
THE EVOLUTION OF MECHANISMS FOR MAKING FITNESS-RELEVANT DECISIONS

Written by Elles Jetten s2564564

MSc programme Biology; track Modelling in the life sciences

Supervisors: F.J.Weissing, M.Kozielska

Theoretical Research in Evolutionary Life Sciences

12-04-2020

TABLE OF CONTENTS

ABSTRACT	3
INTRODUCTION	4
MODEL OVERVIEW	6
<i>Agents</i>	<i>7</i>
<i>Environment.....</i>	<i>7</i>
<i>Reproduction and Inheritance.....</i>	<i>8</i>
RESULTS.....	9
<i>Simultaneous Assessment.....</i>	<i>9</i>
<i>Consecutive Assessment</i>	<i>11</i>
<i>Effects of Expanded Network.....</i>	<i>13</i>
<i>Multiple Cues and Cue Reliability.....</i>	<i>15</i>
DISCUSSION	20
CONCLUSIONS	22
LITERATURE	23
APPENDIX	24

ABSTRACT

Organisms are assumed to have evolved to maximize their fitness. However, fitness is a long-term goal, and decisions are made on the short-term. Short-term decisions are guided and evaluated by the emotion system, and it has been suggested that this system acts as a mediator of an organism's genes to set goals for behaviour. By means of a theoretical modelling study, I address the question how such an evaluation system is shaped by natural selection. To this end, I consider a population of individuals that have to make fitness-relevant decisions. Each individual is endowed with a heritable neural network that processes and evaluates environmental cues and thus guides the individual's patch choice decisions. The decisions made determine the individual's survival and fecundity, and, hence, its lifetime reproductive success. More successful individuals pass on their neural network to more offspring (subject to rare mutations). Over the generations, the neural network evolves, and it is to be expected that, in the end, the network's evaluation of a given habitat patch will closely match the fitness implications of this patch. In a series of individual-based simulations, I investigated the evolution of different types of neural network in slightly different decision situations.

In a first step, I considered the task of choosing one out of n simultaneously offered habitat patches that differ in food availability and predation risk. The simulation revealed that well-performing neural networks evolved rapidly, and that even very simple networks could manage this 'best-of- n ' task. Surprisingly, the networks evolved in replicate simulations differed considerably. Upon closer inspection, it turned out that the fitness landscape of all possible networks is very flat (a 'high plane') close to the fitness peak, implying that a broad spectrum of networks can achieve (almost) the same high performance. In a second step, I considered a consecutive assessment scenario: individuals could inspect one habitat patch at a time and had to decide on whether to stay there or to continue inspecting further patches. The simulations revealed that this task is much more difficult to master. Here an extension in the network was necessary to implicitly contain information of average patches in the environment. As they could not rely only on relative comparison between patches. Finally, I considered the effects of unreliable information and the possibility of having multiple cues on one piece of information. In the simulation it appeared that having multiple unreliable cues can compensate in their unreliability by being combined in evaluation.

INTRODUCTION

We often assume organisms behave in a fitness maximizing way. However, organisms frequently make decisions with short-term effects while fitness is a long-term goal. To behave optimally, organisms should have a way to determine goals on the short-term that will maximize fitness in the long run. Emotion has been suggested as a way for genes to determine goals for behaviour (Rolls, 2013), or similarly as related to putative goals of an organism (Bach & Dayan, 2018). Emotions that have evolved are fixed throughout lifetime and are elicited through information cues. Internal cues relating to the state of an organism and external cues with information about the environment are assessed and result in behavioural changes that are often fixed or pre-programmed to a degree.

Fitness-relevant assessments are needed for learning, to inform which behaviour was beneficial and which was not. Having an assessment or evaluation system and a learning system that guide decision-making allows for flexibility and adaptability. Instead of having a fixed behavioural response to certain stimuli, the combination of predetermined goals and learning can lead to multiple behaviours to obtain rewards. This flexibility allows for novel solutions to occur and can work around when one solution found does not work anymore. However, even though learning can result in flexible behaviour, when guided by emotion there still are limitations to this flexibility. For example, a fear reaction elicits a limited range of behaviour known as fight-or-flight and not all behaviour can be learned in fear induced state (Cain & LeDoux, 2007). Rats can learn to escape fear-inducing stimuli through some forms of operant conditioning. However, they are not able to learn to escape it by poking their nose through a hole.

This raises the question whether emotion or an evaluation system is a general mechanism that can be used to determine both fixed and plastic responses, dependent on the situation. It is important to ask questions about what mechanisms evolved that guide decision-making. Proximate mechanisms will have evolved to maximize ultimate fitness in complex environments. This may have counterintuitive results and, in some cases, may explain seemingly suboptimal behaviour (Vasconcelos et al., 2018). McNamara & Houston (2009) plead for a systematic study of the evolution of simple mechanisms in complex environments. When this is done, we can answer questions about why some behavioural “rules” or algorithms have evolved to be more flexible than others, which is the case with emotions. We can also answer higher level questions about why we find these particular organisational principles that guide decision-making. What conditions are favourable for the evolution of a decision-making structure where there are fixed emotions and semi-flexible learning?

While emotions are often discussed in the context of learning, the evolution of an emotional system can also be studied in the absence of learning. For example, Giske and colleagues (2013) modelled the evolution of a proximate, decision-making mechanism in fish. Their model includes an emotion system that receive sensory input and restricts attention during decision-making which leads to behavioural outcomes, foraging or fleeing. From their results they argue that emotions are necessary in giving a unified perspective on phenotype, as it combines genetics, external environment and physiology to predict behaviour. In simulations they found frequency dependent emergent strategies in different fish much like animal personalities. And it is able to predict when and why these personalities emerge.

The evolution of learning and emotion is a part in a bigger project in which my research takes a part in. Within this context, the goal I work towards is to shed light on the functioning of emotions in real-life

systems. It is a first step to gain more insight into the evolution of an evaluation system which is used for decision-making.

I wanted to answer the question: How might evolution shape an evaluation system? To achieve this, I first attempted to answer the following sub-questions: What are the effects of different decision-making mechanisms on the evolution of the system? What mechanism is necessary to evolve to assess information cues in a fitness optimal way? What is the effect of unreliability of the information on the evolution of an evaluation system?

To answer these questions, I made use of artificial neural networks as a mechanistic model for an evaluation system. Neural networks are used as a proximate mechanism of biological cognition. They provide many degrees of freedom for evolution to work on and are able to do many kinds of computations. I first consider the evolution of agents with an evaluation system that is input to a fixed decision-making rule. Investigating and understanding dynamics in this simple model is the main goal of this research, allowing in future studies to gain more insights into expanded models.

Evolution of the evaluation systems will be studied through an individual-based modelling approach with agents that pass on properties of their evaluation system with a chance of small mutations to simulate evolution. Individual-based models are useful for modelling situations where agents receive imperfect information about their environments, have differences in what they perceive from the environment and their options, but despite these differences the population evolves general rules. (DeAngelis & Diaz, 2019). In this report, the decision-making mechanism is fixed, and will not evolve. However, the evaluation mechanism used by the decision-making mechanism will evolve.

MODEL OVERVIEW

Here I give an overview of the general model structure. More specific aspects will be explained in more detail in the Results section. An individual-based model is made where agents must assess fitness-relevant cues (food and predator density) with the help of a heritable neural network. The network properties are transmitted from parents to offspring and are subject to mutations. The networks evolve because the most successful agents have the highest chance of producing the most offspring.

Two different decision-making tasks are considered, either agents assess patches simultaneously and decide afterwards where to go, or they assess one at a time and decide whether to stay or to go to the next patch.

In figure 1, a schematic of the program workflow is presented. In a first step, agents are initialized by providing them with a network that assigns the same suitability score to all habitat patches (all network weights are equal to zero). The environment is initialized by assigning values to the parameters governing the distribution of food and predators. In each generation, each agent is offered a set of habitat patches. Each patch is characterized by the amount of food offered and the number of predators present, and by food and predator cues indicating food availability and predation pressure. Taking the food and predator cues as its input, the neural network of an agent assigns a suitability score to the patch. As described in more detail later, these suitability scores determine the patch choice of the agents. Once an agent has chosen a habitat patch, the agent's survival will be determined by a chance process, where the probability of survival is positively related to the number of predators in the patch. Surviving agents produce offspring; the number of offspring produced is proportional to the amount of food in the chosen habitat patch. The offspring inherit the network parameters from their parent, subject to mutation. When all offspring are produced, the parents of the current generation are removed (discrete, non-overlapping generations), and the offspring take the role of their parents in the next generation, where the whole cycle is repeated.

