A Dynamical Systems Approach to Optimal Foraging

Siddharth Chaturvedi[1*], Ahmed El-Gazzar[1], Marcel van Gerven[1],

1 Department of Artificial Intelligence, Donders Institute for Brain, Cognition and behaviour, Radboud University, Nijmegen, the Netherlands.

***** siddharth.chaturvedi@donders.ru.nl (SC)

Abstract

Foraging for resources in an environment is a fundamental activity that must be addressed by any biological agent. Thus, modelling this phenomenon in simulations can enhance our understanding of the characteristics of natural intelligence. In this work, we present a novel approach to modelling this phenomenon in silico. We achieve this by using a continuous coupled dynamical system for modelling the system. The dynamical system is composed of three differential equations, representing the position of the agent, the agent's control policy, and the environmental resource dynamics. Crucially, the control policy is implemented as a neural differential equation which allows the control policy to adapt in order to solve the foraging task. Using this setup, we show that when these dynamics are coupled and the controller parameters are optimized to maximize the rate of reward collected, adaptive foraging emerges in the agent. We further show that the internal dynamics of the controller, as a surrogate brain model, closely resemble the dynamics of the evidence accumulation mechanism, which may be used by certain neurons of the dorsal anterior cingulate cortex region in non-human primates, for deciding when to migrate from one patch to another. Finally, we show that by modulating the resource growth rates of the environment, the emergent behaviour of the artificial agent agrees with the predictions of the optimal foraging theory.

Author summary

Intelligence itself is a phenomenon that arises as a result of the interactions of an agent's dynamics with the environment's dynamics under the assumption that the agent seeks optimization of certain objective. Modelling the environment's and agent's dynamics as a single coupled dynamical system can shed light on patterns of intelligence that unfold in time. In this report, we aim to provide a minimal in-silico framework that models the main components involved in natural phenomenon, like optimal foraging, as a coupled dynamical system. Interestingly, we observe similarities between the surrogate brain dynamics of the artificial agent with the evidence accumulation mechanism that can be responsible for decision-making in certain non-human primates performing a similar foraging task. We also observe similarities between trends prescribed by theories

prevalent in behavioural ecology such as the optimal foraging theory and those shown by the artificial agent. Such similarities can help us design artificially intelligent systems that are more explainable and predictable. Furthermore, they can also increase the confidence of researchers to consider using such artificial agent models as a simulation tool to make predictions and test hypotheses about aspects of natural intelligence.

1 Introduction

Natural intelligence is defined as the innate problem-solving abilities of biological agents. It can either be studied by attempting to reverse engineer its functionality using (neurobehavioural) observational data in behaving biological agents or by designing and simulating adaptive artificial agents in complex environments to capture mechanisms of natural intelligence [1]. In this work, we are motivated by Braitenberg's law of uphill analysis and downhill invention which states that it is easier to understand complex systems by reconstructing them from basic components rather than by estimating them from observations [2]. Thus, we seek to explore the latter option of modelling traits of natural intelligence from basic building blocks, as an approach to foster their understanding. We focus on one of the most prevalent traits of natural intelligence, that is, its ability to optimize the act of foraging for resources from its environment in order to satisfy the organism's internal needs.

Foraging refers to the act of searching and gathering resources from the environment. It is ubiquitous and ancient in the sense that organisms ranging from unicellular organisms like bacteria [3, 4] to organisms having complex social structures like primates [5–7] display the urge to forage optimally. Various models depicting different aspects of foraging have been proposed under the umbrella of foraging theory [8]. Foraging models have uniquely predicted certain effects and phenomena regarding natural intelligence such as the effect of a forager's internal energy budget on its sensitivity to the variance in reward or describing the environmental conditions under which a forager should or should not engage in exploration [8].

Patch foraging refers to one such model wherein a forager needs to decide when to quit foraging from a particular patch of resources and migrate to a different one. The caveat lies in the fact that often the value of resources in these patches is modeled as an

exponentially decaying function of time [9]. This aspect of ceasing to consume the current resource for better prospects in a patch foraging task captures a large set of decision-making problems that natural agents face on a regular basis. Examples include, foraging for information instead of food or the act of quitting a problem-solving technique midway in order to explore a new one [10]. It has been shown that foraging activities employ multiple neural systems in the brain [11] to engage a variety of cognitive functions like learning of resource distributions across spatio-temporal scales and route planning [12].

Many optimal solutions and concrete behavioural characteristics of natural agents excelling in foraging tasks already exist and are studied extensively as a subset of optimal foraging theory [8, 13]. Particularly, in the case of patch foraging, the marginal-value theorem (MVT) dictates the optimal policy under some constraints and assumptions [9]. In brief, it states that an agent should leave a patch when the instantaneous rate of return from the patch falls below the long-term average rate of return of the environment. Many animals have been shown to comply with MVT in their natural settings [14–19]. However, there has been some push-back against MVT as it has been observed that temporal-discounting of the resource value promotes overstaying of an animal in the patch [20–22]. Furthermore, there exists some ambiguity in the value of the time horizon for which the average rate of reward of the environment is calculated [8].

Traditionally, patch foraging models have been designed and simulated using fixed decision rules and constraints, such as fixed travel-time between patches [9, 23, 24]. Work has also been done in formulating a normative theory of patch foraging decisions by posing it as a statistical inference problem [25]. More recent work has focused on using reinforcement learning (RL) to learn adaptive foraging behaviour [20, 26–30]. Hebbian learning has also been used to uncover an RL-like controller that displays adaptive foraging behaviour [31]. Work has also been done towards learning parameters of a general neural substrate based on Hebbian learning that forages optimally [32].

Our focus is on modelling how adaptive foraging emerges in relation to environmental dynamics. To this end, we use a dynamical systems approach where the agent and the environment are modeled using coupled differential equations. The motivation behind modelling the brain-environment loop as a coupled continuous

dynamical system is found in the fact that, firstly, many studies of neurons recorded from the cortex reveal the existence of complex temporal dynamics in their firing [33]. Secondly, these dynamics are heavily influenced by the dynamics of the environment in which they are embodied [34, 35]. Interpreting the brain of the agent as a dynamical system whose states evolve across time is the topic of dynamical systems theory in neuroscience [36, 37].

To model the agent's brain (controller) dynamics, we employ a modern interpretation of a Wilson-Cowan model [33] traditionally used in neuroscience to model firing rates of neural populations. The dynamics of the resources growing in the environment are modelled using a Lotka-Volterra (LV) model [38]. The LV model is parameterized by a growth rate and a decay rate which together dictate the carrying capacity of the population. Finally, the agent's locomotion in the environment is defined by a double-integrator (DI) model, describing the equations of motion of a particle in two-dimensional space [39]. Crucially, we assume that the controller model in the coupled model has modifiable parameters which can be used to change the dynamics. Thus it can be interpreted as a neural differential equation (NDE) [40]. Defining the optimal foraging objective as the maximisation of resource consumption while minimising energy expenditure, we may estimate model parameters by optimising the optimal foraging objective using automatic differentiation [41].

