g., executing a single movement in all contexts), whereas the O's in the last generations have developed more complex behaviors that allows them to eat (see Fig. 3 and 4). And the kind of behavior that O's have can influence how much they learn from being taught to predict.

To avoid this problem we trained O's from the first and the last generation by selecting a list of actions and making them all carry out the same actions instead of letting them move by themselves. (The actions were actually chosen randomly.) The results are shown in Figure 9.

— Insert Figure 9 about here —

As this figure shows, the prediction error curve of the last generation is better than that of the first generation. That means that O's that have developed good eating strategies learn to predict faster than those which haven't.

Another way of investigating the question if the possession of good eating strategies helps in learning to predict is to look at the behavior of the O's that are taught to predict during their life. Even if the results of prediction learning during life are not inherited, still networks could learn to predict faster if they have ancestors that have learned to predict. To see if this is true we must determine if the O's of the last generation in the simulation with learning learn to predict faster than the O's of the first generation (i.e., of random O's). However, even if we find that this the case, we can't know if this happens because the O's of the last generation have developed good eating strategies by evolution or because their ancestors have learned to predict during their life. In fact, we already know that developing eating strategies helps in predicting. But if we find that the improvement in learning to predict is greater than the improvement that we have observed in the O's of the last generation in the simulation without life learning, we might conclude that both the development of good eating strategies and the life learning of the preceding generations make the successive generations learn to predict faster. That is actually what happens, as is shown in Figure 10.

— Insert Figure 10 about here —

Parametric sensitivity

Running a simulations of the sort described above requires choosing settings

for a variety of parameters (number of mutations, population size, proportion between O's that reproduce and number of offspring, type of environment and so on). We carried out a number of simulations in order to understand the sensitivity of this model to different values of those parameters which we suspected would be more critical. Simulations were of the same basic form as those described above (and include learning to predict). Those parameter setting used in Simulation 2 will be referred to as 'the standard'.

In the first manipulation, we asked how the performance changes as a consequence of modifying the value of the population size. Populations of 64, 100 (the standard) and 144 O's were used in the three simulations. The proportion of the number of O's that reproduce and the number of offspring was kept constant (16 O's reproduce generating 4 offspring in the first case, 20 O's reproduce with 5 offspring in the second, and 24 reproduce with 6 offspring in the last case). Each offspring was generated by mutating 5 connection weights.

— Insert Figures 11 and 12 about here —

Figure 11 shows the peak performance (i.e. the performance of the best O's of each generation) and Figure 12 the average performance (i.e. the mean eating ability of entire generations). As can be seen, the intermediate population size yields the best results for both peak and average performance. Performance decreases slightly in the other two simulations.

We then asked how varying the number of offspring might affect peformance. In the first case, we had 6 offspring for each of the 16 best O's that reproduce; in the second, 5 offspring and 20 O's that reproduce (i.e. the standard); and in the third case, 4 offspring for each of the 25 best that reproduce. The population size was kept constant around 100 O's (96 O's in the first simulation and 100 in the other two). Each offspring was generated by mutating 5 connection weights. Figure 13 shows peak performance and Figure 14 average performance.

— Insert Figures 13 and 14 about here —

In this case, the simulation with the intermediate proportion between O's that reproduce and number of offspring yields somewhat better results for both peak and mean food eaten at early stages of development; the effect is minimal in the final generations.

Figures 15 and 16 show the results of simulations with different numbers of mutations (Figure 15 shows peak performance and Figure 16 shows average performance). The performance resulting from 2, 5, and 8 mutations (5 is the standard) is shown. Each simulation uses a population of 100 O's, 20 best O's that reproduce, and 5 offspring .

The simulation in which 5 connections of each O are mutated produce the best results eat the early stages of evolution, but again, by the final generations there appear to be no large difference in effects.

Discussion

We suggested at the outset that we were expecting a mutual positive influence of the prediction task (learned during an O's life) on the eating task (developed by evolution) because these two tasks appear to be inter-related. The requirements for one task might overlap with those of the other. The results appear to support this hypothesis; we may ask more specifically why this might be so.

Let us first consider how the two tasks appear to be related. In order to produce the correct output for each particular input (for a given task), a network must transform the input pattern into a hidden pattern more similar to the desired output than the input itself and then transform the hidden pattern into the desired output. The hidden representation of an input pattern is also called its internal representation, and the kind of internal representation that an input pattern has depends on the weights of the connections which go from the input to the hidden units. Good internal representations (and therefore good set of weights from the input to the hidden units) are task dependent, which means that they depend, in addition to the input pattern themselves, on the output patterns that are those desired for each input.

In the case of our O's, we have two different tasks: the prediction task and the eating task, but both tasks must be carried out by a single set of units at weights. What we need is to find good internal representations of the input which are appropriate both for the prediction task and for the eating task. If good representations for the eating task and good representations for the prediction task are similar we can expect that a network that is able to perform one of the two tasks will learn the second task faster because it starts from a point that is closer to the solution of that task. Therefore, when we say that the eating task and the prediction task are related what we mean is that they share good internal representations.

It is useful to imagine one particular example. When the food is exactly in front of an O, the sensory input represents the food as being at an angle of 0 degrees. If the food is slightly to the right of the network, it lies at an angle slightly greater than 0, while if the food is a little to the left it generates an input angle little less than 360 degrees. This means that the O receives very different inputs in situations which are very similar.

Consider the consequences of this for the prediction and movement tasks. Let us suppose the current action is a left turn (of 90 degrees). For purposes of prediction, the O must must generate very similar outputs (something like 95 degrees for the food that was slightly on the right and 85 degrees for the food that was slightly on the left). This means that a good internal representation for the prediction task for these

two input angles must be one that represents them in a similar way. Now let us consider what happens in the same case from the point of view of movement and eating strategies. When the food is in front of the O—whether or not it is slightly to the left or right—the most useful action will be go forward. Thus, it is also true for the eating task that two inputs which are very different must generate internal representations which are similar.

If we suppose that the prediction task and the eating task are positively related (i.e., they share good internal representations) it becomes easier to see why the development of good eating strategies by natural selection should make the prediction task easier to learn (as shown in Figure 9). The O's of the final generations have good internal representations for the eating task and, therefore begin their existence with internal representations that facilitate learning to predict. But how the prediction task that O's learn during their life help in developing good eating strategies (Figure 6 and 7) even if the prediction ability is not inherited?

A network with a given architecture can be represented as a point in a N-dimensional space where N is the number of the connections in the network. The weight on a connection determines the position the network occupies on the particular dimension representing that connection. In this way the points in such a space can represent all possible networks, i.e. the networks with all possible combinations of weights on their connections. If, for example, the initial population is made up of 100 networks with a random assignment of weights on their 56 connections, these networks will be represented by 100 points randomly scattered in a 56-dimensional weight space. Furthermore, the performance of a network on a particular task can be represented by an N+1th dimension, i.e. by an additional dimension that is added to the N dimensions of the weight space. There will be a performance surface with N+1 dimensions that goes through all the weight space and and has higher and lower points. Each point in the weight space (i.e. each possible network) will be located on this surface. If it is located on a high point of the surface this means that its performance on the task is good; if it is on a low point, its performance is bad.

Selective reproduction means that only the networks that are located in high points of the performance surface reproduce themselves, i.e. generate copies of themselves. This automatically implies that on the average the performance of the successive generation will be better than that of the previous generation.

However, progress that can be obtained in this way, i.e., with a simple selective reproduction mechanism, is limited since no offspring can ever be better than its parent. This limitation can be overcome by introducing a second evolutionary mechanism, random mutation. By modifying in a random way and within limits the connection weights of the copies (offspring), these copies will be distributed in the space around the point occupied by their parent. We can call this surrounding space the

"landscape" of the point. Some points in the landscape will be higher and some lower on the performance surface with respect to the given point. This means that some offspring will have a better and others a worse performance than their parent. But since selective reproduction ensures that the better offspring are more likely to reproduce than the worse offspring, the random mutation mechanism allows descendants to be better than their ancestors.

However, the two evolutionary mechanisms that we have described, selective reproduction and random mutation, do not exhaust the range of possible evolutionary mechanisms. There is a further possibility that becomes clear if we consider with some attention the nature of the landscape that surrounds a given point.

This landscape can be either good or bad. It is good if it contains many points that are higher than the given point, and it is bad if it contains many points that are lower than that point. This has an effect when one introduces random mutations. If the landscape of a given point (parent) is good, its mutated offspring are more likely to be better than the parent itself, and this will have a positive effect on the evolutionary process. On the other hand, if a point has a bad landscape, its mutated offspring are more likely to be worse than their parent and they will not contribute positively to the evolutionary process because they won't reproduce.

The problem is that neither the selective reproduction mechanism nor the random mutation mechanism are by themselves capable of utilizing this fact. These two mechanisms "see" the height of a point on the performance surface and can utilize this information, but they are not sensitive to the landscape which surrounds that point. This can have negative consequences for the evolutionary process.

For example, two points which are at the same height have the same probability of reproducing themselves independently of their respective surrounding landscapes. But in fact, it would be more useful if the point with a better landscape had a greater likelihood of reproduction, since its offspring would be more likely to be better than their parent. Or, even worse, of two points located at different heights the higher point would reproduce even if the lower point had a a much better landscape.

We believe that the role of learning in evolution is that of a third evolutionary mechanism, a mechanism that allows the evolutionary process to see the landscape surrounding each candidate for reproduction and to utilize this information. As we know, learning causes a progressive modification of the weights of individual networks and this means that during life the point which represents a given network moves progressively through weight space and on the performance surface. At the time of reproduction, all the positions that the network has moved through on the performance surface determine if the network will reproduce or not. In this way, reproduction depends on the effects that learning has had on the individual network, even if what is transmitted to offspring is, in a Darwinian and not Lamarckian fashion, the

position in the weight space that the point occupied at birth, before learning; or, more exactly, positions near the original position, because of mutations.