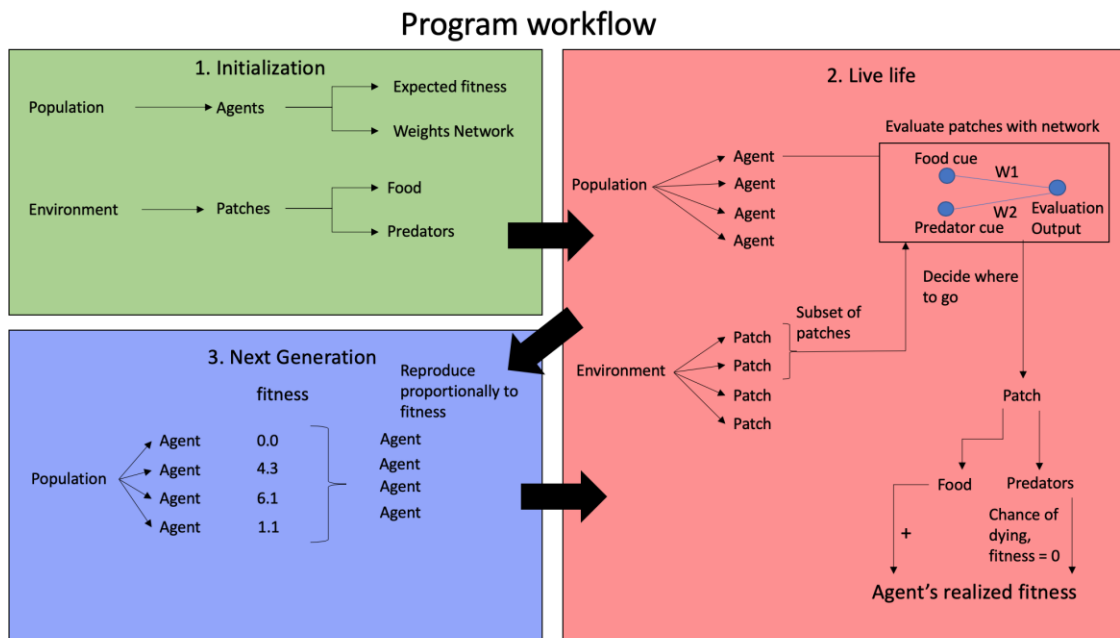


Figure 1 Schematic of the program flow for simulations

AGENTS

Agents have evaluation networks they use to assess the suitability of their environment. This evaluation network is a very simple neural network with some input nodes. To start as simple as possible, I first considered networks with only two inputs. The inputs of the network come from cues, c_f and c_p , that contain information about food availability and predator density in a given habitat patch. The inputs are multiplied by the respective weights in the network, also called w_f and w_p from this point. So, the suitability or evaluation value will take the following value:

$$eval = w_f * c_f + w_p * c_p$$

Therefore, the weights determine the functioning of the network by deciding the output of the network as is depicted in figure 1. Depending on the choice scenario (see below), the agent uses the suitability scores assigned to habitat patches for choosing one patch where it will settle and spend the rest of its life.

ENVIRONMENT

The environment in the model is divided into 50 discrete patches. These patches contain food and predators that affect the fitness of the agents. During simulation agents must choose a patch from the environment using their evaluation network. There are two ways implemented in which way they chose a patch from the environment; those two ways are described in more detail below. Once they have chosen a patch, predators in the patch have a chance of killing an agent and reducing its lifetime reproductive success to 0. For n number of predators and for p_d as the probability to die from one predator, the chance of surviving p_s is given by:

$$p_s = (1 - p_d)^n$$

For all the results discussed in this report food is assumed to be drawn from a uniform distribution between 0 and 10 and predators from a Poisson distribution with $\mu = 2$. Aside from the realized fitness of an agent, another variable is considered in this model which is expected fitness. This is defined as the fitness gained from food intake, f , in proportion to the chance of surviving the predator attacks, p_s :

The fecundity of the agent is set proportional to (in fact equalized with) f , the amount of food found in the patch. Accordingly, an agent has a 'realized fitness' of zero if it is killed by a predator and f if it survives predator attack. Since it avoids predation with probability p_s , the expected fitness of the agent is given by:

$$F_{ex} = f * p_s$$

Expected fitness gives a better indication of the performance of a neural network, because a good decision (i.e., the choice of a good habitat patch) might still result in 0 realized fitness.

The patches are initialized by drawing food and predators from random distributions. For all the results discussed in this report food is assumed to be drawn from a uniform distribution between 0 and 10 and predators from a Poisson distribution with mean $\lambda = 2$.

REPRODUCTION AND INHERITANCE

After all agents have made their decisions and obtained their realized fitness score, reproduction and inheritance take place. For this, the population of 1000 agents is replaced by 1000 offspring. Generations are non-overlapping and reproduction happens based on a weighted lottery with realized fitness as the weighing factor. This means that the expected number of offspring produced by a given agent is proportional of the realized fitness of this agent. Agents are haploid and reproduce asexually, meaning only one parent is needed per offspring. Offspring will inherit the weights of the evaluation network from their parent. Each weight can mutate with mutation probability 0.02 per weight, a small real number is added to the parental weight; this 'mutational step size' is drawn at random from a normal distribution with mean zero and standard deviation 0.01. The interaction of selection (the best-performing networks tend to produce more offspring) and mutation results in the adaptive evolution of the weights and, hence, the evaluation networks.

RESULTS

SIMULTANEOUS ASSESSMENT

In the simultaneous assessment scenario, each agent uses its network to evaluate a set of 10 randomly chosen habitat patches. Subsequently, each agent settles in the patch with the highest evaluation value. Figure 2AB shows how the network weights w_f and w_p evolve in the first 20,000 generations of a typical simulation run. A simple calculation reveals that two networks with the same weight ratio w_p/w_f are selectively equivalent, because they result in exactly the same rank order of the 10 patches. Therefore, Figure 2C shows the evolution of this weight ratio.

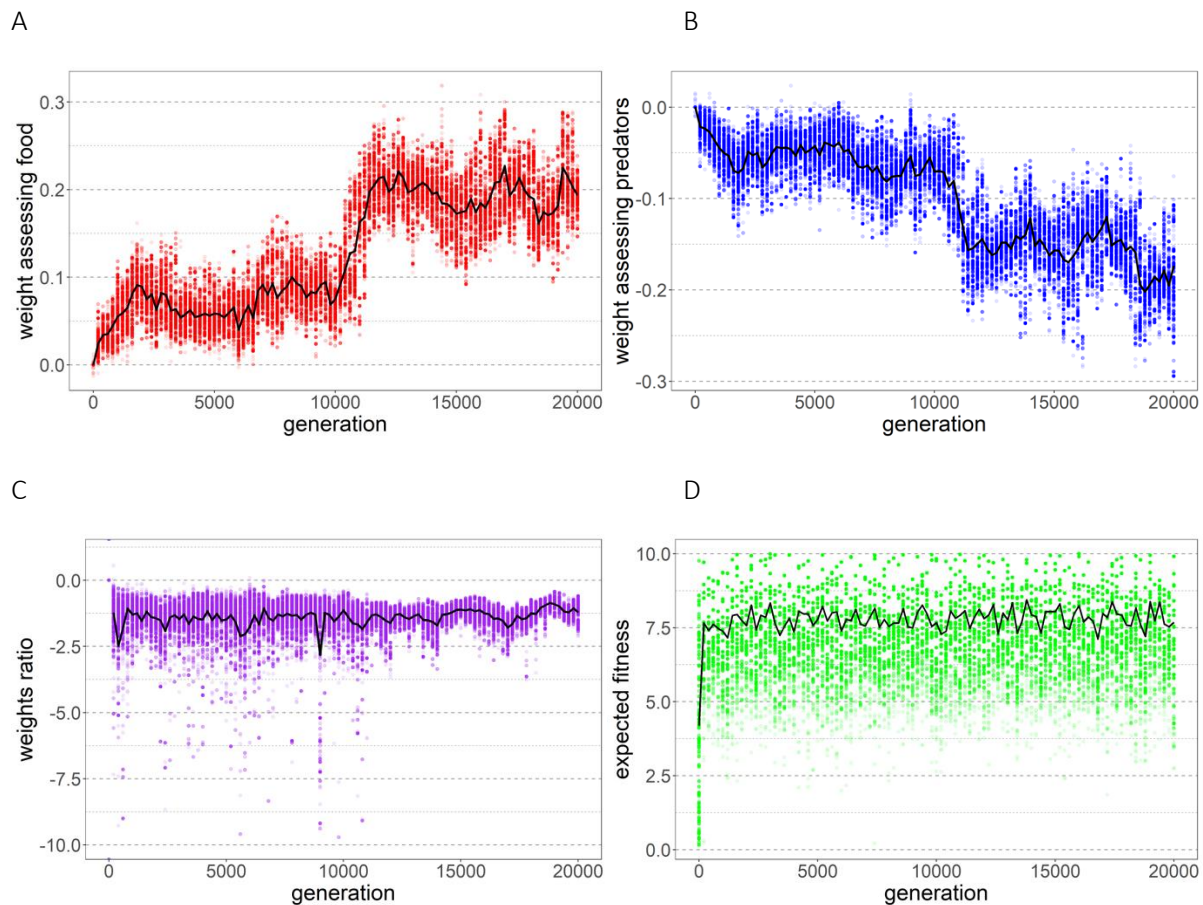


Figure 2, evolution of evaluation networks in a simultaneous choice scenario. Top panels show weights assessing food and predators, the bottom left the ratio of these weights and bottom right shows the expected fitness over generations. The black lines indicate the average in the population and the dots are individual values.

In figure 2A and 2B we see w_f that assesses food and w_p that assesses predators evolve. w_f and w_p seem to inversely parallel each other, suggesting the ratio of the weights are important. The ratio of the weights is presented in figure 2C. We see the ratio converging. However, there remains seemingly wide range of ratios

present in the population. I would have expected a certain ratio be optimal and be approached over time. The large amount of variation in the weights and the ratio of the weights is notable. Looking at the fitness of the agents, in figure 2D we see that fitness very quickly rises but does not improve much more after a few generations. This indicates a weak selection pressure and shows an equilibrium has been reached.