In summary, through this work, we introduce a general and minimal framework to model and analyse the emergence of adaptive foraging behaviour in artificial agents, by posing the agent and the environment as a coupled dynamical system. Then, we use the framework to show that phenomena of interest emerge from a very simple coupling and optimization of the dynamics. This is reminiscent of neuronal dynamics observed in single-cell recordings from non-human primates while patch-foraging in a computerised task [42]. We also observe a shift in patch-residing times of the agent when the growth rate of the resources in the environment is modulated. This observation aligns with the predictions of the optimal foraging theory. That is, a forager tends to overstay in a patch when the environment becomes poorer in terms of resources [8, 9].

This paper is structured as follows. Section 2 describes the formulation of the coupled dynamical system and the process used for its optimization. Section 3 presents all the experiments performed using this setup. Finally, Section 4 discusses empirical

findings and lays down directions for future work.

2 Methods

2.1 Modelling the dynamics

Our setup is a coupled dynamical system consisting of three components: A position model, a control model, and a resource model. The position and control model together comprise the agent whereas the resource model comprises the environment. Each component is modeled as a dynamical system, and their interactions are captured by a set of coupled differential equations. Specifically, in our approach, we model everything in terms of forced ordinary differential equations (ODEs) of the form

$$\dot{x}(t) = f_\theta(x(t), u(t)) \tag{1}$$

where $x(t)$ is the state of the system at time t, $\dot{x}(t)$ is its time derivative, $u(t)$ is the forcing function, f is the state equation that models the system dynamics and θ are (optional) free parameters. For conciseness, we often omit the time index from our notation. Based on this general notation, we specify the different model components.

2.1.1 Position model

The agent is modelled as a particle moving on the surface of a torus. The toroidal environment is chosen to remove spatial boundary conditions and facilitate a continuous state space for the agent's position. This serves as a basic representation of an agent moving in 2D space. An overview of the setup can be seen in Fig 1. The motion of the agent is modelled using a deterministic double integrator, which implies that the dynamics of the position model f_p is given by

$$f_p(x_p, u_p) = Ax_p + Bu_p \tag{2}$$

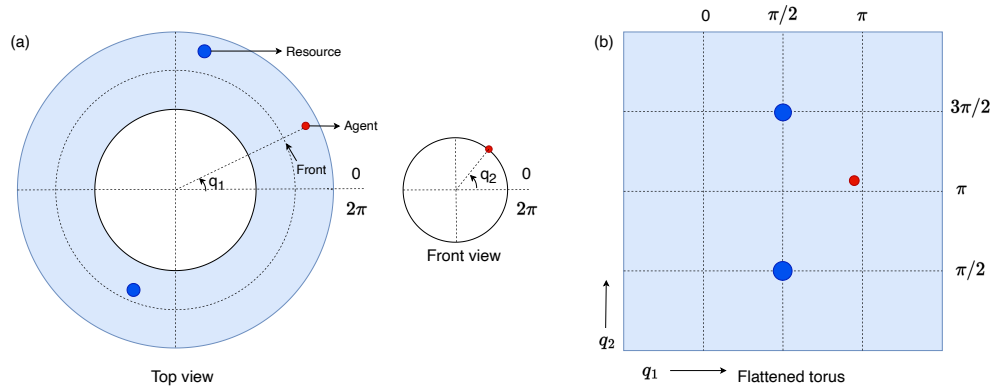


Fig 1. A toroidal environment with two degrees of freedom. (a) The top view and the front view of the environment. The agent is depicted with a red disc and the resources with blue discs. The resources are growing at random locations. (b) A flattened view of the torus with the resources now growing at $(\pi/2, \pi/2)$ and at $(\pi/2, 3\pi/2)$.

where $x_p(t) = [q_a(t), \dot{q}_a(t)]^\top$ with $q_a(t) \in \mathbb{R}^2$ being the position of the agent in 2D space. The system matrices are given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\gamma & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\gamma \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{3}$$

where γ is a damping factor. We further assume that the control vector $u_p(t) \in \mathbb{R}^2$ is a linear readout of the neuronal firing rates $r(t)$ given by

$$u_p(t) = \tanh(Dr(t))\,. \tag{4}$$

Here $D \in \mathbb{R}^{2\times n}$ is a readout matrix, mapping the firing rates of n neurons to the accelerations. The tanh function ensures that the magnitudes of the accelerations along both the directions remain bounded. Firing rate dynamics are described in Section 2.1.3.

2.1.2 Resource model

We model the distribution of resources by choosing $m = 2$ fixed locations on the surface of torus which are capable of growing resources as shown in Fig 1(b). These locations are given by $(\pi/2, \pi/2)$ and $(\pi/2, 3\pi/2)$. The information about the location of the

resources is stored in a constant vector $q_r \in \mathbb{R}^4$. Thus, in this model the dynamics only exist in terms of the value of these resources $x_r(t) = [s_1(t), s_2(t)]^\top$, where $s_i(t) \in \mathbb{R}$ is the value of the i-th resource. The resource value dynamics f_r are based on a Lotka-Volterra (LV) style equation, such that

$$f_r(x_r, u_r) = \epsilon \circ x_r - \sigma \circ x_r^2 - \beta \circ x_r \circ u_r \tag{5}$$

where $\epsilon \in \mathbb{R}^+$ is the constant growth rate, $\sigma \in \mathbb{R}^+$ models the constant decay rate and $\beta \in \mathbb{R}^+$ models the constant consumption rate of the resources. The operator $\circ$ represents an element-wise multiplication. The control vector $u_r(t) = w(q_a(t), q_r), \in \mathbb{R}^2$ represents the distance between the agent's position $q_a(t)$ and the (fixed) resource locations q_r. The distance function w is computed based on the sines and cosines of $q_a(t)$ and q_r so that the continuous nature of the agent's position with respect to the resources remains intact and boundary conditions are not imposed. The structure of w is detailed in Eq 8 of the Supporting Information.

2.1.3 Control model

The controller model represents a surrogate brain model which determines how the agent controls its position based on its observations. The controller dynamics f_c are modeled using a modern interpretation of a Wilson-Cowan model [33]

$$f_c(x_c, u_c) = \alpha \circ (-x_c + Jr(x_c) + b + Eu_c) \tag{6}$$

where the system's state $x_c(t) = [z_1(t), z_2(t), \ldots, z_n(t)]^\top \in \mathbb{R}^n$ represents the summed and filtered synaptic currents z_i of n neurons and $r(x_c(t)) = \tanh(x_c(t))$ represents their firing rates. The firing rates are also used as the control input to the position model in Eq 4. The recurrent weights of the CTRNN are given by $J \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ is a bias term and $E \in \mathbb{R}^{n \times p}$ are the input weights. The CTRNN dynamics are further determined by $\alpha = [\tau_1^{-1}, \tau_2^{-1}, \ldots, \tau_n^{-1}]^\top$, where $\tau_i \in \mathbb{R}^+$ is the time constant for the ith neuron.