Thus, the fundamental effect of learning on evolution is to modify the ranking of the individual networks on task performance, i.e. to influence what is the set of individuals that will reproduce. The direction in which learning modifies the set of reproducing individuals is to include in this set a greater number of individuals that have a good landscape. This should improve the outcome of the evolutionary mechanism because it makes the search in the weight space more effective, and so it can explain (at least partially) the better performance that we obtain in Simulation 2 (i.e., with learning) as compared with Simulation 1 (i.e., without learning). This point has also been made by Hinton & Nowlan (1987) and Belew (1989). However, we think that this is not the whole story.

The task which is learned during life has a performance surface similar to the surface of the task on the basis of which selection operates (this is a notion of relatedness between two tasks which is connected to, but not identical with, our previous notion of sharing the same good internal representations). We will call the first task the *learning task* and the second the *evolutionary task*. In our simulations, the learning task is the task of predicting the changes in the relative position of a food element as a consequence of one's own actions. The evolutionary task is the task of approaching and eating food elements.

During life, learning modifies a network's weights and therefore causes a displacement of the point representing the network in weight space, and more precisely a displacement towards positions that are higher on the performance surface of the learning task; the network learns to predict. But since learning modifies the network's weights it will cause a displacement of the point on the performance surface of the evolutionary task as well. We know that the learning task and the evolutionary task are positively related, which means that their performance surfaces have similar shapes. A set of weights (a point in the weight space) which produces good performance on the learning task will tend to produce good performance on the evolutionary task also. This means that if the learning task and the evolutionary task are related, learning not only allows the evolutionary process to see the landscape of each candidate, it also allows the evolutionary process to explore that part of the landscape which corresponds to the best performance for both tasks.

At this point there is only a last logical step to be taken. How much an individual learns during its life depends on its landscape in the weight space. If the point's landscape is a good one, i.e. it contains many points that are higher than the initial point, the effect of learning will be a final position that is higher than in the case of a bad landscape. As a consequence, at the end of life those individuals that have a good landscape will tend to have a higher position on the learning task surface. But since

we are dealing with the case of positively related tasks, if an individual has a good landscape in the learning task it will tend to have a good landscape in the evolutionary task as well. Therefore, at the end of life those O's that learn more and faster will probably tend to reproduce. This implies that the evolution process will select for O's that learn a task (prediction in our case) faster and better even if O's are strictly selected based on their eating performance. This could explain why we found that the O's of Simulation 2 (i.e., O's that learned to predict during their life) were able to learn to predict faster that O's of Simulation 1 even though they did not inherit this ability (Figure 9). In addition, this could explain why the O's of Simulation 2 started improving their eating performance in the last part of their life only after some generations, which is when O's that learn to predict better have been selected (Figure 8).

Conclusion

We began at the outset by raising the question, Can purposeful behavior evolve in neural networks without an explicit teacher that trains the networks to the specific behavior? We raised the question because purposeful activity seems to be a hallmark of biological systems. We rejected the solution of explicit training (or pre-wiring) because, with a few exceptions, this solution is generally not used in the biological world.

The simple answer to the question, based on the results we have presented seems to be that yes, goal-directed behavior can indeed emerge from an evolutionary process which exploits the interaction of random variation with a selection mechanism. The selection mechanism supplies the force that makes the behavior adaptive, but—and this is important—it does not supply or teach the form of the behavior itself. In our case, the selection mechanism took the form of differential reproduction in favor of networks with greater food resources. The selection is neither cognizant of nor does it operate on the behaviors by which networks accumulate food. These behaviors are invisible to the selection mechanism. However, our mechanism provides an environment in which certain behaviors (in our case, motion toward food) resulted in the flourishing of individuals who display that behavior; individuals who did not were culled from the population. Insofar as these results are precisely what the lessons of evolutionary biology lead us to expect, they should not be at all surprising.

There are some important ways in which our simulations deviate from biological fact. The most obvious is that we introduce variation directly at the level of the individual (i.e., network weights), whereas genetic change normally occurs at the level of the genetic code which serves as a specification language for the individual. We believe that this simplifying assumption does not rob our results of their validity. We have also chosen to rely on mutation and asexual reproduction rather than sexual reproduction and chromosomal cross-over, even though the latter is a much more potent force for change; and this appears to be true for artificial genetic systems as well

as natural (Holland, 1975). Our problem was simply that since our variation occurs at the level of the individual, and since the connectivity pattern of neural networks forms a distributed pattern, it is meaningless to 'mix and match' parts of different networks. (Two neural networks may be functionally identical but mirror images of one another. In this case it is obvious that the results of crossover could be disastrous, and no offspring might be viable). Until we have a better idea of what an appropriate 'genetic code' for a neural network would be, we cannot take advantage of cross-over. Our current inclination, incidentally, is that this genetic code itself ought to co-evolve with the individual.

Our results also highlight another point which is sometimes obscured by the ego-centrism of introspection. For most of us, our own goals, purposes, and motives have a peculiarly special status. For some, an essential attribute of intelligent behavior is that it be directed toward some end, no matter how frivolous or apparently unproductive. Introspection tends not to be very revealing about the source of these goals but rather to cloud them with a quasi-mystical character. We believe that the networks in these simulations may be truly said to have acquired goals, and that their behavior is intentional. Furthermore we believe that the status of these intentions is fundamentally no different than the intentions we perceive in ourselves. Clearly, our own intentions are considerably more complex, subtle, indirect, and often less well understood than those of these simple networks. But we believe that the approach described here may be helpful in exploring ways in which such complex behavors may arise through originally random accidents which happen to have beneficial consequences for reproduction.

Finally, we show that learning during the course of an individual's lifetime can interact in a useful an non-obvious manner with the evolution in a species of purposeful behavior. Such interaction appears to be extremely complex and powerful, at least in the case in which the task learned during life is positively related to the task useful from the evolutionary point of view. Interestingly, learning appears to function as an additional mechanism that guides the evolutionary process. Evolution, on the other hand, seems to be able to push learning in the directions from which it can obtain the best advantage.

REFERENCES

Baldwin, J.M. (1896). A new factor in evolution. *American Naturalist*, 30, 441-451.

Belew, R.K. (1989). Evolution, learning and culture: Computational metaphors for adaptive algorithms. CSE Technical Report CS89-156. University of California, San Diego.

Cecconi, F. & Parisi, D. (1989). Neural networks that learn to predict where food is and also to eat it. Istituto di Psicologia, National Research Councils (CNR), Rome, Italy.

Dawkins, R. (1987). *The blind watchmaker*. New York: Norton.

Hanson, S., & Olson, C.R. (1990). *Connectionist Modeling and Brain Function.* Cambridge, MA: MIT Press.

Hinton, G.E. & Nowlan S.J. (1987). How learning guide evolution. *Complex Systems*, 1, 495-502.

Holland, J.J. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press.

Miyata, Y. (1988). The learning and planning of actions. ICS Report 8802. Institute for Cognitive Science, University of California, San Diego.

Rumelhart, D.E., Hinton G.E., and Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (Eds.), *Parallel Distributed Processing. Vol.1: Foundations.* Cambridge, NA.: MIT Press.

Rumelhart, D.E., McClelland, J.L. (1986). Parallel Distributed Processing. Cambridge, MA.: MIT Press.

Waddington, C.H. (1942). Canalization of development and the inheritance of acquired characters. *Nature*, 150, 563-565.