To investigate both the variation in weights and ratio and the seemingly lack of selection pressure I looked at the performance of different ratios. This performance is measured in terms of fitness loss, which is the difference in fitness between the phenotype of the agent and the fittest phenotype possible. This term is related to the genetic or mutational load (Agrawal & Whitlock, 2012) of the different weight combinations of an agent. It signifies how deleterious it is to have a certain genotype.

To investigate systematically how the performance of a network depends on its ratio of weights w_p/w_f , we created many such weight ratios ranging from -2.0 to -0.5. For any given weight ratio, a network was produced by setting the food weight to $w_f=0.5$ and the predator weight to $w_p = (w_p/w_f)*0.5$. Subsequently, the performance of the network was assessed by letting it rank 10,000 randomly produced sets of 10 habitat patches. From each evaluation round the highest expected fitness that could be achieved by picking the best available patch, F_{best} was taken, as well as the fitness of the patch chosen by using the evaluation network F_{chosen} . The fitness loss per round F_l , is calculated and is defined by:

$$F_l = F_{best} - F_{chosen}$$

When F_l is 0, the best patch was chosen from the subset. When it is non-zero another patch has been chosen that may be close to or further away from the optimum, signifying the size of the mistake that that has been made.

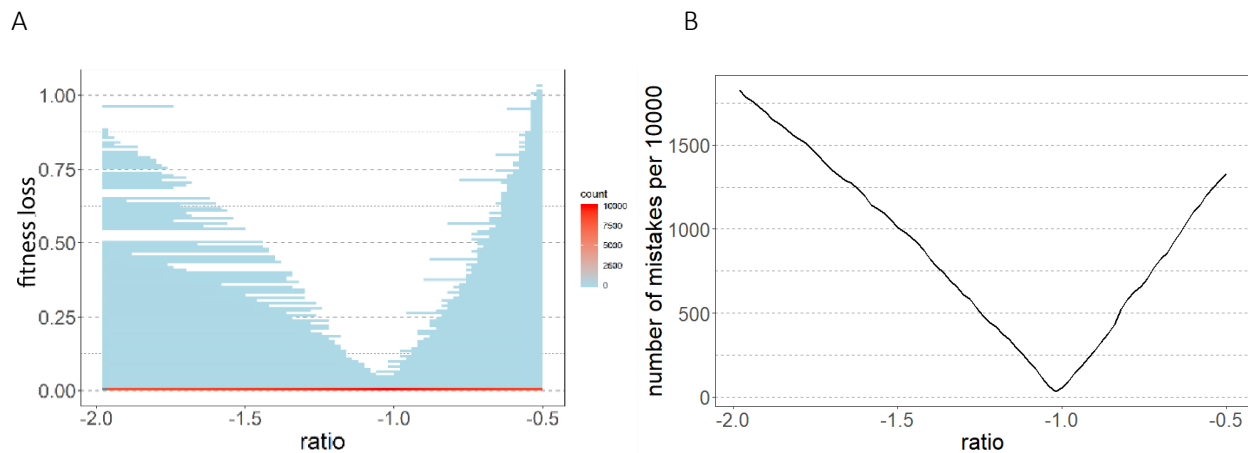


Figure 3. fitness loss and number of mistakes in a fixed weights simulation, no evolution occurs. The left panel shows a density plot of the size of fitness loss for each ratio while the right panel shows the number of mistakes. Every ratio assessed 10000 rounds of randomly generated patches.

In figure 3A we can see a density plot where the size of the fitness loss is shown. The smallest mistakes are made around a ratio of just under -1.0. We see that the surrounding ratios have an increased average fitness loss, of up to 1.0 in the ratios observed ratios from the simulation described above. In figure 3B the number of mistakes is shown, where again around this ratio the least number of mistakes are being made.

There seem to be a clear optimal ratio where fitness is highest, however, in most of the cases, non-optimal ratios also make optimal decisions. Surrounding ratios still have a low rate of mistakes, with 1000 being a 10% mistake rate. The mistakes that are made are small as well, around 0.5 deviation while fitness ranges from 0 to 10. There seems to be such a little difference between the optimal ratio and ratios around the optimal that the force of selection has less influence than the stochastic mutations resulting in a lot of variation. A wide range of ratios can make close to optimal decisions. This can be seen as well in figure 4. The fitness landscape is very flat, and a wide range of weights achieve optimal behaviour.

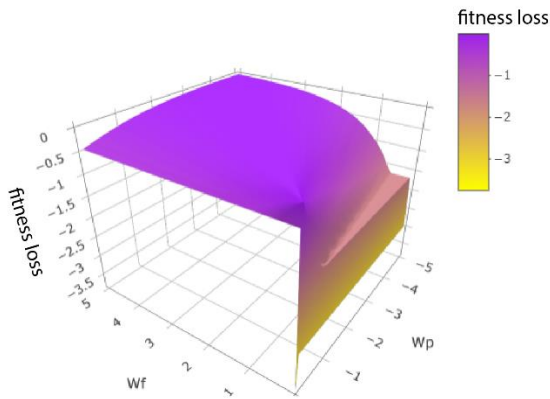


Figure 4, fitness landscape in simultaneous choice scenario. Weights for each assessment were fixed.

This conclusion is robust and does not change in models with continuous mortality rates in the patches instead of discrete predators.

CONSECUTIVE ASSESSMENT

In consecutive assessments agents assess one patch at a time. When a patch is assessed the evaluation-value is interpreted as the logit value (Wakefield, 2013) of the probability to settle in the patch. In other words, if the evaluation value of the patch is $eval$, the probability to settle in the patch is given by (see Figure A1 in the appendix for an illustration):

$$p(eval) = \frac{1}{1 + e^{-eval}}$$

If an agent rejects the patch, it will move on to the next patch in the environment, but is unable to go back to previous patches. However, agents cannot be infinitely picky, after 50 rejected patches they are forced to take the last patch. This is to simulate the time constraint they have in picking a patch. It also functions as a cost to being too picky, because after 50 patches they are forced to take the one that is left. The fitness loss in this case would probably be high.

A

B

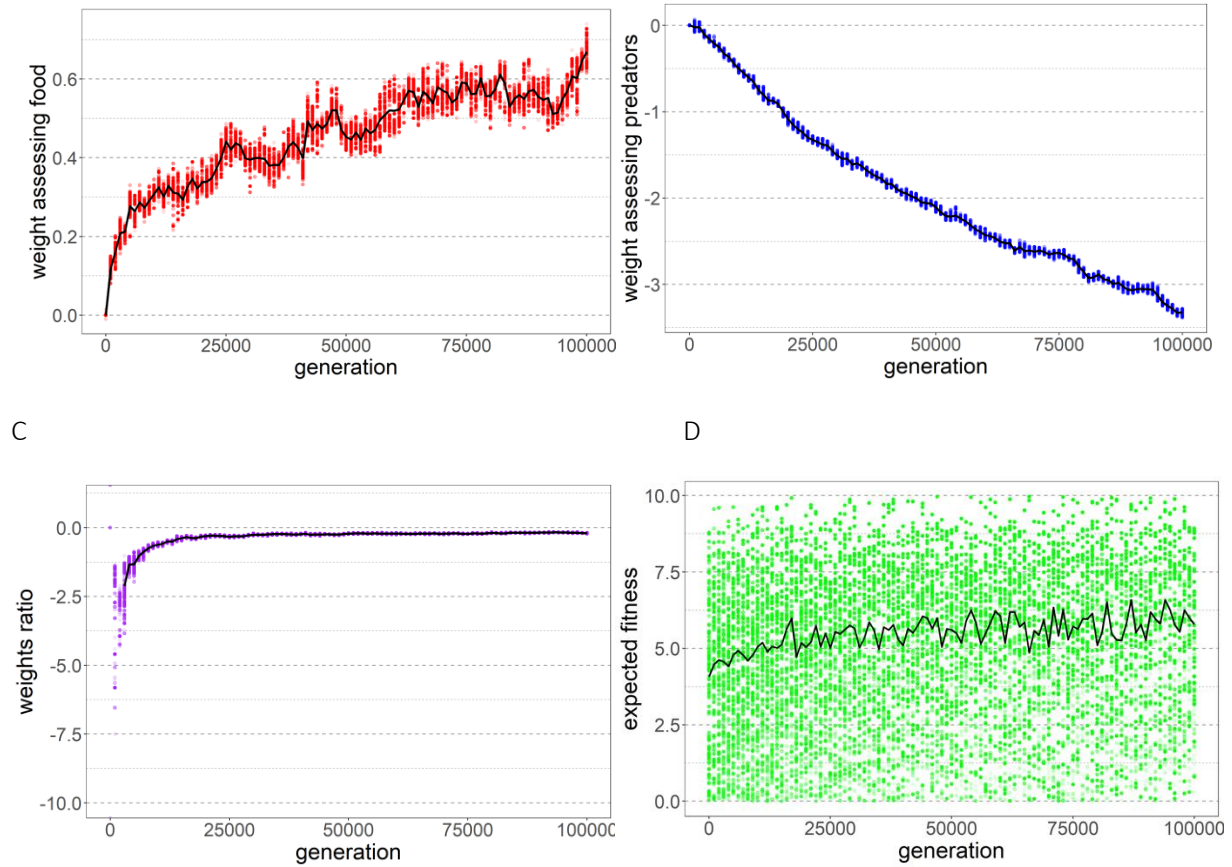


Figure 5 evolution of weights of an evaluation network in consecutive choice scenario. Top panels show evolving weights, the bottom left panel shows the ratio of these weights over generations. The bottom right panel shows the expected fitness over generations. All black lines indicate the average in the population

Compared to simultaneous assessment, looking at figure 5B w_p seems to evolve gradually and stochastic effects do not have such a large influence. In figure 5A it seems w_f starts off evolving in a clear direction after which it flattens and fluctuates. The ratio of the weights converges over the course of the simulation as seen in 5C. In figure 5D we see the fitness of the agents growing over generations. However, in these figures it is unclear whether the weights might have stopped changing, as the fitness has not levelled out.