In this case, the control vector $u_c \in \mathbb{R}^p$ is a vector of p observations sensed by the agent. These observations consist of the sines and cosines of the agent's position, the

velocity of the agent in the 2D space, the value of the resources, and the net amount of resources consumed by the agent at time t as detailed in the following section. Its exact form is described in Eq 11 of the Supporting Information. The initial value of the state vector is obtained by passing the control through a linear readout matrix R such that $x_c(0) = Ru_c(0)$. Further, all the learnable parameters of the position and control models form a subset of the learnable parameters θ such that $\theta = \{D, J, b, E, \alpha, R\}$.

2.2 Optimal foraging objective

We aim to optimize the agent in order to evolve it into a rate-maximizing forager. Thus, the aim of the agent is to maximize the average amount of resources it consumes over time and minimize the amount of control exerted. That is, the goal is to learn a policy parameterized by θ such that

$$J(\theta) = \mathbb{E}_\theta \left[\int_0^T h(t)dt \right] \tag{7}$$

with $h(t) = ||e(t)||_1 - ||u_p(t)||_2^2$ is maximized. Here, $e(t) = \beta \circ x_r(t) \circ u_r(t)$, represents resource consumption, as presented in Eq 5. We also impose a cost on movement by penalizing the square of control input u_p of the position model in Eq 2.

2.3 Learning procedure

Having access to the deterministic dynamics of the system, we use a simple learning algorithm based on the truncated backpropagation-through-time (t-BPTT) procedure [43] to find the optimal values of θ. Firstly, for each learning trial, we sample 64 initial points of the coupled dynamical system (see Supporting Information for details on system initialization). Following that we simulate the trajectories corresponding to the sampled initial points. Each trajectory is composed of 1000 time steps wherein the initial step size is set to $0.04s$. Thus, each trajectory represents the simulation of the system for $40s$. For computing the gradients according to the t-BPTT algorithm, the truncated interval is set to 200 time steps per trajectory. The value of θ is updated after every learning trial by averaging the gradients collected over all the trajectories sampled in that learning trial with respect to the objective function defined in Eq 7.

2.4 Simulation details

The model was written using the Jax library, which utilizes just-in-time compilation coupled with an accelerated-linear-algebra compiler to implement computations in parallel [44]. The Diffrax [45] and Equinox [46] libraries were used to model the differential equations and solve them respectively. More specifically, the Tsit5 ODE solver [47] is used for numerical integration of the system equations. The solver further uses a proportional-integral-derivative controller with relative and absolute tolerances of 10^{-3} and 10^{-6} respectively, to control the step size of the simulations. The Optax library [48] was used for optimization during learning. We used an Adam optimizer to calculate the gradients and update the learnable parameters [49]. The learning rate for the optimizer was set to 3×10^{-4} and the calculated gradients were clipped between the values of -1.0 and 1.0. All simulations were run on an Apple M1 CPU.

3 Results

3.1 Emergence of foraging

After optimizing the model's learnable parameters θ, the optimal policy reflects the following behaviour qualitatively. Firstly, after being spawned at a random location, the agent migrates to a patch where a resource grows. Then, the agent demonstrates position control around the patch and depletes the resource value in it to a particular level. Finally, without completely depleting the resource value, the agent moves on towards the other patch. The agent then keeps repeating this behaviour by oscillating between the two patches. Fig 2(a) represents this policy. Fig 2(b) shows the increase in the foraging objective while training, demonstrating effective learning of the task. Fig 2(c) shows behaviour of the agent at various stages of training, starting from a random agent to an optimized agent. The degree of optimization is reflected in the amount of resources gathered by the agent at various stages of training. It can be seen that the agent learns the act of migration between resource patches rather quickly. We observe that the bulk of training time is used up to optimize the patch residing time.

Since in the setup, the dynamics of the resource values in the future depends on the agent's current patch residing time, the calculation of optimal patch residing time using

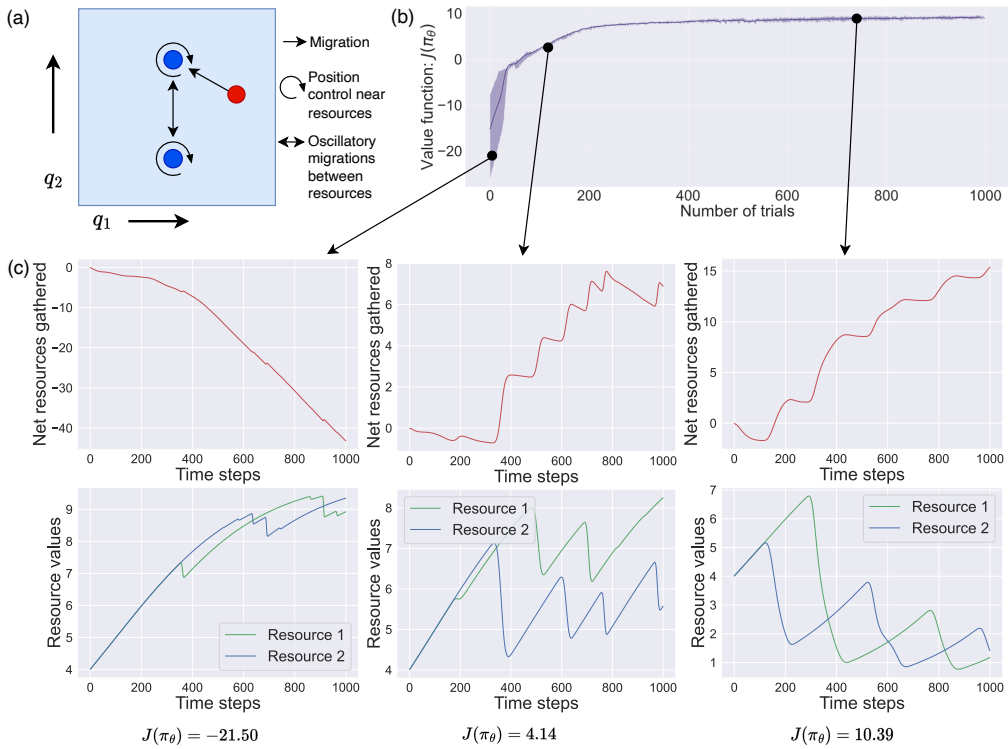


Fig 2. Results of learning. (a) Learned adaptive foraging behaviour. (b) Loss curve representing the objective function in Eq 7. (c) Net amount of resources collected by the agent and the resource values at various stages of training for a sampled trial.