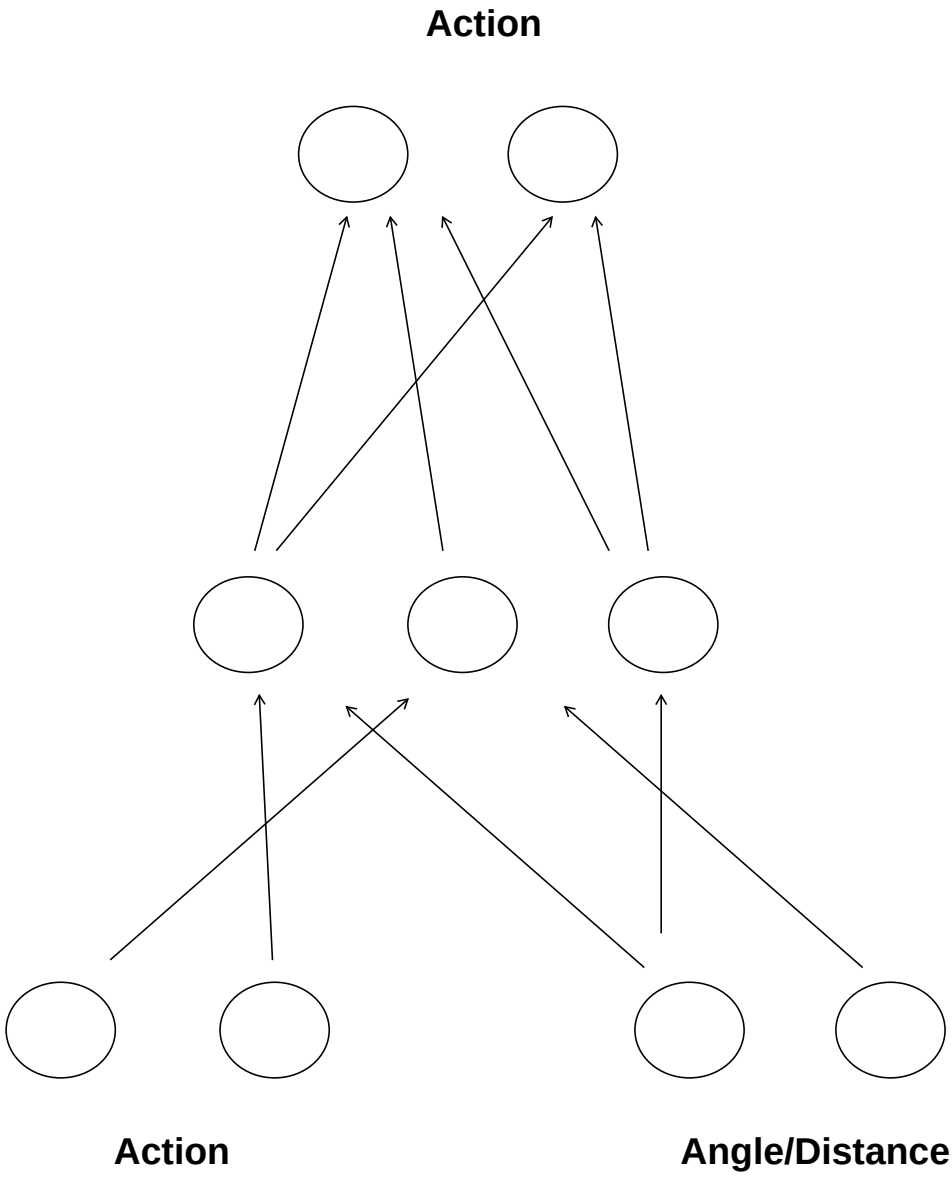

Figure 1

Architecture used in simulation 1

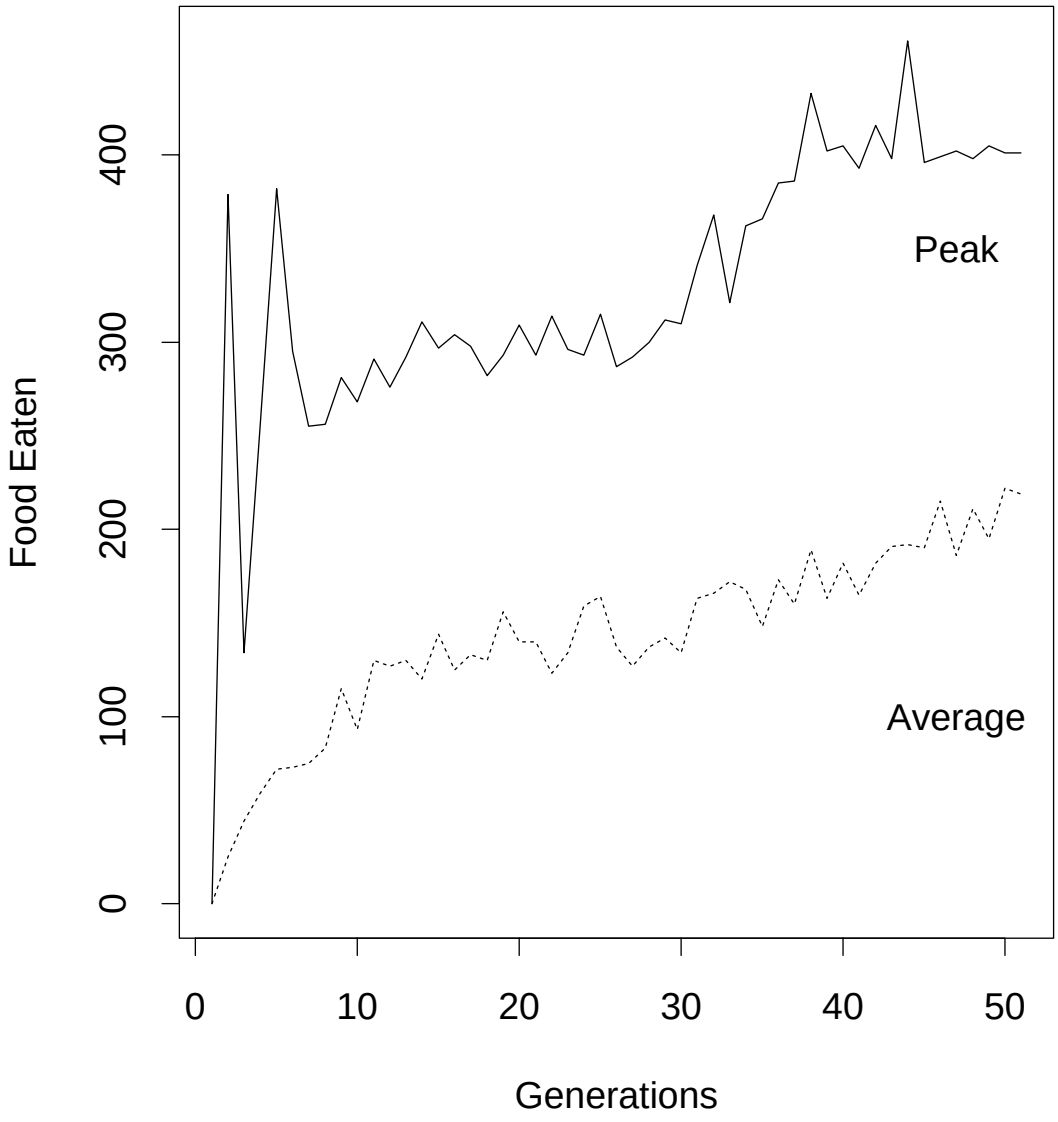

Figure 2

Average and peak food eaten (ordinate) as a function of generation (abscissa)

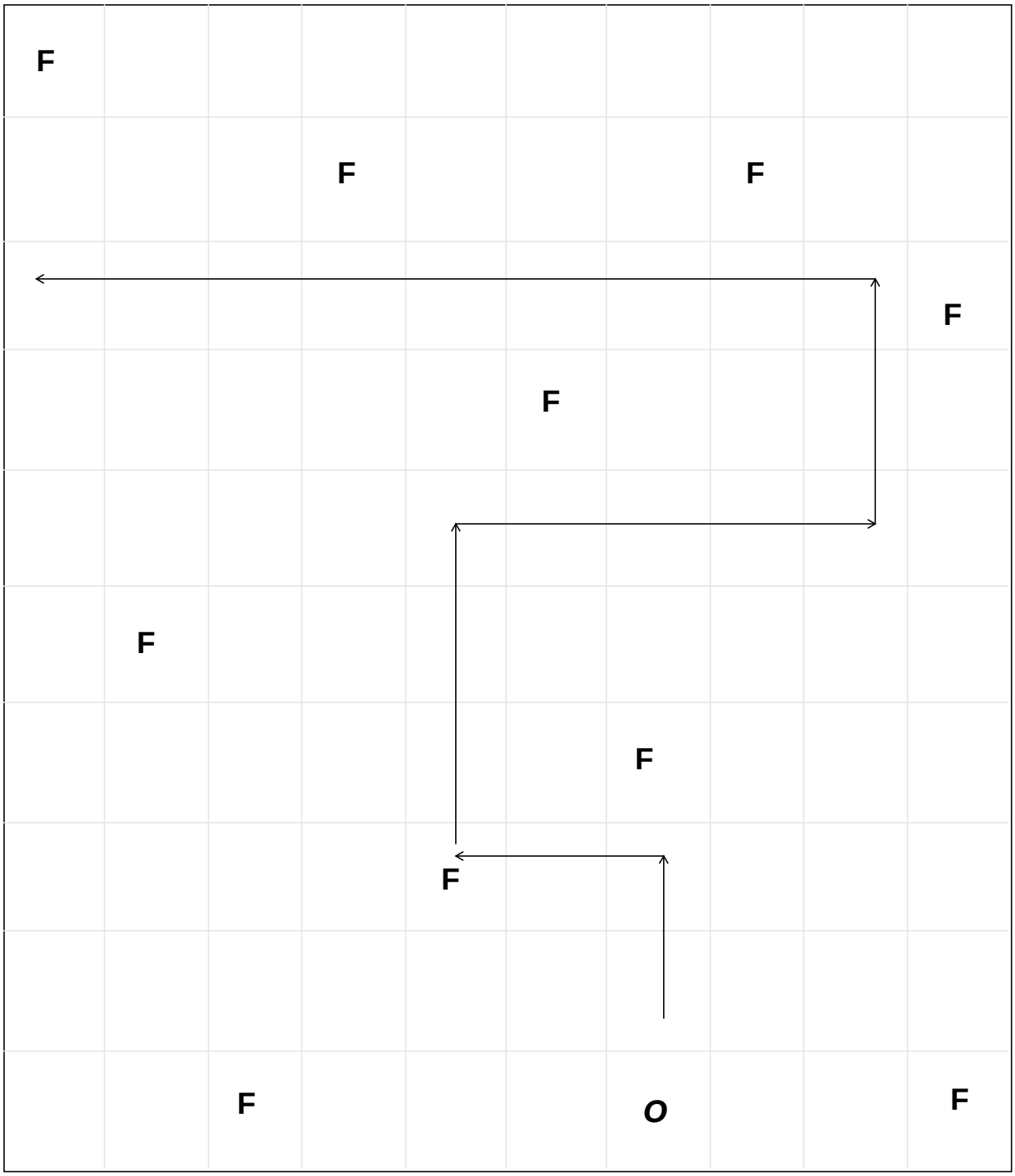

Figure 3

Path taken by randomly selected ***O*** from G0.