EFFECTS OF EXPANDED NETWORK

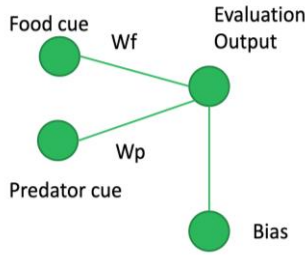


Figure 6. Illustration of a network that is expanded by adding a bias

For the following simulations I look at the effect of an expansion to the network. In the following simulations there will be a constant that can evolve added to the network. So, the way the evaluation is calculated will change from just inputs * weights, to the following equation:

$$Eval = w_f * fc + w_p * pc + b$$

Where fc and pc are food cue and predator cue and b is the bias, an evolving value that is independent of information cues. A schematic of the network with a bias can be seen in figure 6. Because the evaluation value in the consecutive assessment is transformed by a sigmoidal curve to a probability, not just the ratio of the weights is relevant for functioning, but also the absolute evaluation value. A bias is a good way to shift the range of possible values to increase choosiness in agents. For example, if a patch contains neither food nor predators, without a bias the evaluation value will always be 0. See figure 7 for a visual representation of the sigmoidal curve with different values for the bias. If for example the patch found had no food and no predators, the chance of staying with a bias=0 will be 50%. In the case where bias=5 it will be around 100% and with bias=-5 it is almost 0%.

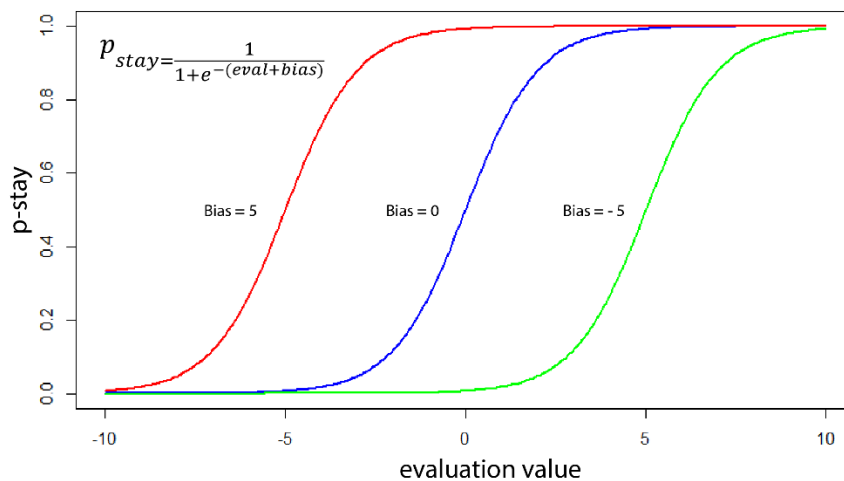


Figure 7 the sigmoidal function used to translate an evaluation value to the probability of an agent to stay on a certain patch, for different values of the bias.

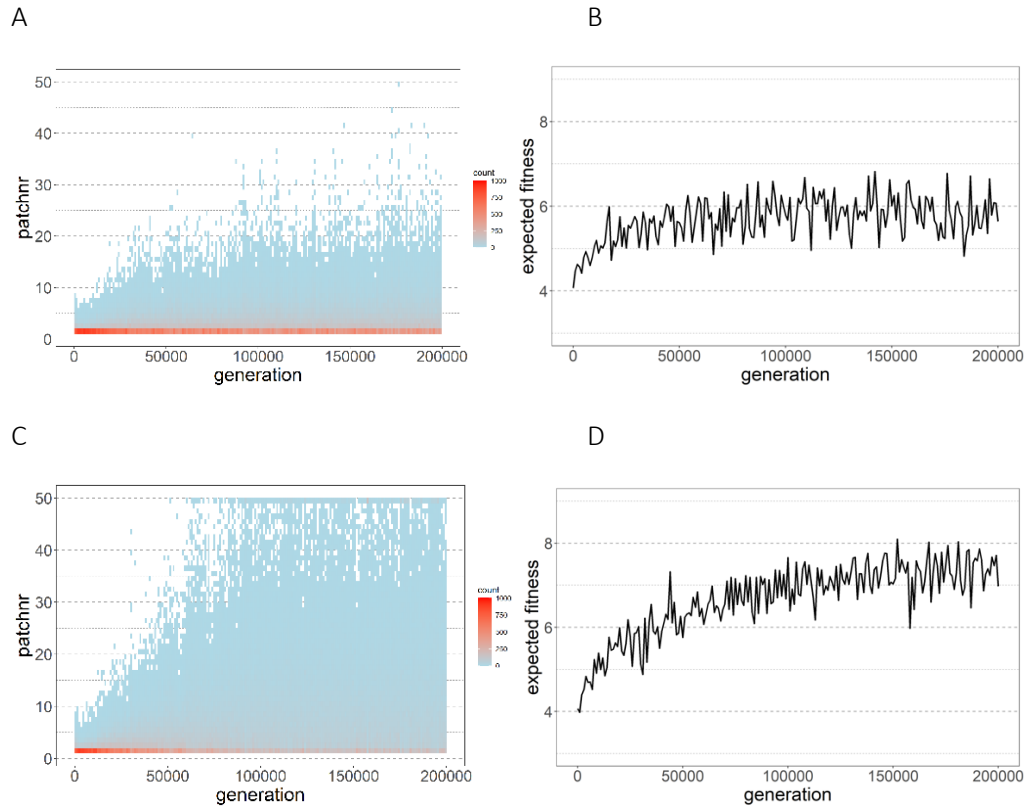


Figure 8 evolution of networks in consecutive choice scenario comparing simulations of networks with and without bias. the top panels show results without an added bias to the network and bottom panels show results with a bias added. Left two panels show density plots of the chosen patch number in the assessment round, higher numbers indicate a higher choosiness. The two panels on the right show the average expected fitness of the population.

In figure 8A compared to 8C, we can see that an evolving bias leads to an increased choosiness of the agents. 8B and 8D show that this choosiness leads to an increased expected fitness of the agents. An added bias in the case of consecutive assessment is needed to lead to better decision-making in the case of this decision-making rule.

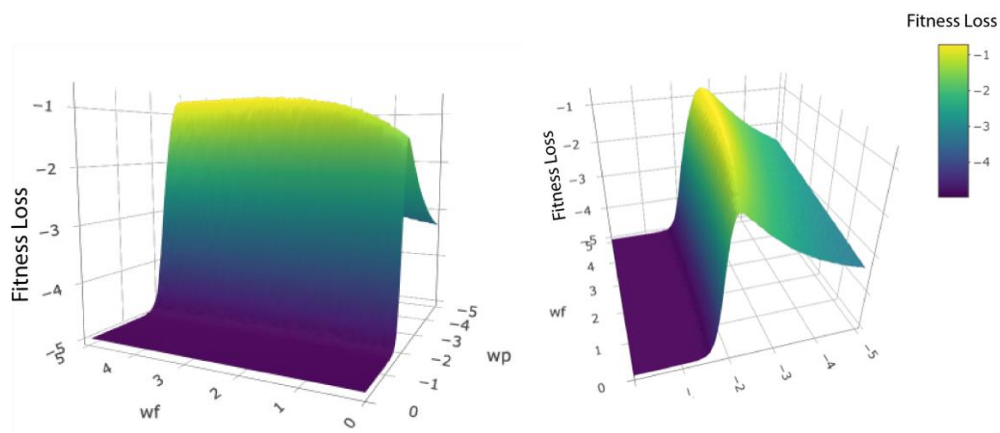


Figure 9, 3D plot of a fitness landscape of the consecutive choice scenario. Weights and bias are fixed and do not evolve. The value for the bias is the same for all points, it is set to -10. The two panels give a view from different angles.

Figure 9 displays different fitness loss for different combinations of weights. As in the simulation for simultaneous assessment, the fitness loss is the difference in the best available choice and the choice made. Different weight combinations were fixed during simulation and networks with these combinations chose patches consecutively for 1000 rounds. The bias of the network was fixed to -10 in each case, this value was chosen on the basis of the resulting network in simulations where the bias was allowed to evolve. In figure 9 we find a ridge of optimal values of the weights. When the optimal is reached there seems to be multiple weight combinations on the ridge with similar fitness values. This may cause some random movements of the weights on the ridge once it is reached in simulations. The ridge's position is dependent on the value of the bias.