MVT is not a straightforward task [8]. The reason being that overstaying in a patch can translate to more time for the other patch to regenerate. On the other hand, staying in a patch for too long can lead to the loss of opportunity of gaining more resource value at a faster rate. This trade-off may lead to a general overstaying of the agent in a patch with respect to the optimal time of leaving as predicted by the MVT. As seen in Fig 2(c), in general, for an optimal agent, the patch residing times increase with each patch visit as the environment becomes poorer in terms of the resource values.

3.2 Evidence accumulation

Neural recordings obtained from the dorsal anterior cingulate cortex (dACC) of non-human primates that were tasked to carry out a computerized patch foraging task, suggest that the underlying computations used by some neurons in the area, for deciding when to leave a patch, may represent an evidence accumulation mechanism [42]. This mechanism generally employs the use of a decision variable, where

the value of the variable represents the amount of evidence in support of a particular decision. This value accumulates over time and once it reaches a certain threshold, the decision is made. A decision is made sooner if the rate or slope at which evidence is accumulated for it is in general greater [50, 51] as shown in Fig 3(a). Moreover, a decision can also be made sooner if the amount of evidence required for it is lesser, in other words, the magnitude of the threshold that needs to be crossed by the decision variable is lesser [52] as shown in Fig 3(b). Evidence accumulation has been nominated as a general decision-making mechanism that underlies many value-based decisions in organisms including patch foraging decisions of leaving a patch [53]. We observe a similar trend in neurons of an optimized agent in the setup.

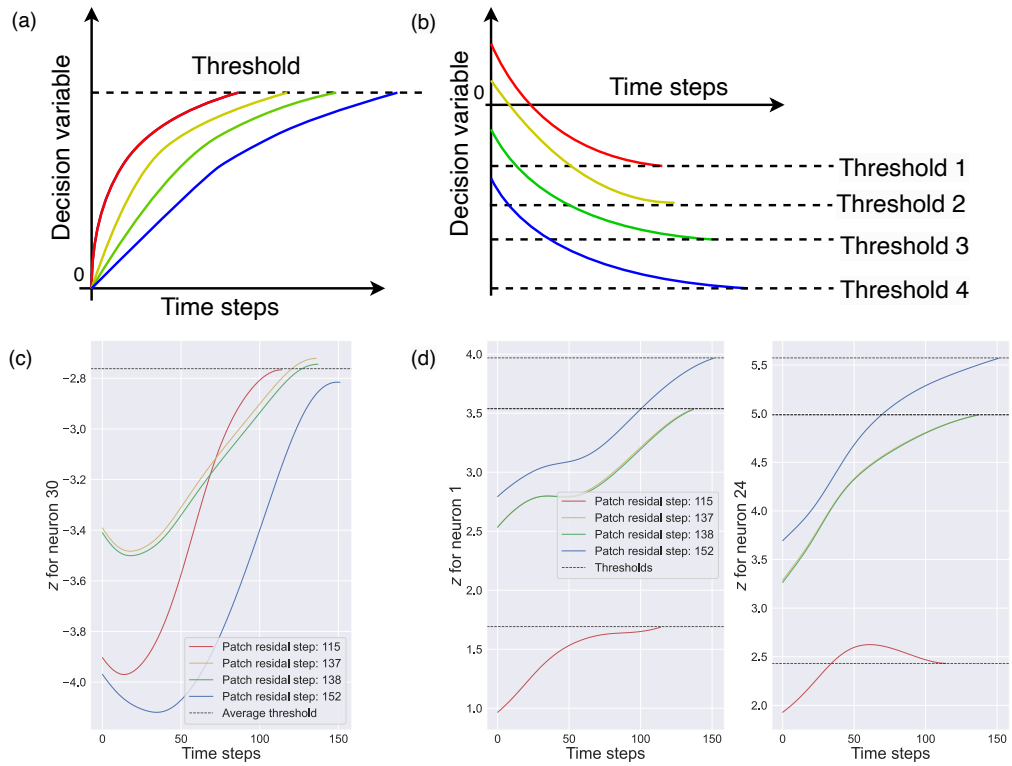


Fig 3. Evidence accumulation. (a) Evidence accumulation mechanism in an ideal case: the agent takes a particular decision, for instance, leaving a patch when the magnitude of a decision variable crosses a threshold value. In this case, the decision is made sooner if the magnitude of rate or slope of accumulation is higher as, then the decision value will cross the threshold value faster, for instance the red trajectory reaches the threshold value in least amount of time thus decision in this case will be taken the soonest. (b) In this case, the threshold beyond which a decision is taken, is itself set at a lower magnitude for the case where the decision in taken sooner, for instance, the threshold value for the red trajectory (Threshold 1) is set at the lowest magnitude thus decision for it will be taken the soonest on the contrary, in the case of the blue trajectory the magnitude of threshold value (Threshold 4) is the highest, thus the decision in this case will take the most time. (c) and (d) Value of the neuron state z in Eq 6 for three empirically selected neurons for four sampled trajectories having different initial conditions. The trajectories are cropped from the times of second patch entry to the times of second patch departure. In (c), we observe faster evidence accumulation via higher slope value i.e. the trajectory displaying the highest slope value reaches the average threshold value and exits the patch the soonest. In (d) we see the other version of evidence accumulation where the magnitude of the threshold value for the earlier patch departure is set to a lower value.

Fig 3(c) and (d) show the dynamics of some empirically selected neurons from the control model described in Section 2.1.3, for different sampled trajectories during the second patch visit. In Fig 3(c) it can be seen that the neuron accumulated evidence with a higher slope when the patch residing time is lesser. In contrast, in Fig 3(d) we observe the another evidence accumulation mechanism wherein the value of the

threshold beyond which the agent decides to leave the patch, is set at a higher magnitude for longer patch residing times. Further, when the patch residing time is very similar (137 and 138 for the yellow and green trajectories respectively in Fig 3(c) and (d)), the neuronal dynamics follow a very similar curve while respecting the trends used by the evidence accumulation mechanism to undertake a decision. This suggests that the learned control model uses a mechanism very similar to evidence accumulation.

3.3 Different growth rates

A salient conclusion of optimal foraging theory, regarding patch foraging, is that the optimal time of residing in a patch for a forager increases if the environment becomes poorer in terms of resources [8, 9]. The environment can become poor in resources if the distance between the patches or the travel time between them grows. Alternatively, if the maximum possible value of the resources in the patch becomes less, the environment can be termed poorer in terms of resources. In this experiment, we trained agents to forage in three different environments with different growth rates of the resources in the patches. This allowed us to see if the simulation results align with the OFT predictions. Environments share a common value of the decay rate while using different growth rates. This translates into different carrying capacities of the patches in each environment.