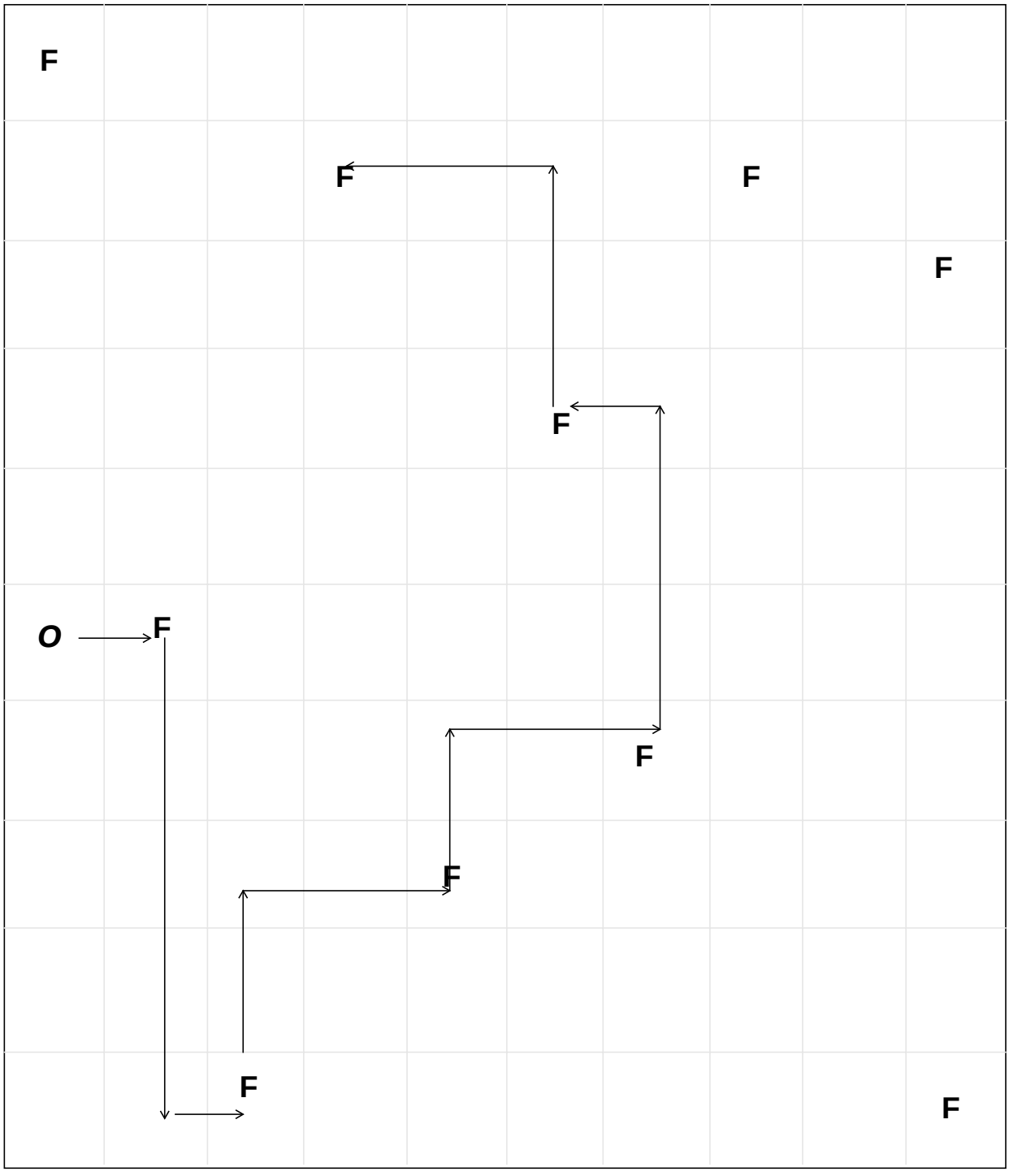

Figure 4

Path taken by randomly selected ***O*** from G49

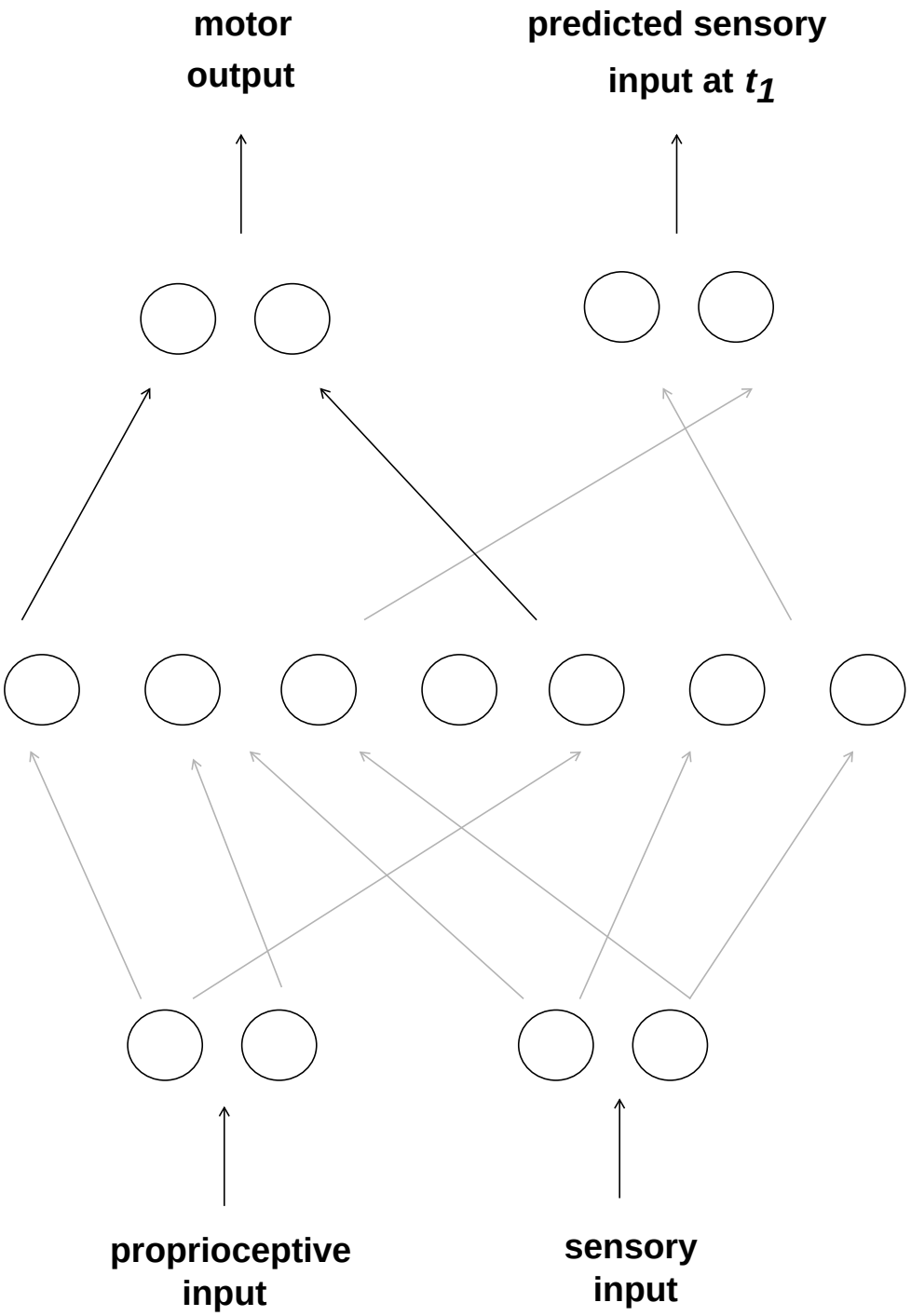

Figure 5

Architecture used in simulation 2

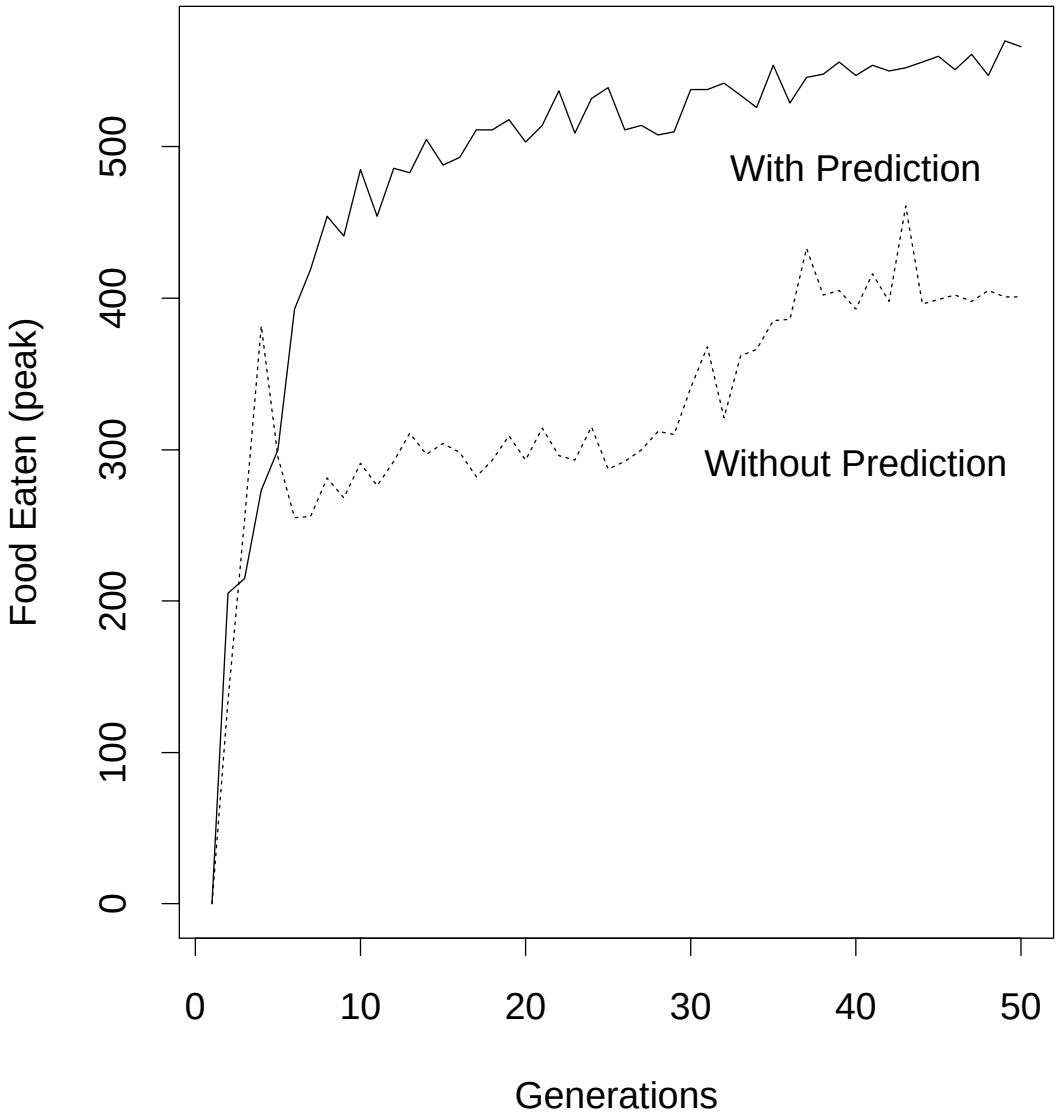
Food Eaten (peak)
0
100
200
300
400
500
0
10
20
30
40
50
Generations
With Prediction
Without Prediction