Comparing the fitness landscape of simulations where patch choice is done by consecutive assessment as opposed to simultaneous, we see a narrower region for the optimal values of the weights. The fitness landscape and therefore the course of evolution differs between different decision-making mechanisms.

Another effect I considered is a fitness cost for rejecting a patch. We can easily imagine that rejecting patches reduces the time left to forage and increases the energy that is necessary and thus reduces fitness. Results gained so far have turned out to be robust to the implementation of cost, as in the behaviour of the system or what evolves has not changed much. Cost is implemented by reducing the fitness for every rejected patch. A small cost of 0.1 per rejected patch is used for results from figure A2 in the Appendix. According to expectations, an added cost to rejection leads to networks that evolve with lower pickiness. The average number of patches that are rejected declines as it is no longer worth for the agents to wait out for a really good patch. With it the fitness of the evolved networks do not reach the same values.

MULTIPLE CUES AND CUE RELIABILITY

So far, I assumed agents have perfect knowledge of their environment. However, in the next simulations we wanted to see what the effects of unreliable information cues are on the evolution of the evaluation network. For the following results the consecutive decision-making rule is used.

There are multiple ways to model lower reliability and add noise to the information cue from a patch. Methods used here are adjusted from (Botero et al., 2015). First, the noise of the cue can be added from a uniform distribution. With lower reliabilities the range of noise should get larger, until with completely unreliable cues it encompasses the range of the entire environment. Not just noise is added in this case, but it is added onto the food or predator value in the patch that is adjusted toward the average in the environment. So that with completely unreliable cues there is no information at all from the cue, it is just a random number drawn from the same range as the environment. Reliability is given a value of 1.0, perfect information, to 0.0, no information about the patch. Food cues are drawn from a uniform distribution with the following parameters:

$$\mu = \hat{f} * R + (1.0 - R) * \bar{P}_{env}$$

$$[min, max] = \mu \pm (1.0 - R) * \bar{P}_{env}$$

Here \hat{f} is the actual value of the food or predators in the patch. \bar{P}_{env} is the average value of food of the patches in the environment, and R is the reliability of the information cue.

The second way I modelled unreliability is by just adding noise drawn from a normal distribution to the value of food found in the patch. Food cues are drawn from a normal distribution with the following parameters:

$$\mu = \hat{f}$$

$$\sigma = \frac{\bar{P}_{env}(1.0 - R)}{3}$$

In this case, when reliability is set to 0.0, the noise is drawn from a normal distribution with a standard deviation of a third of the average of the environment. Plus and minus three times the standard deviation of a normal distribution will be in the range of 99.73% of all values that fall in that normal distribution. This way, the range of the noise will be close to the range of values found in the environment. Graphs of cue distributions for different reliabilities and distributions can be found in Appendix figure A3.

For predators a similar approach is taken, however because predators in the environment are drawn from a Poisson distribution the range from the environment needs to be calculated differently. The Poisson distribution from which the predators is drawn in the simulations has a mean of 2 unless mentioned otherwise. So, to get a distribution for cues with a comparable range as predator distribution of the environment, I calculate the range where 95% of the Poisson distribution values fall. The probability that k predators are in the patch is given by:

$$p = \frac{\lambda^k e^{-\lambda}}{k!}$$

Where λ is the mean of the Poisson distribution and k is the number of predators. I calculated the k_{95} predators where $\sum_0^k p \geq 95\%$ to get the approximate range of the Poisson distribution. Now cues for predators are drawn from uniform and normal distributions with parameters as calculated above, but $P_{env} = \frac{k_{95}}{2}$. Graphs showing the predator cue distributions for different reliabilities is shown in appendix figure A3.

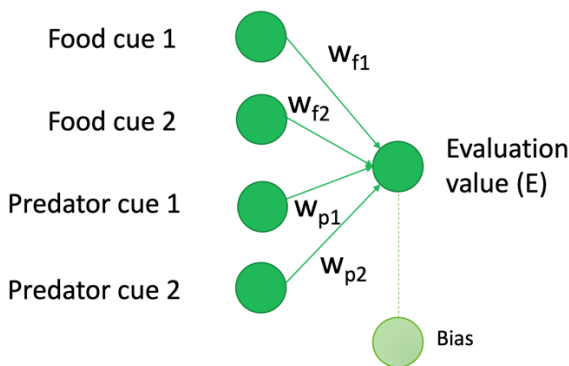


Figure 10, schematic of the network with four input cues and bias

In the following results, the network is expanded to have 4 input cues. Two each for food and predators. See figure 10 for a schematic of the used network. The reliability of these cues may be different from each other. Reliabilities will vary from 0.0-1.0 and cues are dependent on them as described above. The

combination of reliabilities will both be applied to the predator and the food cues. For example, with $R=1.0$ and $R = 0.0$, the food cues have $R= 1.0$ and $R=0.0$ and the predator cues will have $R=1.0$ and $R=0.0$.

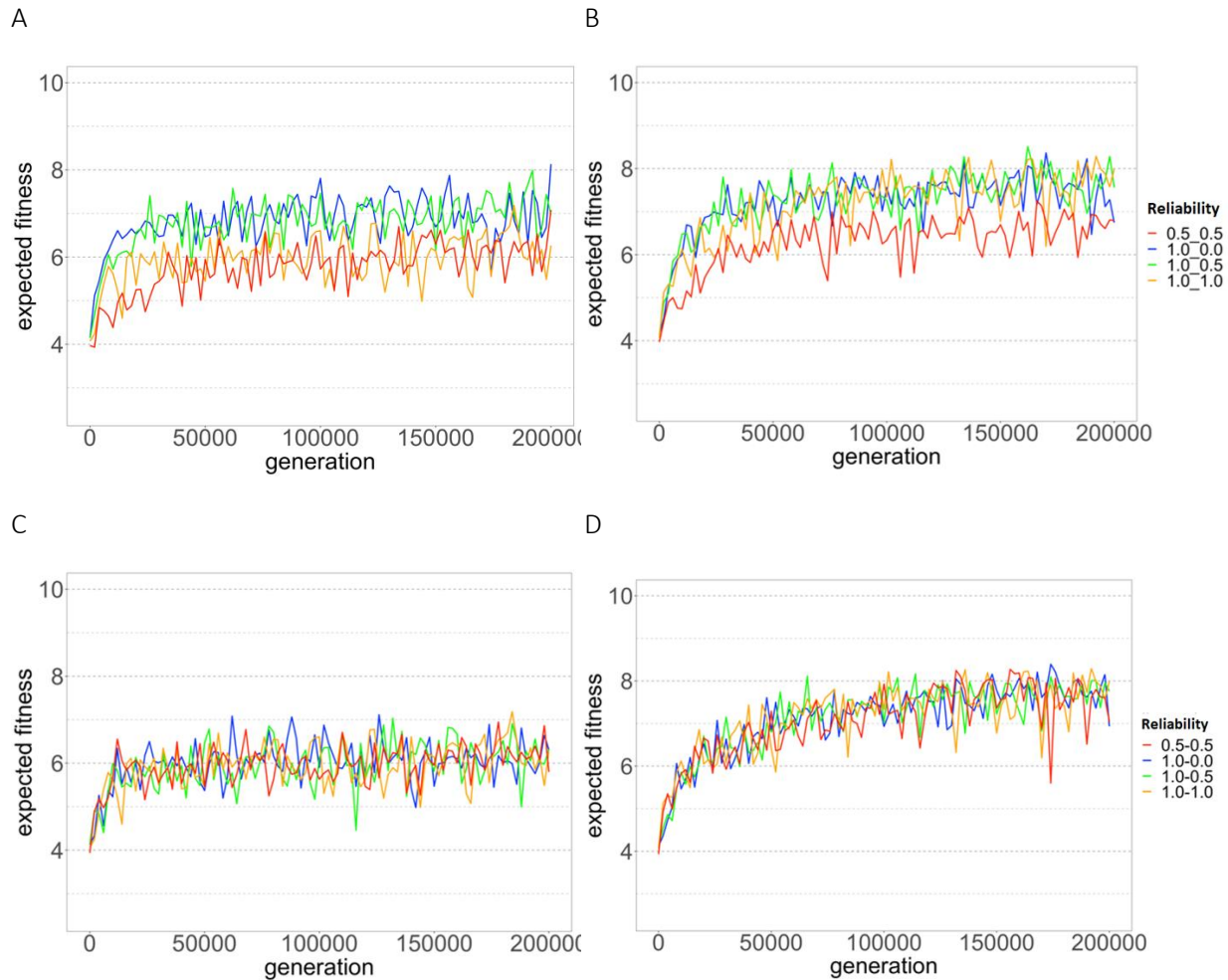


Figure 11 each panel shows four simulations of the evolution of expanded evaluation networks in a consecutive choice assessment. Different colors indicate different reliability combinations of the cues that are input. The top panels have unreliable cues drawn from a uniform distribution and according to the Botero methods. The bottom panels have unreliable cues drawn from a normal distribution. The left panels are simulations where networks are not expanded with a bias, and the right two panels are from simulations where bias is added to the networks of the agents.

In figure 11A and 11B we see the expected fitness of different reliabilities evolved with and without a bias with cues drawn for a uniform distribution with adjusted means. The added bias has the effect of increasing the mean fitness of the population. Between reliability combinations there is no clear difference in average fitness. A combination of two unreliable cues, $R = 0.5_0.5$, is just as efficient as having two fully reliable cues. It appears having multiple information sources compensated for the unreliability of the cues.