Fig 4(a) shows a comparison of the average patch residing times calculated for the first three patch visits across different trajectories in the three environments (mean of 143, 107.33, and 101.4 time-steps for growth rates of 0.8, 1.0, and 1.2 respectively). Results show that the mean patch residing time for the agent decreases as the environment becomes richer in resources. In Figs 4(b), it can be seen that the environment with higher growth rate generates more resources, which, in turn, leads to an optimal agent collecting more resources and spending less time in a patch.

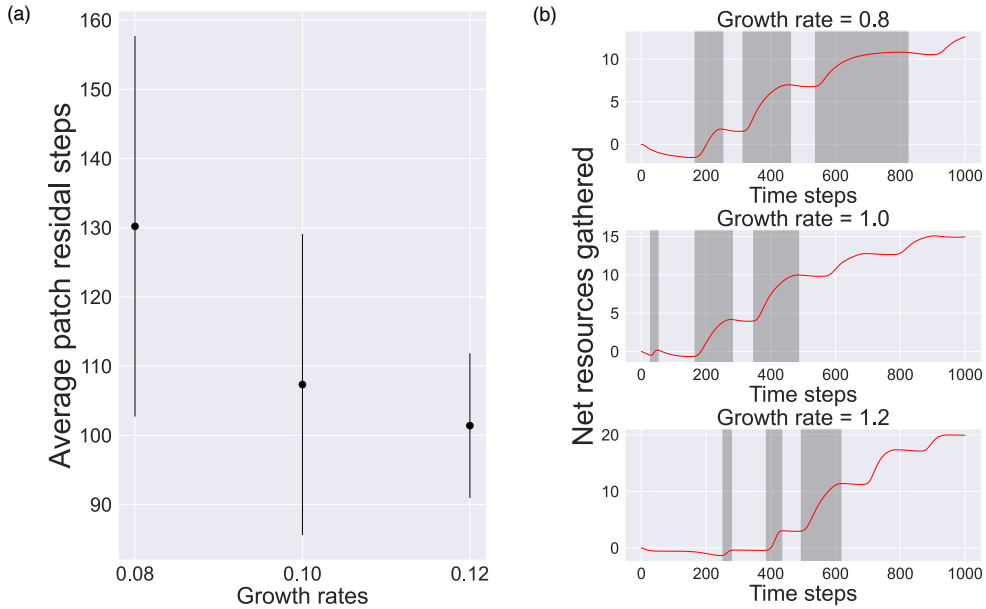


Fig 4. Comparisons of foraging behaviour in three different environments. (a) Average patch residing times for the first three patch visits for different trajectories sampled in the three environments. (b) Net amount of resources gathered by the agents during a sampled trial in each environment. The grey regions indicate the first three patch residing periods, which decreases for the sampled trial as the growth rate increases across environments (mean of 173.0, 92.33 and 65.67 time-steps for growth rate of 0.8, 1.0 and 1.2 respectively).

4 Discussion

We presented a general and minimal setup to model and simulate the dynamics involved when an adaptive agent learns to forage. The setup is general in the sense that we combine the salient aspects of the environment and agent dynamics that implement foraging at large, as one coupled dynamical system. This was done such that the continuous nature of the dynamics are preserved. The setup is minimal in the sense that, since we have the access to the complete deterministic dynamics of the system, simple supervised learning algorithms such as t-BPTT can be used to optimize the free parameters of the coupled system.

We showed that the optimized controller model, as a surrogate brain model, consists of neurons which display evidence accumulation dynamics that are similar to the dynamics observed in the dACC region of non-human primates that have learned to patch forage. Further, when the environment parameters like the growth rate of

resources were tweaked, adaptive foraging behaviour matched the predictions made by optimal foraging theory. Namely, the agent stayed in the patches for longer periods of time when the environment became poorer in terms of resources.

The advantage of our dynamical systems approach to optimal foraging is that complex emergent behaviour can be studied using a minimal setup which only requires the numerical simulation of a system of differential equations and the optimisation of free parameters using automatic differentiation. This setup facilitates hypothesis testing since assumptions about the agent and the environment translate into specific changes in the differential equations that make up the optimal foraging model.

The model presented, contains the minimal components needed for optimal foraging to emerge and lacks some of the realism that exist in nature. For instance, currently the agent receives complete and deterministic information about the environment. It will be interesting to study the change in dynamics of the agent's brain when its sensors become noisy or when it receives partial observations from the world as in the case of natural organisms [54]. We may also extend the environment in various ways. Noisy resource dynamics can be facilitated by resorting to stochastic differential equations [25]. Modelling the distribution of resources across space can be handled using partial differential equations [55]. Downsides of these extensions are the increased computational complexity involved in simulating such systems.

Another interesting future avenue would be to extend the current setup to a multi-agent setup and analyse the multi-agent strategies that agents learn in order to forage together [56]. More specifically, the analysis of competitive and collaborative nature of these strategies may uncover some interesting parallels that are also observed in nature [57, 58].

Finally, the current setup estimates free parameters using automatic differentiation. Ultimately, gradient-free approaches such as evolutionary algorithms may prove to be more convenient as they do not require differentiation through the environment. Furthermore, this would allow studying how the system of differential equations evolves structurally under environmental pressure [59].

In conclusion, with the development of a minimal model of optimal foraging, cast in terms of a system of coupled differential equations, we aim to provide researchers with a new approach to studying how complexity emerges as natural agents adapt to their

environment.

Supporting information

This section presents some of the finer modelling details and empirical choices used to model the system.

Distance function

The function $w(q_a, q_r) \in \mathbb{R}^2$ used to compute the distance metric between the agent's position q_a (at some time t) and the resource locations q_r in Section 2.1.2 is given by

$$w(q_a, q_r) = \left[e^{\frac{-l}{d(q_{r_1}, q_a)}},\, e^{\frac{-l}{d(q_{r_2}, q_a)}} \right]^\top \tag{8}$$

where l represents the steepness of the Gaussian kernel and is set to -7.0 and $d(q_{r_i}, q_a) \in \mathbb{R}^+$ is given by

$$d(q_{r_i}, q_a) = ||\sin(q^1_{r_i}) - \sin(q^1_a),\, \cos(q^1_{r_i}) - \cos(q^1_a),\, \sin(q^2_{r_i}) - \sin(q^2_a),\, \cos(q^2_{r_i}) - \cos(q^2_a)||_2\,. \tag{9}$$

Here, $q_{r_i} = (q^1_{r_i}, q^2_{r_i})$ represents the coordinates of the resources r_i along the two axes of the torus. For the two resources growing in the environment these are given by $q_{r_1} = (\pi/2, \pi/2)$ and $q_{r_2} = (\pi/2, 3\pi/2)$. Further, $q^1_a(t) \in \mathbb{R}$ and $q^2_a(t) \in \mathbb{R}$ represents the agent's position at time t along the two axes of the torus.