Figure 6

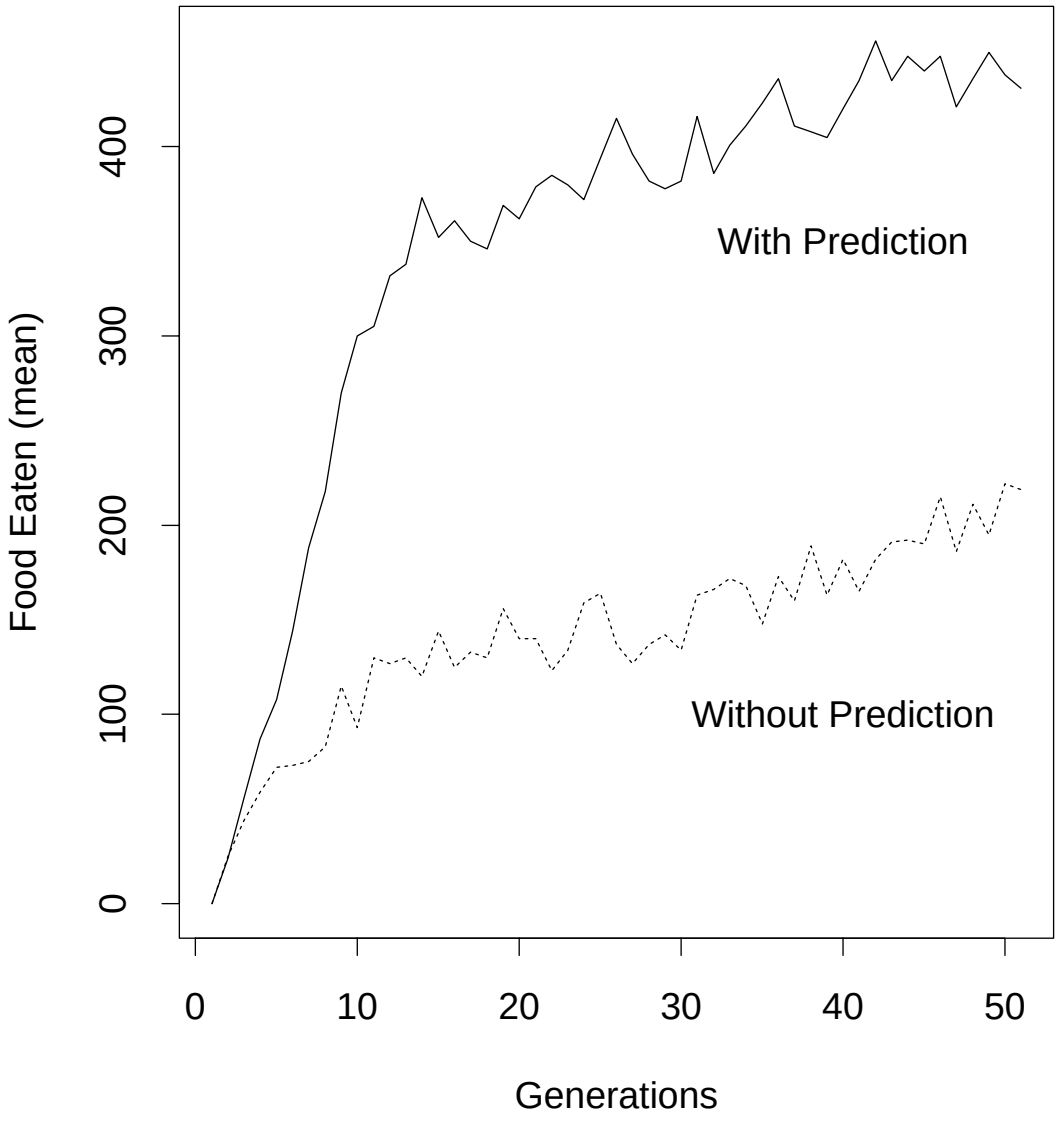
Food Eaten (mean)
0
100
200
300
400
With Prediction
Without Prediction
0
10
20
30
40
50
Generations

Figure 7

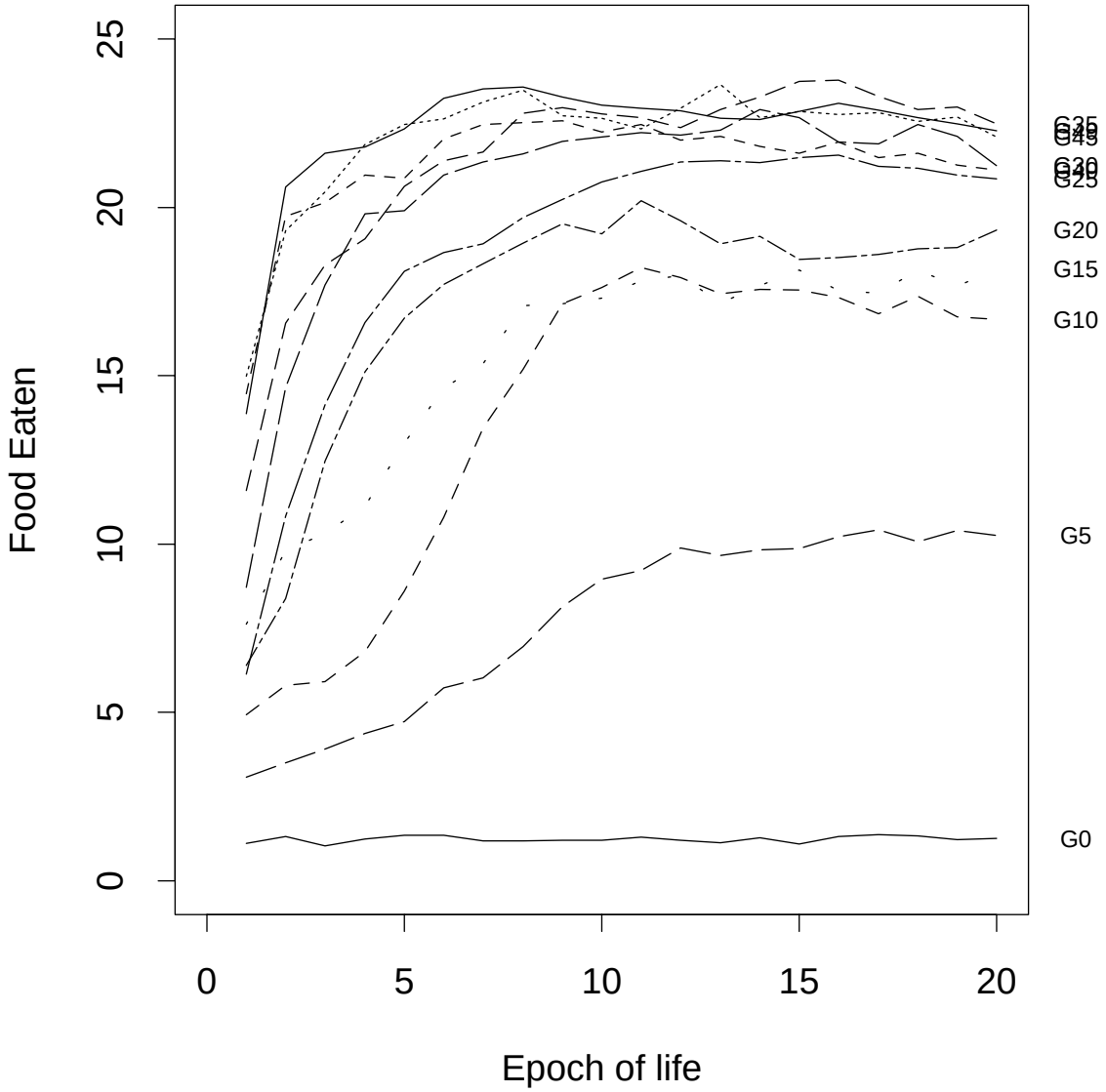
Food Eaten
0
5
10
15
20
25
0
5
10
15
20
Epoch of life
G35
G45
G30
G40
G25
G20
G15
G10
G5
G0