In figure 11C and 11D, we see the mean expected fitness for different reliabilities with and without bias and cues drawn from a normal distribution. The result shown at first glance may contain some surprises. In the case without bias, the combination of reliabilities that perform the best are with one cue fully reliable

and one cue of lower reliability. While when having two fully reliable cues, they don't perform just as well. This unexpected result can be explained when looking at how the unreliable cues are determined. The closer to 0.0 reliability the cues get, the more they will on average be equal the average of the entire environment. Meaning, that if an agent finds a patch where the unreliable cue is much lower than the fully reliable cue, the patch found is a good patch compared to the rest of the environment.

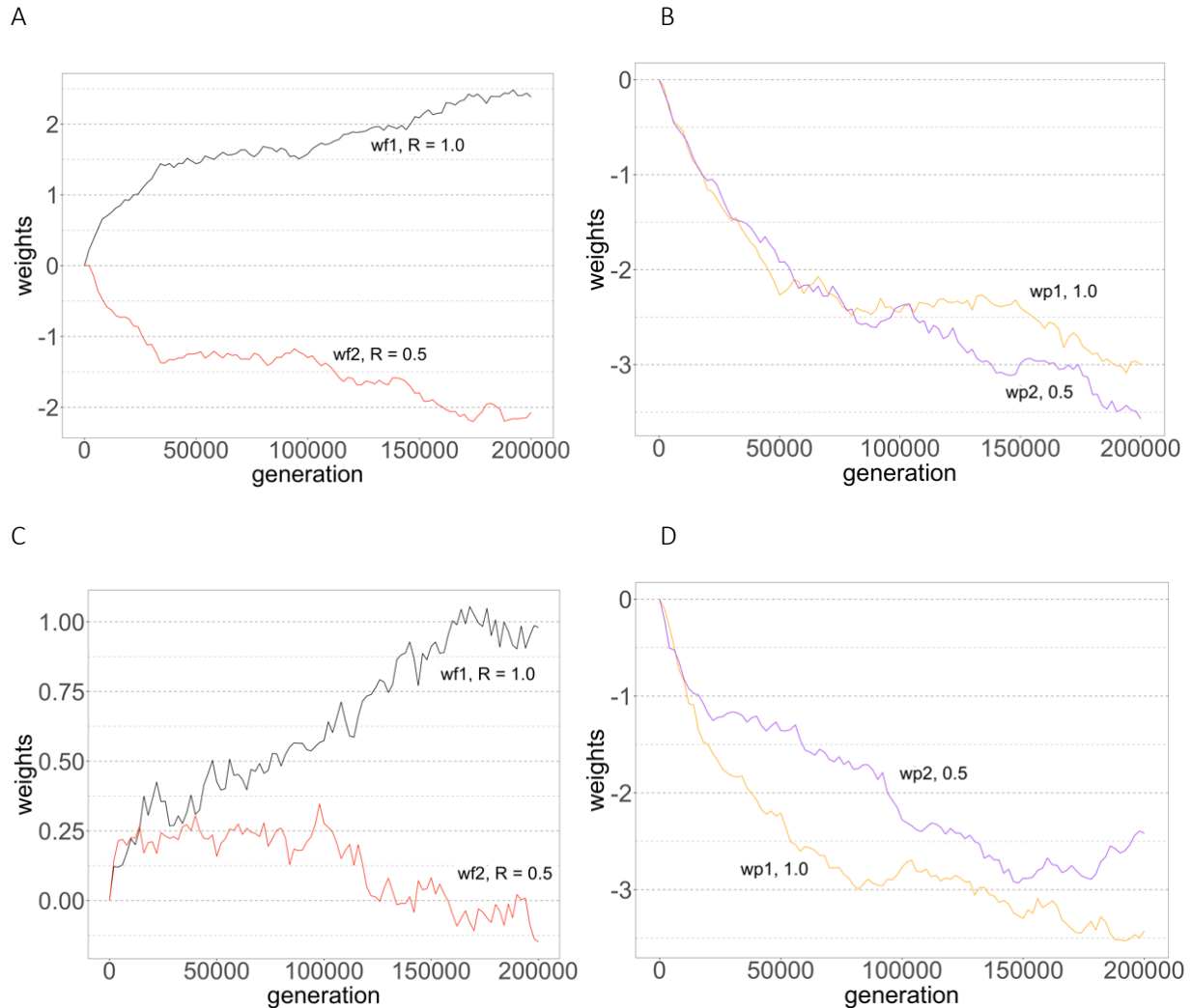


Figure 12 the evolution of evaluation networks with 2 weights assessing food and 2 weights assessing predators, reliability of weights differ as can be seen in the figure. In the top panels cues that are not completely reliable are drawn from a uniform distribution and using the Botero (2015) methods while bottom panels are drawn from normal distributions

Looking at figure 12A, we see the weights that have evolved where one cue had $R = 0.5$ and the other $R = 1.0$ with cues drawn from a uniform distribution with adjusted means. The weight paired to the unreliable cue evolves to a number below zero. Because no bias evolves in this simulation, there is no way to lower all evaluation values to make sure it is sensitive to good or above average patches. However, with the unreliable cue the agents can compensate by using the information that is in the unreliable cue: information about averages in all patches in the environment. In figure 12B we see that for the weights

that assess the predator cues this “trick” is not possible. Because the weight should already evolve to a negative value.

This result is further confirmed by results in 12C, where different cue reliabilities evolve without bias, but unreliable cues are drawn from a normal distribution where the mean is not adjusted towards the mean in the environment. The weight for the unreliable cue evolves towards a positive value and seems to evolve around lower values than the weight for the fully reliable cue. It is not possible to gain much information about the environment from this cue, and therefore we do not see it evolve like does in simulations where cues are drawn from uniform distributions with adjusted means.

The weights for lower reliability cues evolve to values closer to zero than the fully reliable cues. As expected, because agents already receive perfect information it does not make sense to look at unreliable cues at all. However, the values are not completely 0, which might mean it does not matter too much. To test this, I could have allowed for a mutation that would remove the unreliable cue input from the network and see if that would evolve.

DISCUSSION

From my model it is shown that decision-making mechanisms like the simultaneous assessment and consecutive assessments affect the fitness landscape and evolution of the population. My results show that the fitness landscape of simultaneous decision-making is very flat, and a number of weight combinations achieve optimal or near-optimal assessment. Whereas the fitness landscape of the consecutive assessment task showed a narrower ridge on which the weight values performed best. These results show the importance of modelling decision-making as a whole, as simplifications made about the decision-making rule will affect the evolution of the emotional and eventually also learning system.

I furthermore show that simple neural networks can evolve to readily assess information cues and improve performance compared to random choice, as they do in the first generation when weights are initialized to 0. In the simultaneous assessment task, the simple neural network is able to reach optimal performance. However, in the consecutive assessment task the population greatly benefitted from having a bias added to the network.

I looked at what the effects are of having unreliable information cues. When receiving unreliable information, I have shown that having multiple information sources can compensate for the noise of the sources. On the other hand, I expected that when receiving a fully reliable cue, that would be the only information source taken into account. While I did find that the unreliable cue in this case approximates 0, there are also some fluctuations of the weight. A definitive answer to whether just having a fully reliable cue would evolve could be obtained by introducing mutations in the topology of the network.

Furthermore, I obtained surprising results when investigating unreliable cues drawn from uniform distributions with adjusted means. I found in the combination of a fully reliable cue and one unreliable cue that the weight for the unreliable cue evolves towards a negative value. As described above this can be explained by overlooking that the unreliable cue now contains information about the average food value found in patches in the environment. In absence of a bias in the network, the agent can use this to their advantage and shift the evaluation values of most bad patches downwards. This result is interesting from a model design point of view rather than a biologically relevant one. My goal was to implement unreliable cues that contain less information about the patches, but by choosing a distribution with adjusted means I accidentally gave them information about their environment. Once bias was added to the evolution, I saw that this “trick” was no longer necessary. The bias evolves to a value that shifts the chances of an agent staying on a patch which was independent of the cues but dependent on the distributions of food and predators in the environment. Which was a nice control to get out of this design flaw.

In the consecutive choice scenario, an improvement was seen in the performance when a bias was added to the network. However, it is difficult to determine whether the simple neural network is able to achieve the optimal strategy. Without the ability of going back or having perfect knowledge, it is impossible to always pick the best patch from the subset of 50 patches the agent assesses. What is possible is for knowledge about the environment, what distributions food and predators are drawn from to initialize the patches, to be contained implicitly in the weights and bias of the network. So, they would have knowledge about what patches they encounter are above average compared to the entire pool of patches. What they are not able to is sense how many patches they have already rejected. One could imagine more readily taking an okay patch when you have rejected 45 out of 50 compared to when you are encountering your first patch.