Observation vector

The observation vector $u_c(t) \in \mathbb{R}^p$ used in Section 2.1.3 as control input to the dynamics of the control model is a concatenation of vectors $l_a(t) \in \mathbb{R}^6$, $x_r(t) \in \mathbb{R}^2$ and a scalar $h(t) = ||e(t)||_1 - ||u_p(t)||^2_2$, which represents the net amount of resources consumed by the agent at time t, such that $p = 9$. Firstly, $l_a(t)$ is given by

$$l_a(t) = [\sin^*(q_a(t)),\, \cos^*(q_a(t)),\, \dot{q}_a(t)]^\top \tag{10}$$

where $q_a(t) \in \mathbb{R}^2$ represent the position of the agent. The $\sin^*$ and $\cos^*$ functions represent an element-wise sine and cosine of a vector passed to them. Secondly, $x_r(t) \in \mathbb{R}^2$ from Section 2.1.2 represents the value of resources in the two patches. Thus we have

$$u_c(t) = [l_a(t),\, x_r(t),\, h(t)]^\top. \tag{11}$$

System initialization

For each trajectory in a learning trial, the equations of the couple dynamical systems are initialized in the following way:

Position model: The initial values of q_a and $\dot{q}_a$ in Section 2.1.1 are sampled from a random uniform distribution, distributed between $-\pi$ and π. Thus the agent can initially be spawned at any point on the surface of the torus with angular velocities of $\pm\pi\, rad/s$ along the two axes. The value of damping factor γ is set to 0.3.

Resource model: The initial value of both the resources is set to 4.0. The value of the growth rate ϵ in Eq 5 is set to 0.1 by default. The value of decay rate σ is always set to 0.01. Thus the carrying capacity which is defined as the maximum value to which a resource can grow becomes $\frac{\epsilon}{\sigma} = 10.0$. The consumption rate β for the agent is set to 0.5.

Control model: The initial values of the membrane potential vector $x_c \in R^n$ as used in Eq 6 is set by multiplying the initial observation vector $u_c \in \mathbb{R}^p$ in Eq 11 to a matrix $R \in \mathbb{R}^{n \times p}$. Such that $x_c(0) = Ru_c(0)$. The number of neurons n is set to 40 and, as highlighted in the previous section, the number of observations p is set to nine. The values of all the learnable parameters θ are initialized using Glorot initialization [60]. In the case of the neuronal time constants as used in Eq 6, we first initialized a value $c_i \in \mathbb{R}$ using Glorot-Normal initialization for the ith neuron. Then we used the equation $\tau_i = \frac{10}{1+e^{-c_i}}$ to initialize the value of ${\tau_i}^{-1}$ which formed the elements of the α vector. This was done to maintain the value of time constants above zero for physical realism.

Code

The code used in the report can be found at
`https://github.com/i-m-iron-man/NDE_Foraging.git`.

Acknowledgments

This publication is part of the project Dutch Brain Interface Initiative (DBI^2) with project number 024.005.022 of the research programme Gravitation which is (partly) financed by the Dutch Research Council (NWO).

References

1. Van Gerven M. Computational foundations of natural intelligence. Frontiers in computational neuroscience. 2017; p. 112.
2. Braitenberg V. Vehicles: Experiments in synthetic psychology. MIT press; 1986.
3. Guo C, Tang H, Niu B, Lee CBP. A survey of bacterial foraging optimization. Neurocomputing. 2021;452:728–746.
4. Chen H, Zhu Y, Hu K, et al. Adaptive bacterial foraging optimization. In: Abstract and Applied Analysis. vol. 2011. Hindawi; 2011.
5. Blanchard TC, Hayden BY. Monkeys are more patient in a foraging task than in a standard intertemporal choice task. PloS one. 2015;10(2):e0117057.
6. Eisenreich* BR, Hayden BY, Zimmermann J. Macaques are risk-averse in a freely moving foraging task. Scientific reports. 2019;9(1):15091.
7. Heilbronner SR, Rosati AG, Stevens JR, Hare B, Hauser MD. A fruit in the hand or two in the bush? Divergent risk preferences in chimpanzees and bonobos. Biology Letters. 2008;4(3):246–249.
8. Stephens DW, Krebs JR. Foraging theory. vol. 1. Princeton university press; 1986.

9. Charnov EL. Optimal foraging, the marginal value theorem. Theoretical population biology. 1976;9(2):129–136.

10. Pirolli P, Card S. Information foraging in information access environments. In: Proceedings of the SIGCHI conference on Human factors in computing systems; 1995. p. 51–58.

11. Hills TT. Animal foraging and the evolution of goal-directed cognition. Cognitive science. 2006;30(1):3–41.

12. Calhoun AJ, Hayden BY. The foraging brain. Current Opinion in Behavioral Sciences. 2015;5:24–31.

13. Pyke G. Optimal foraging theory: an introduction. In: Encyclopedia of animal behavior. Elsevier Academic Press; 2019. p. 111–117.

14. Pacheco-Cobos L, Winterhalder B, Cuatianquiz-Lima C, Rosetti MF, Hudson R, Ross CT. Nahua mushroom gatherers use area-restricted search strategies that conform to marginal value theorem predictions. Proceedings of the National Academy of Sciences. 2019;116(21):10339–10347.

15. Vertechi P, Lottem E, Sarra D, Godinho B, Treves I, Quendera T, et al. Inference-based decisions in a hidden state foraging task: differential contributions of prefrontal cortical areas. Neuron. 2020;106(1):166–176.

16. Vasquez RA, Kacelnik A. Foraging rate versus sociality in the starling Sturnus vulgaris. Proceedings of the Royal Society of London Series B: Biological Sciences. 2000;267(1439):157–164.

17. Cowie RJ. Optimal foraging in great tits (Parus major). Nature. 1977;268(5616):137–139.

18. Krebs JR, Ryan JC, Charnov EL. Hunting by expectation or optimal foraging? A study of patch use by chickadees. Animal behaviour. 1974;22:953–IN3.

19. Waage JK. Foraging for patchily-distributed hosts by the parasitoid, Nemeritis canescens. The Journal of Animal Ecology. 1979; p. 353–371.

20. Wispinski NJ, Butcher A, Mathewson KW, Chapman CS, Botvinick MM, Pilarski PM. Adaptive patch foraging in deep reinforcement learning agents. arXiv preprint arXiv:221008085. 2022;.

21. Harhen NC, Bornstein AM. Overharvesting in human patch foraging reflects rational structure learning and adaptive planning. Proceedings of the National Academy of Sciences. 2023;120(13):e2216524120.