Figure 8

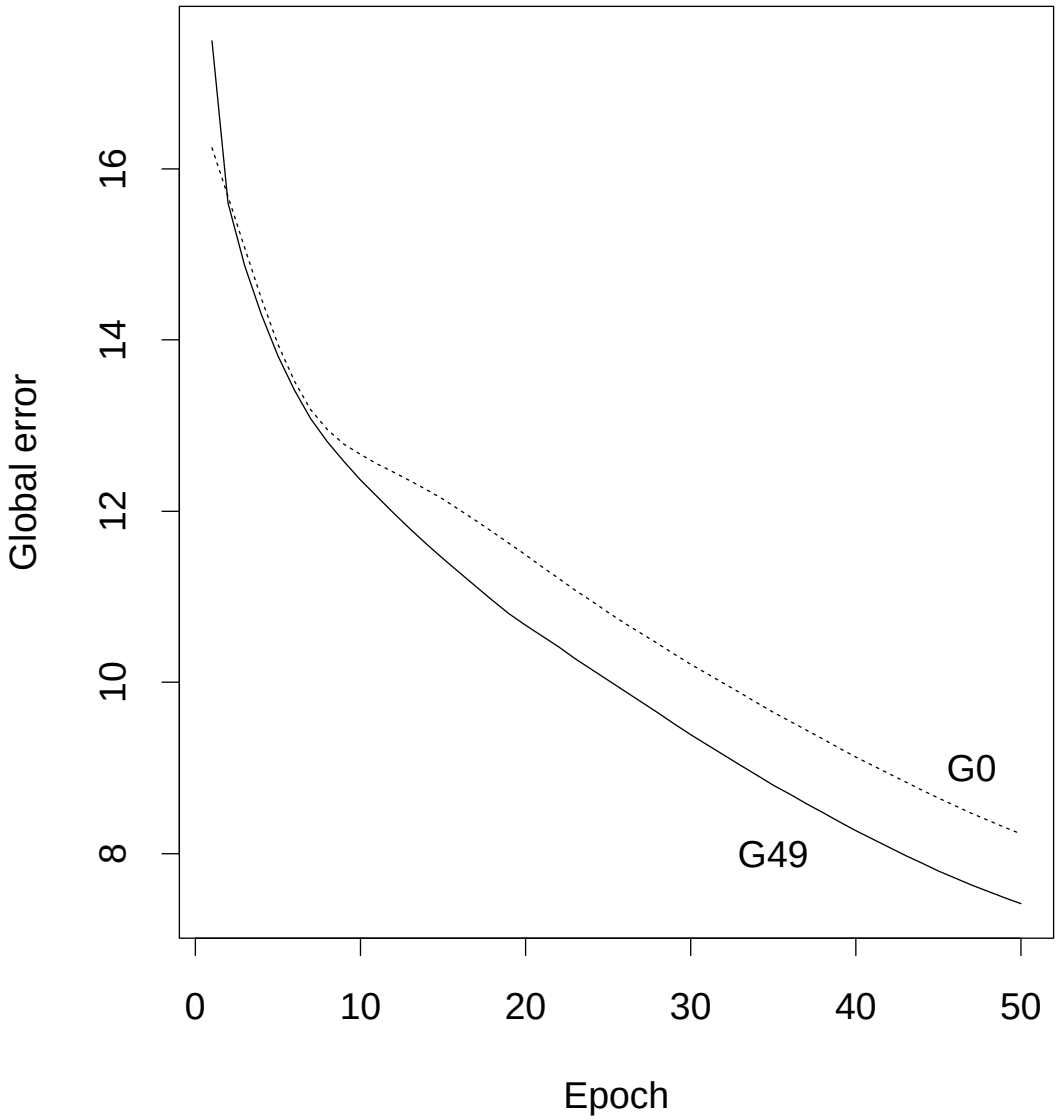

Figure 9

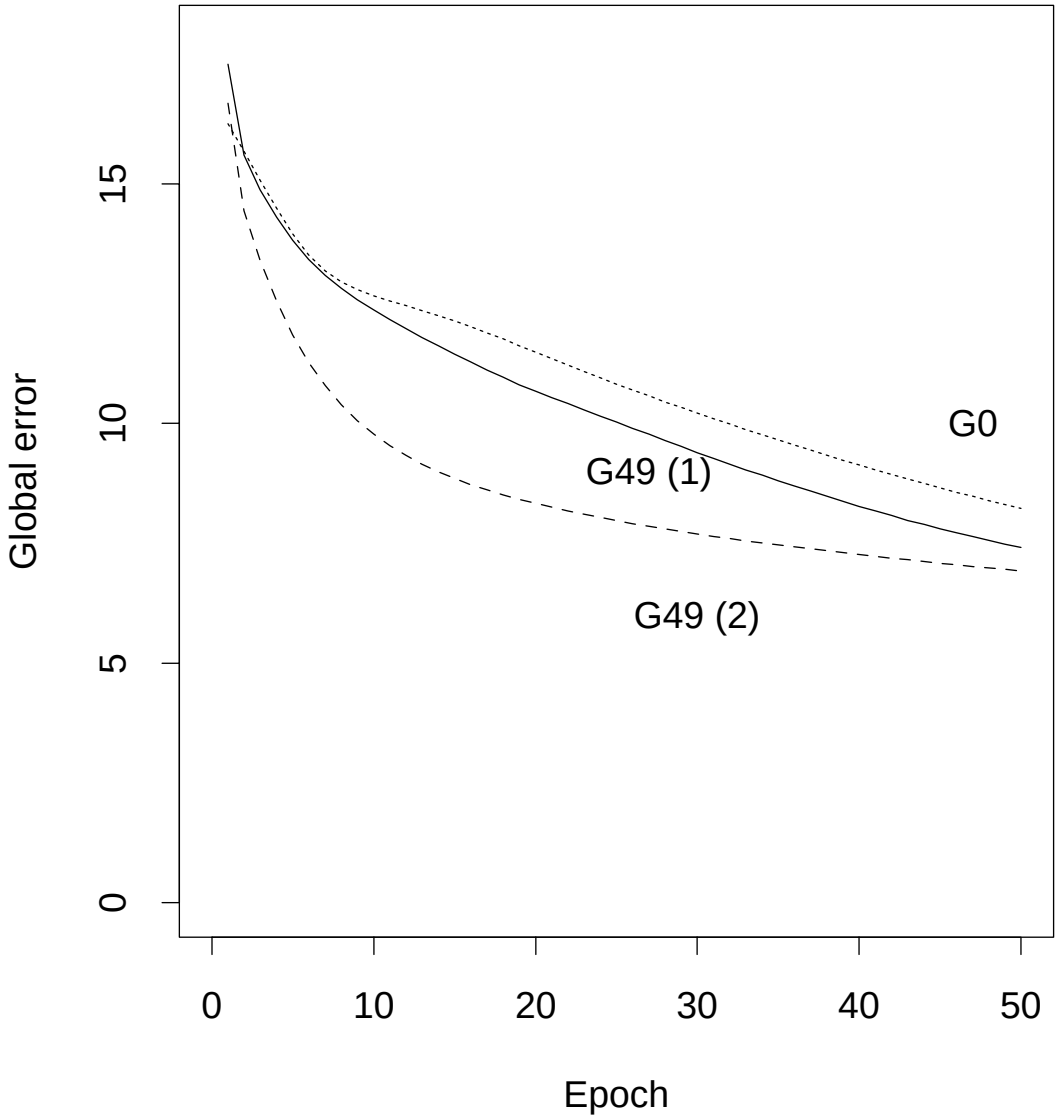

Figure 10

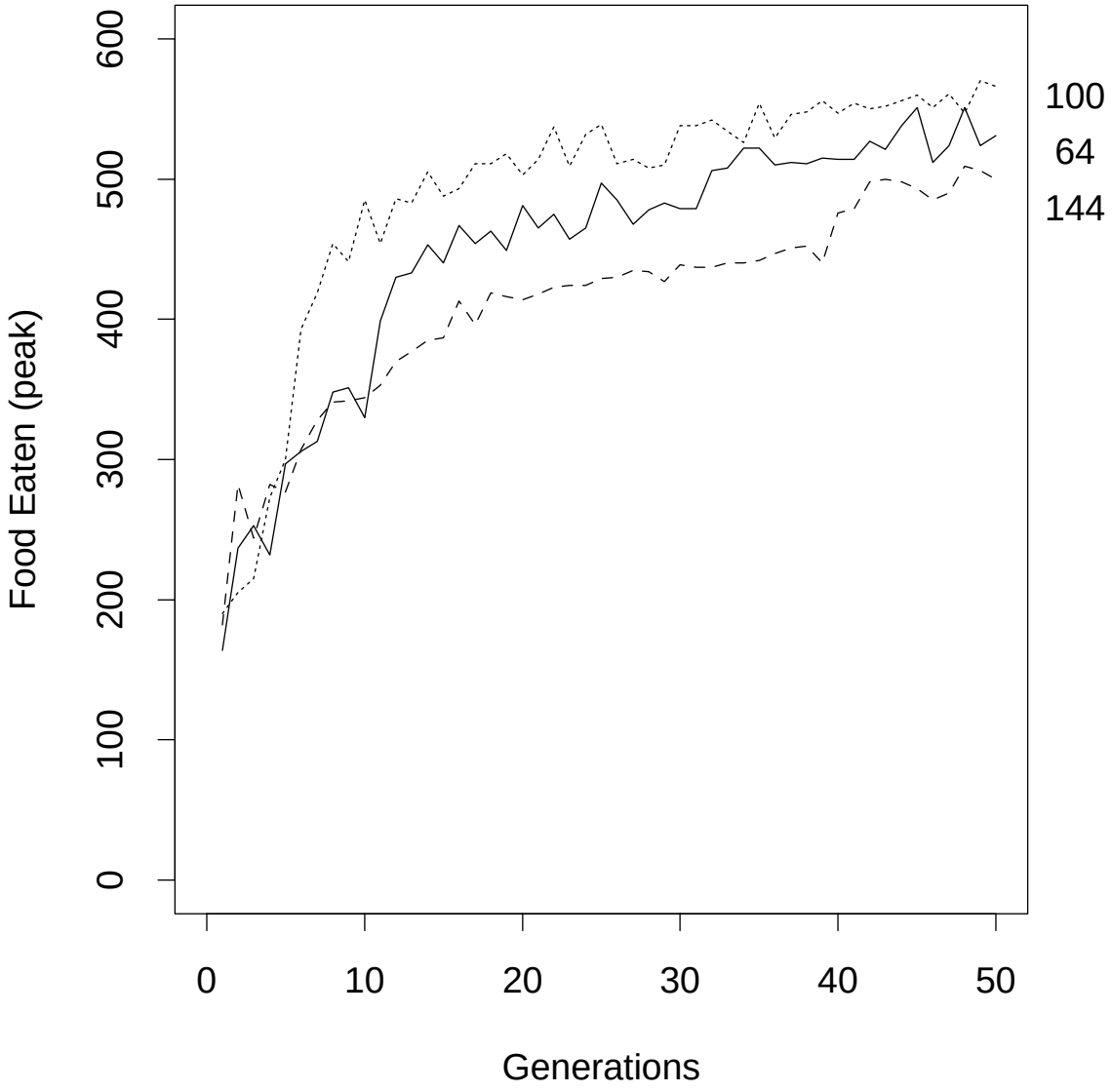
Food Eaten (peak)
0
100
200
300
400
500
600
0
10
20
30
40
50
Generations
100
64
144