It may be possible to mathematically determine the optimal stopping strategy. In (Kuchta, 2017) a threshold was formulated that results in the highest probability of choosing the highest element out of a sequence with elements drawn from a known distribution. To determine whether you should stop however, the size of the sequence has to be known and the number of the element that is being assessed. In my simulation agents did not have access to this information. It might be interesting for future research to add this information to the input that agents receive. It would also be good to compare the performance of the agents to the mathematically optimal stopping formula, however there are some difficulties in doing so. First off, agents in the simulation try to maximize fitness, whereas in this paper they formulated a way to maximize the probability of getting the highest fitness from a sequence. Meaning they would never take a patch that is not the highest seen until that point, which often would result in fitness loss. The second difficulty comes in knowing the distribution of expected fitnesses from the patches. This is determined from food which comes from a uniform distribution and predators from a Poisson distribution according to the following equation:

$$\text{Expected Fitness} = f * 0.9^p$$

Where f is food and p are predators. It may be possible to determine from the food and predator distributions what the fitness distribution will be, however it is beyond my knowledge how to do this.

We have seen that different decision-making mechanisms affect the evolution of an evaluation system. In my model I have assumed a fixed decision-making mechanism, however in real systems this too will have evolved, and it will have evolved in synchrony with any evaluation mechanism. This surely will have an effect on the outcome and course of the evolutionary process. An interesting question which is not further addressed in this research is what decision-making rule would evolve. There has been found some evidence that this probably is a general mechanism that is able to be applied both in a simultaneous and sequential decision-making scenario (Freidin et al., 2009). It involves an added latency for an animal to take options that are lower in value. They found that this behaviour often results in optimal foraging strategies.

Furthermore, adding learning to the decision-making process will have further unforeseen implications to the functioning and development of an evaluation system. Modelling evolution of integrated modules together is key to gaining more insight into the functioning and development of these systems. When taking learning into account we can predict when learning would be adaptive. Whether learning evolves is very dependent on environmental complexity as described in (Botero et al., 2015; Dridi & Lehmann, 2016; Dunlap & Stephens, 2016). They predict that for learning to evolve the environment needs to be variable to a certain extent over a single lifetime, otherwise fixed responses can be used which are less costly. Secondly, they predict that when information is unreliable, making the environment unpredictable, neither learning nor any fixed response may evolve. The information that has no predictive value would not be taken into account.

The next step towards the goal this report previously described, will be to add the possibility for learning to the decision-making process. Expectations will be that under circumstances of high enough predictability and high enough variability in the environment learning will evolve even if it is costly. The model should therefore also vary environments between generations. This could be in the form of having multiple food sources that are scarce at some points and abundant at others.

In my model the environment stays stable, food and predators in patches are always drawn from the same distributions. However, when looking at reliability of cue information I found that multiple information

sources are able to compensate for noise added to the information cues. This implies that learning may more readily evolve as well if multiple information sources are available in the environment and that they may compensate for how well they can predict certain outcomes. However, learning would only be of added value if the environment is variable.

As briefly addressed above, one further expansion of this model that could be interesting would be for changing the topology of the evaluation network. This can either be by adding more inputs, like information about the number of patches rejected, or could be the addition of hidden layers. A hidden layer is a layer of neurons or nodes between the input and output layer of the network, they enable non-linear transformations of the inputs (Hornik et al., 1989). With an added cost to more complex networks, it would be interesting to see how environment would influence to complexity of the network that is required or optimal. Would in the consecutive assessment task for example agents benefit from having hidden layers added to their network?

One of the limitations in this model is that with the methods implemented it is impossible to investigate completely unreliable cues. When I tried to give agents uninformative cues, they received information about their environment through the uniform distribution with adjusted means. With uninformative cues drawn from normal distribution, the average is still the same as the real value of food or predators in the patch. Because it is normally distributed, they are still likely to receive correct information.

CONCLUSIONS

Choosing the best option out of several options while weighing them before deciding, is not a hard problem to solve as in the case of simultaneous assessment. A wide range of simple neural networks reach optimal performance or very close to it. The fitness landscape was found to be flat. Consecutive assessment is a much harder problem to solve for a simple network that just has an input and output layer. An evolving bias is really needed to give some information on which patches are below average. Having a bias, the simple networks were able to greatly increase the performance of the agents and their fitness. Here the fitness landscape was found to have a narrower ridge on which the fitness peak was reached. We can therefore conclude that the simple networks alone are able to robustly handle the decision-making tasks discussed in this report and that the decision-making mechanism affect the fitness landscape and hence the evolutionary outcome.

I have shown that an agent receiving multiple information cues about the same patch property is able to compensate for multiple unreliable cues. Just like this may happen in real biological system, for example colour and smell of food. The cues may have different reliabilities for predicting the actual information they are interested in, e.g. the caloric content. If the agent was given multiple cues for the same piece of information, depending on the reliability of the cues it was advantageous to have multiple information sources as they could compensate for the lack of reliability of both. Two unreliable cues will both be used. However, if there is a cue present that conveys 100% accurate information, any other cue will be ignored as there is no reason to compensate.

LITERATURE

- Agrawal, A. F., & Whitlock, M. C. (2012). Mutation Load: The Fitness of Individuals in Populations Where Deleterious Alleles Are Abundant. *Annual Review of Ecology, Evolution, and Systematics*, 43(1), 115–135. <https://doi.org/10.1146/annurev-ecolsys-110411-160257>
- Bach, D. R., & Dayan, P. (2018). Opinion: Algorithms for survival: A comparative perspective on emotions. *Nature Reviews Neuroscience*, 18, 311–319.
- Botero, C. A., Weissing, F. J., Wright, J., & Rubenstein, D. R. (2015). Evolutionary tipping points in the capacity to adapt to environmental change. *Proceedings of the National Academy of Sciences of the United States of America*, 112(1), 184–189. <https://doi.org/10.1073/pnas.1408589111>
- Cain, C. K., & LeDoux, J. E. (2007). Escape from fear: A detailed behavioral analysis of two atypical responses reinforced by CS termination. *Journal of Experimental Psychology*. 33(4), 451–463. <https://doi.org/10.1037/0097-7403.33.4.451>
- DeAngelis, D. L., & Diaz, S. G. (2019). Decision-making in agent-based modeling: A current review and future prospectus. *Frontiers in Ecology and Evolution*, 6, 1–15. <https://doi.org/10.3389/fevo.2018.00237>
- Dridi, S., & Lehmann, L. (2016). Environmental complexity favors the evolution of learning. *Behavioral Ecology*, 27(3), 842–850. <https://doi.org/10.1093/beheco/arv184>
- Dunlap, A. S., & Stephens, D. W. (2016). Reliability, uncertainty, and costs in the evolution of animal learning. *Current Opinion in Behavioral Sciences*, 12, 73–79. <https://doi.org/10.1016/j.cobeha.2016.09.010>
- Freidin, E., Aw, J., & Kacelnik, A. (2009). Sequential and simultaneous choices: Testing the diet selection and sequential choice models. *Behavioural Processes*, 80(3), 218–223. <https://doi.org/10.1016/j.beproc.2008.12.001>
- Giske, J., Eliassen, S., Fiksen, Ø., Jakobsen, P. J., Aksnes, D. L., Jørgensen, C., & Mangel, M. (2013). Effects of the emotion system on adaptive behavior. *American Naturalist*, 182(6), 689–703. <https://doi.org/10.1086/673533>
- Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366. [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- Kuchta, M. (2017). Iterated full information secretary problem. *Mathematical Methods of Operations Research*, 86(2), 277–292. <https://doi.org/10.1007/s00186-017-0594-0>
- McNamara, J. M., & Houston, A. I. (2009). Integrating function and mechanism. *Trends in Ecology and Evolution*, 24(12), 670–675. <https://doi.org/10.1016/j.tree.2009.05.011>
- Rolls, E. T. (2013). *Emotion and Decision-making Explained*. Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780199659890.001.0001>
- Vasconcelos, M., Machado, A., & Pandeirada, J. N. S. (2018). Ultimate explanations and suboptimal choice. *Behavioural Processes*, 152, 63–72. <https://doi.org/10.1016/j.beproc.2018.03.023>

APPENDIX

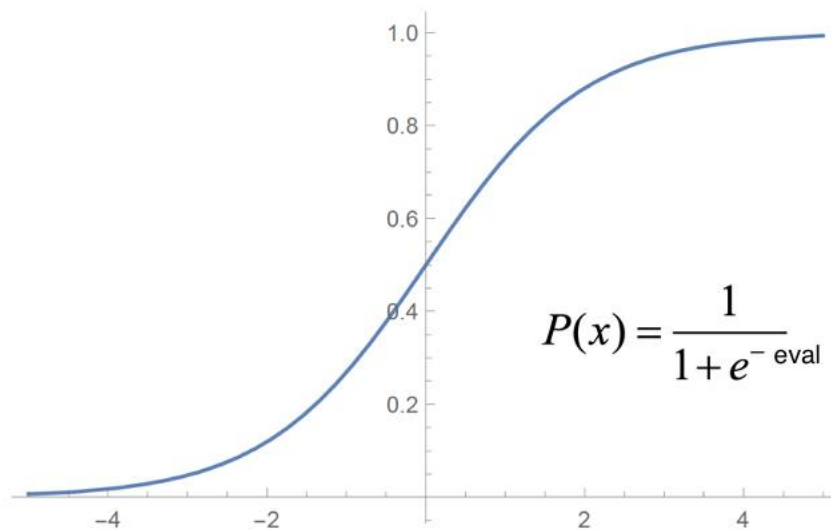


Figure A1, sigmoidal curve used to transform the evaluation value to a probability from 0 to 1.0