22. Kendall RK, Wikenheiser AM. Quitting while you're ahead: Patch foraging and temporal cognition. Behavioral Neuroscience. 2022;136(5):467.

23. Pleasants JM. Optimal foraging by nectarivores: a test of the marginal-value theorem. The American Naturalist. 1989;134(1):51–71.

24. Tang W, Bennett DA. Agent-based modeling of animal movement: a review. Geography Compass. 2010;4(7):682–700.

25. Kilpatrick ZP, Davidson JD, Hady AE. Normative theory of patch foraging decisions. arXiv preprint arXiv:200410671. 2020;.

26. Kolling N, Akam T. (Reinforcement?) Learning to forage optimally. Current opinion in neurobiology. 2017;46:162–169.

27. Goldshtein A, Handel M, Eitan O, Bonstein A, Shaler T, Collet S, et al. Reinforcement learning enables resource partitioning in foraging bats. Current Biology. 2020;30(20):4096–4102.

28. Constantino SM, Daw ND. Learning the opportunity cost of time in a patch-foraging task. Cognitive, Affective, & Behavioral Neuroscience. 2015;15:837–853.

29. Miller ML, Ringelman KM, Eadie JM, Schank JC. Time to fly: A comparison of marginal value theorem approximations in an agent-based model of foraging waterfowl. Ecological Modelling. 2017;351:77–86.

30. Morimoto J. Foraging decisions as multi-armed bandit problems: Applying reinforcement learning algorithms to foraging data. Journal of theoretical biology. 2019;467:48–56.

31. Niv Y, Joel D, Meilijson I, Ruppin E. Evolution of reinforcement learning in uncertain environments: a simple explanation for complex foraging behaviors. Adaptive Behavior. 2002;10(1):5–24.

32. Montague PR, Dayan P, Person C, Sejnowski TJ. Bee foraging in uncertain environments using predictive hebbian learning. Nature. 1995;377(6551):725–728.

33. Sussillo D. Neural circuits as computational dynamical systems. Current opinion in neurobiology. 2014;25:156–163.

34. Sale A, Berardi N, Maffei L. Environment and brain plasticity: towards an endogenous pharmacotherapy. Physiological reviews. 2014;94(1):189–234.

35. Krech D, Rosenzweig MR, Bennett EL. Environmental impoverishment, social isolation and changes in brain chemistry and anatomy. Physiology & Behavior. 1966;1(2):99–104.

36. Izhikevich EM. Dynamical systems in neuroscience. MIT press; 2007.

37. Port RF, Van Gelder T. Mind as motion: Explorations in the dynamics of cognition. MIT press; 1995.

38. Lotka AJ. Contribution to the theory of periodic reactions. The Journal of Physical Chemistry. 2002;14(3):271–274.

39. Rao VG, Bernstein DS. Naive control of the double integrator. IEEE Control Systems Magazine. 2001;21(5):86–97.

40. Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. Advances in neural information processing systems. 2018;31.

41. Baydin AG, Pearlmutter BA, Radul AA, Siskind JM, Baydin AG, Pearlmutter BA, et al. Automatic Differentiation in Machine Learning: a Survey. Journal of Machine Learning Research. 2018;18:1–43.

42. Hayden BY, Pearson JM, Platt ML. Neuronal basis of sequential foraging decisions in a patchy environment. Nature neuroscience. 2011;14(7):933–939.

43. Williams RJ, Peng J. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. Neural computation. 1990;2(4):490–501.

44. Bradbury J, Frostig R, Hawkins P, Johnson MJ, Leary C, Maclaurin D, et al.. JAX: composable transformations of Python+NumPy programs; 2018. Available from: `http://github.com/google/jax`.

45. Kidger P. On neural differential equations. arXiv preprint arXiv:220202435. 2022;.

46. Kidger P, Garcia C. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. Differentiable Programming workshop at Neural Information Processing Systems 2021. 2021;.

47. Tsitouras C. Runge–Kutta pairs of order 5 (4) satisfying only the first column simplifying assumption. Computers & Mathematics with Applications. 2011;62(2):770–775.

48. DeepMind, Babuschkin I, Baumli K, Bell A, Bhupatiraju S, Bruce J, et al.. The DeepMind JAX Ecosystem; 2020. Available from: `http://github.com/deepmind`.

49. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.

50. Gold JI, Shadlen MN. The neural basis of decision making. Annu Rev Neurosci. 2007;30:535–574.

51. Ratcliff R, Smith PL, Brown SD, McKoon G. Diffusion decision model: Current issues and history. Trends in cognitive sciences. 2016;20(4):260–281.

52. Banca P, Vestergaard MD, Rankov V, Baek K, Mitchell S, Lapa T, et al. Evidence accumulation in obsessive-compulsive disorder: the role of uncertainty and monetary reward on perceptual decision-making thresholds. Neuropsychopharmacology. 2015;40(5):1192–1202.

53. Davidson JD, El Hady A. Foraging as an evidence accumulation process. PLoS computational biology. 2019;15(7):e1007060.

54. Murphy RR. Biological and cognitive foundations of intelligent sensor fusion. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans. 1996;26(1):42–51.

55. Evans LC. Partial differential equations. vol. 19. American Mathematical Society; 2022.

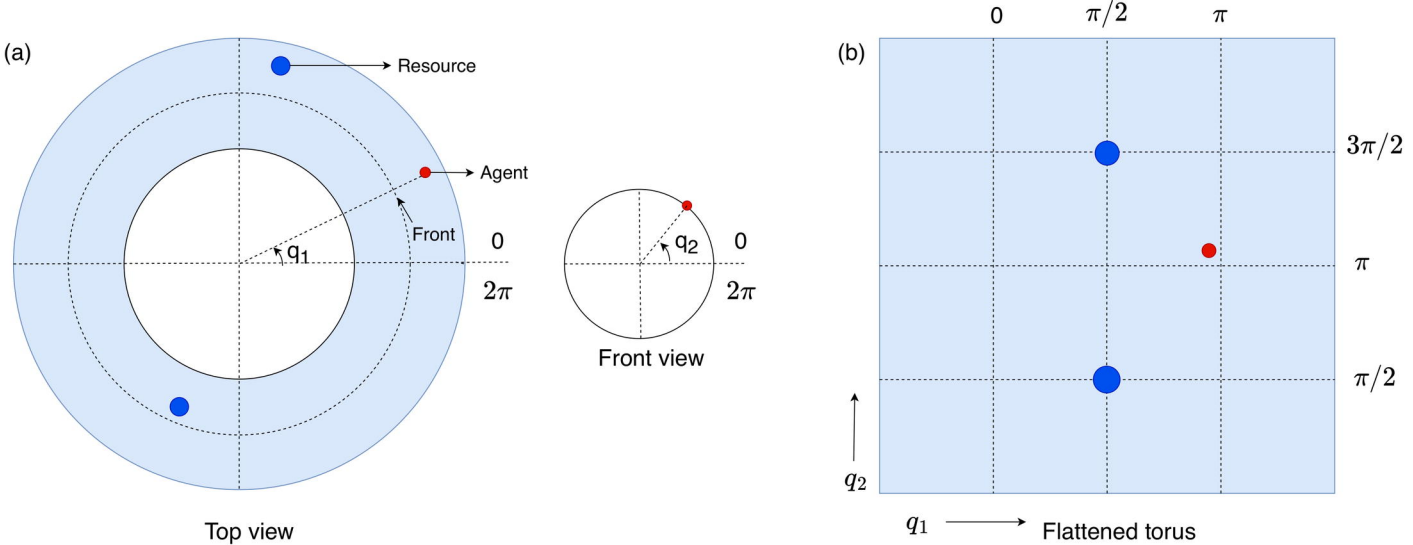
56. Zedadra O, Jouandeau N, Seridi H, Fortino G. Multi-Agent Foraging: state-of-the-art and research challenges. Complex Adaptive Systems Modeling. 2017;5(1):1–24.