Figure 11

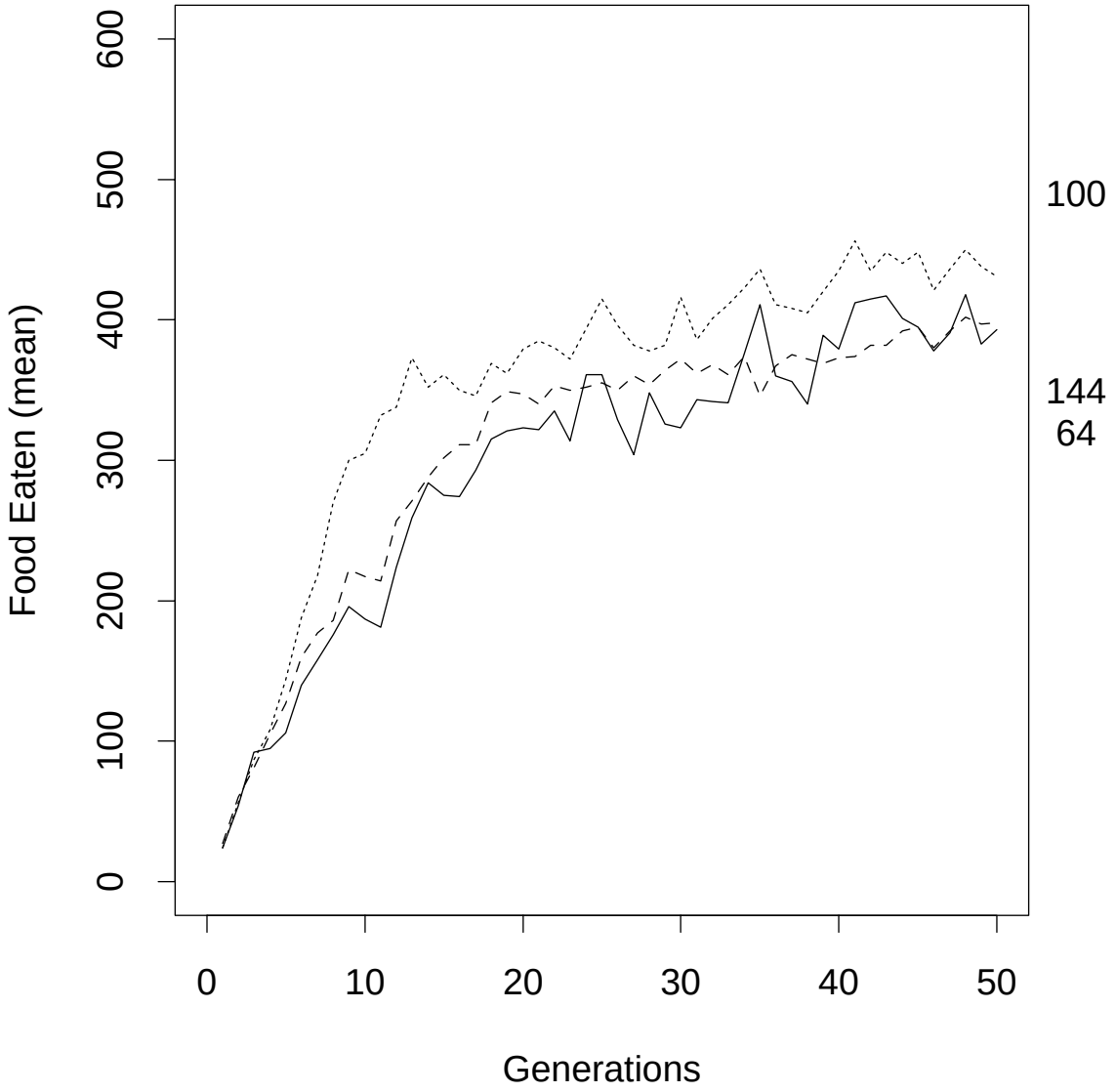
Food Eaten (mean)
0
100
200
300
400
500
600
0
10
20
30
40
50
Generations
100
144
64

Figure 12

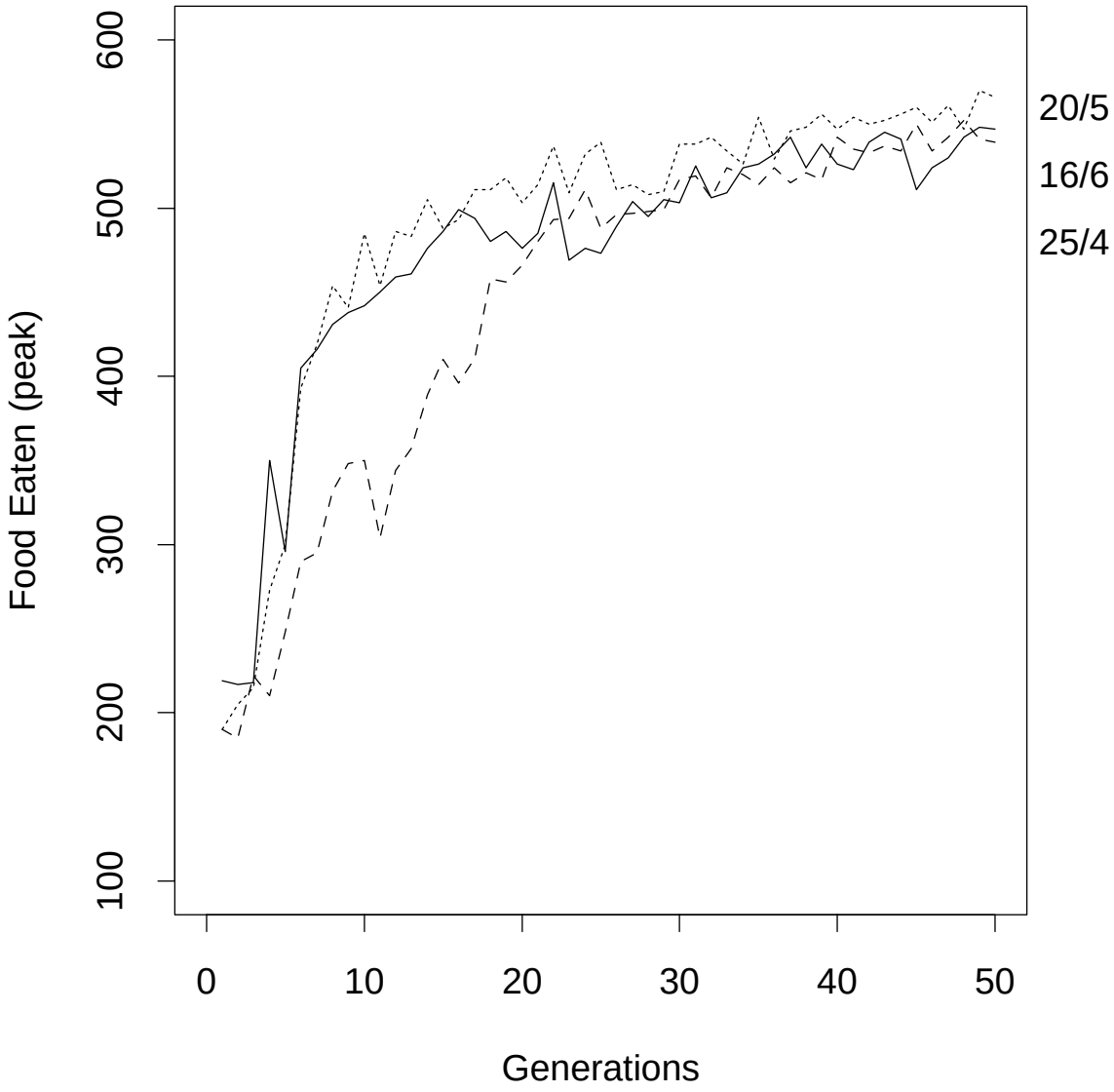
Food Eaten (peak)
100
200
300
400
500
600
0
10
20
30
40
50
Generations
20/5
16/6
25/4