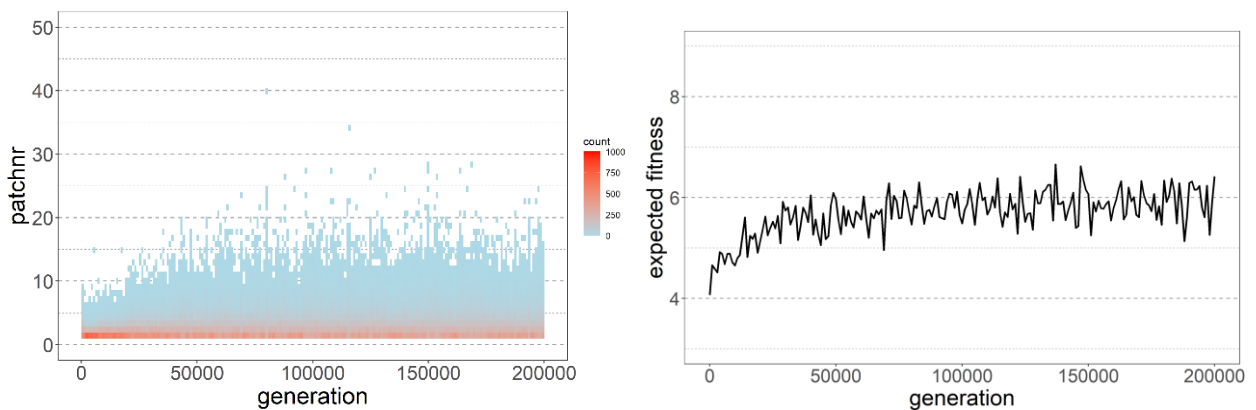
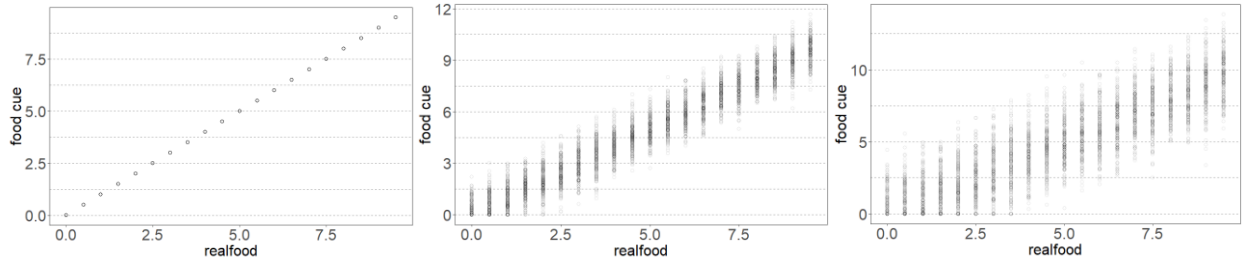
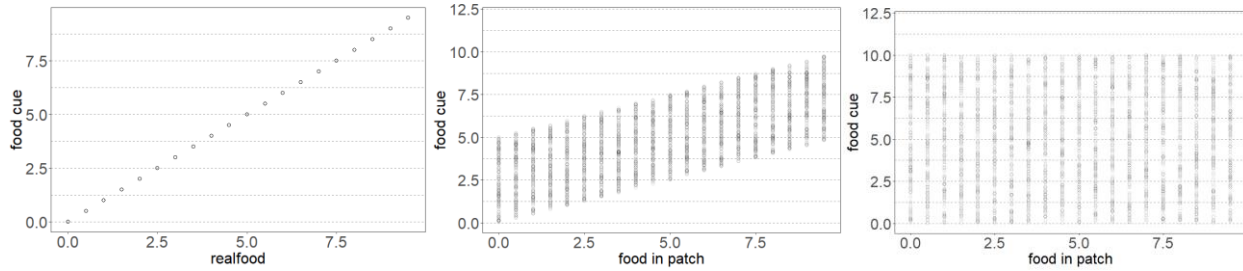


Figure A2, evolution in a consecutive choice scenario, agents have a network without bias present. There is an added cost to the rejection of a patch of 0.01. The best a patch can be is 10 in the case of 10 food and no predators. The left panel shows the number of the patch which was chosen and indicates the choosiness of the agents. When agents on average choose a larger patch number, it means they have rejected the ones before. the right panel is the average expected fitness of the agents in the population.

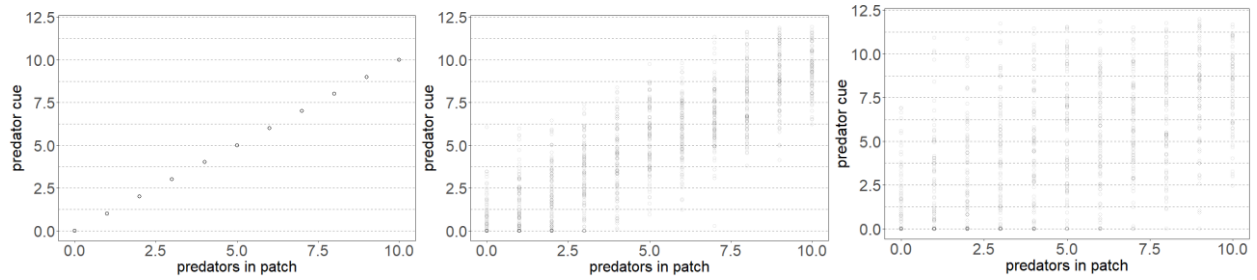
Food Normal



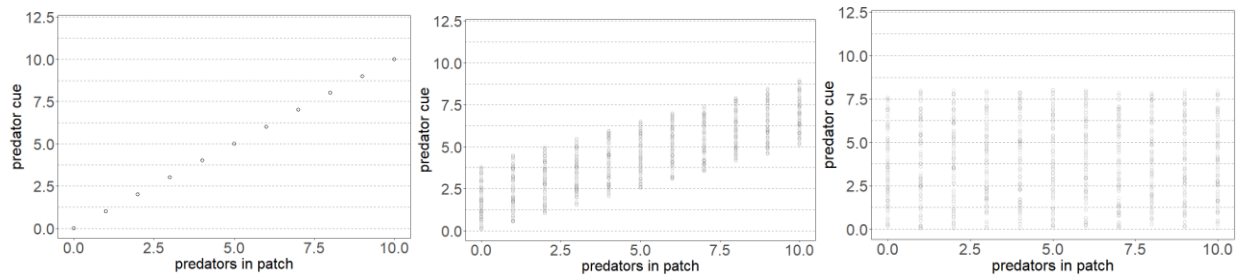
Food Uniform



Predators Normal



Predators Uniform



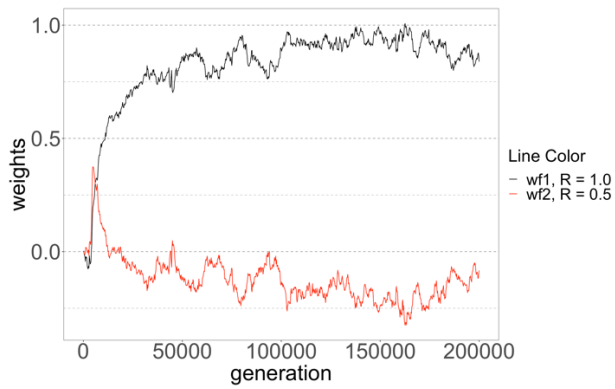
$R = 1.0$

$R = 0.5$

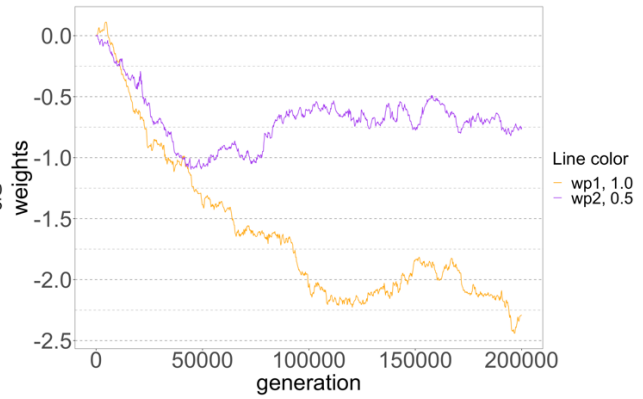
$R = 0.0$

Figure A3, graphs of cue distributions for different reliabilities drawn from normal distributions, from left to right 1.0, 0.5 and 0.0 reliability. On the x-axis the actual value of the food in a patch and on the y-axis the distribution of the information cue value.

A



B



C

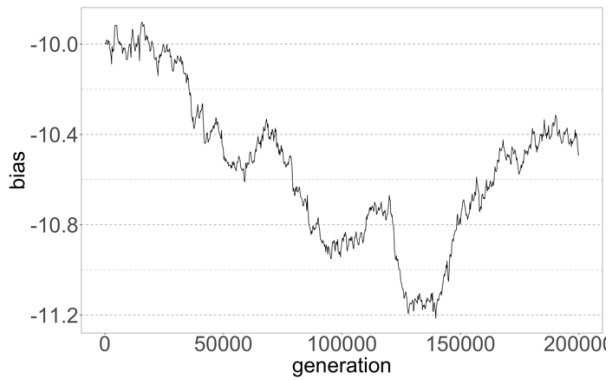


Figure A4 weights from simulation with reliability combinations of $R=1.0$ and $R=0.5$, bias evolves too and is initialized at -10. Cues are drawn from uniform distribution.