Figure 13

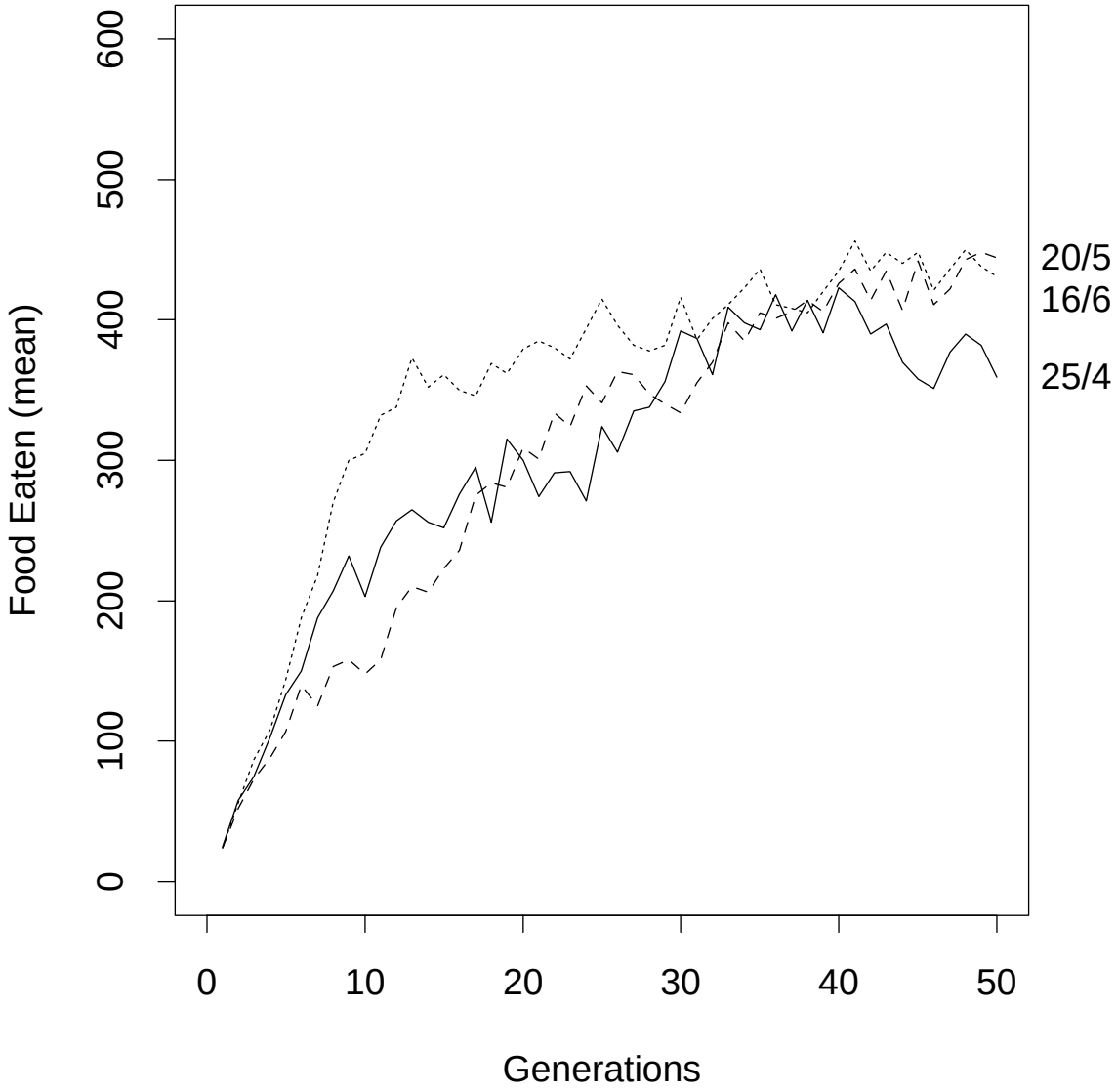
Food Eaten (mean)
0
100
200
300
400
500
600
0
10
20
30
40
50
Generations
20/5
16/6
25/4

Figure 14

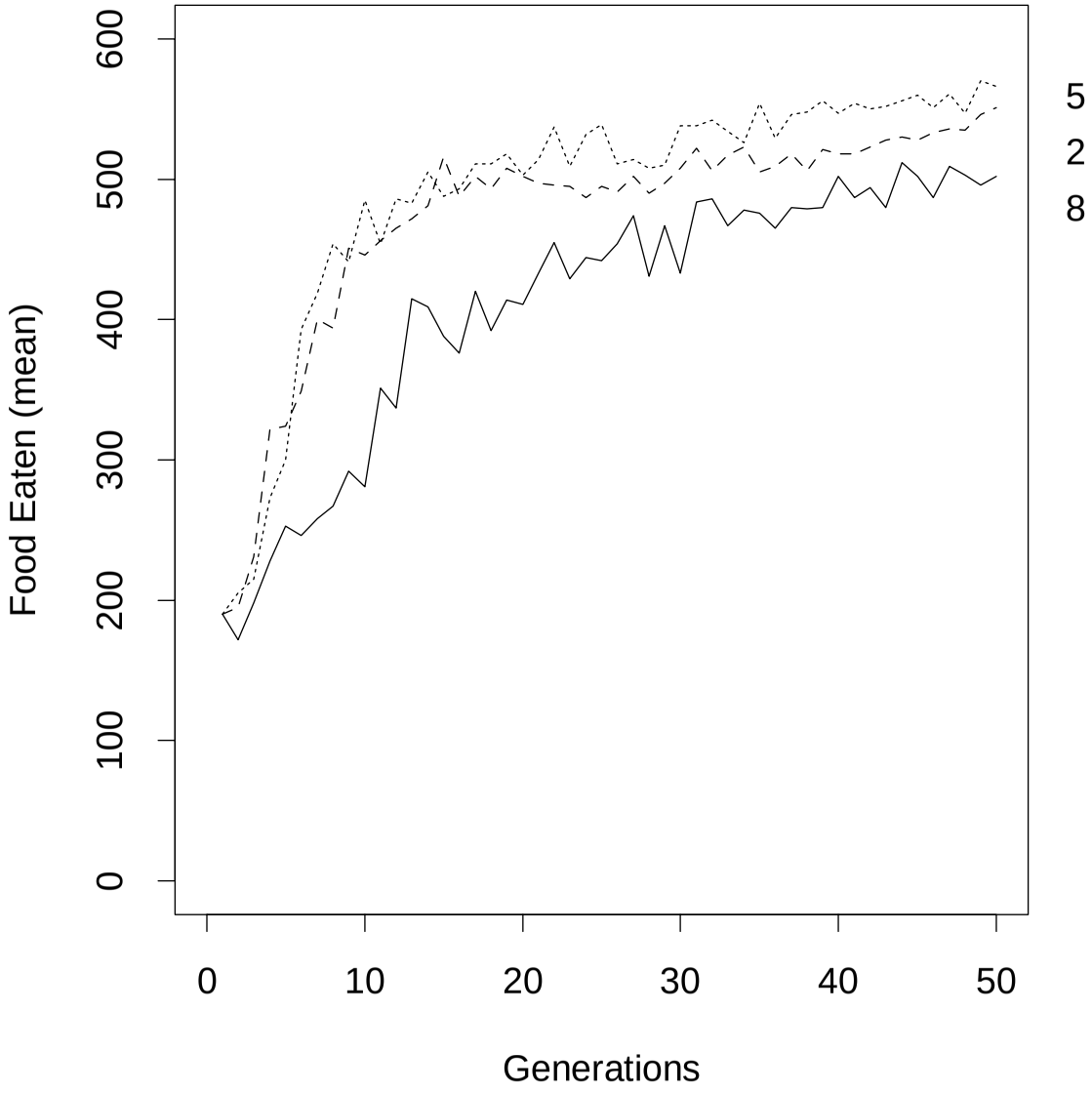
Food Eaten (mean)
0
100
200
300
400
500
600
0
10
20
30
40
50
Generations
5
2
8

Figure 15

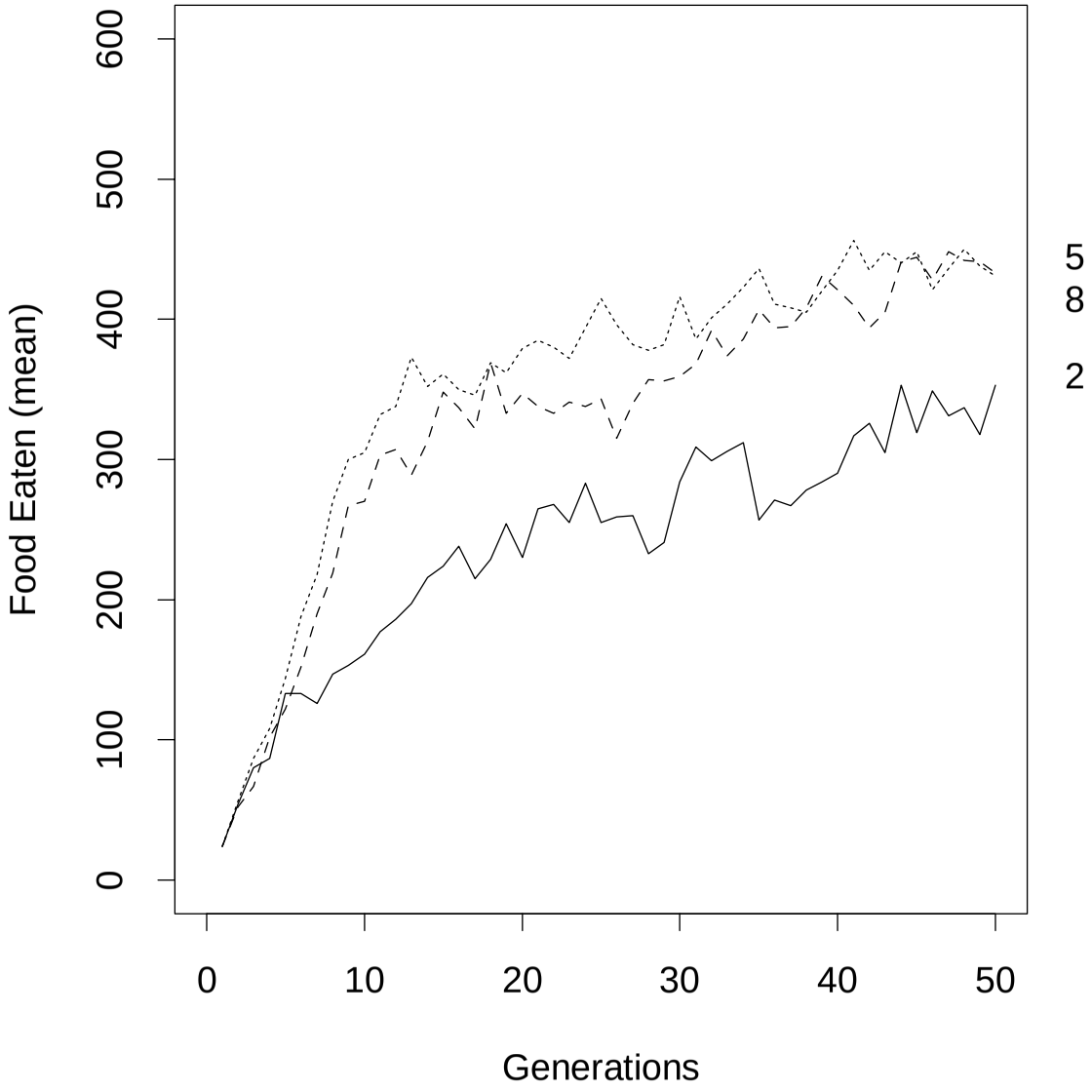
Food Eaten (mean)
0
100
200
300
400
500
600
0
10
20
30
40
50
Generations
5
8
2

Figure 16