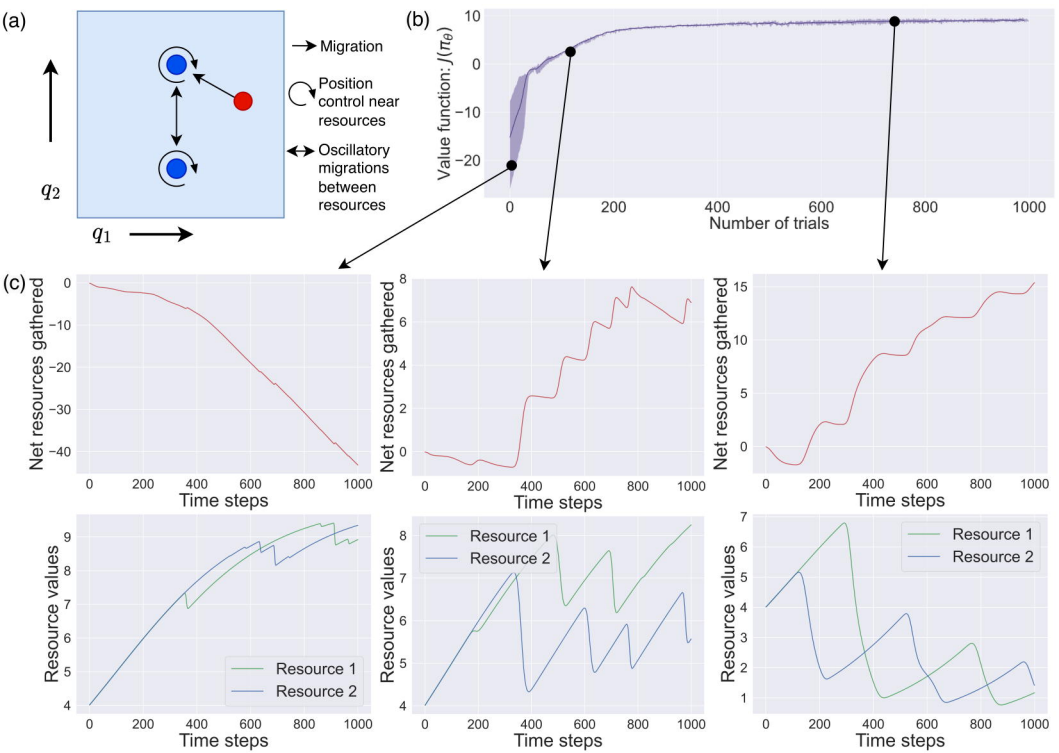
57. Zedadra O, Seridi H, Jouandeau N, Fortino G. A cooperative switching algorithm for multi-agent foraging. Engineering Applications of Artificial Intelligence. 2016;50:302–319.

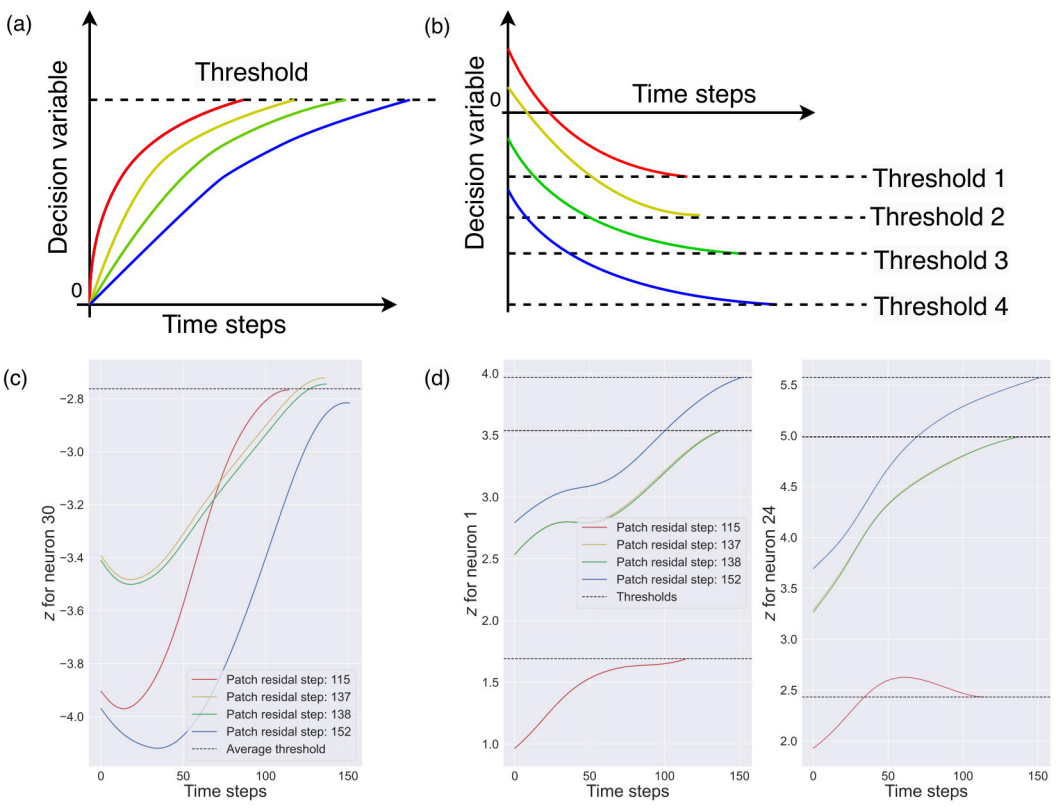
58. Mann RP. Evolution of heterogeneous perceptual limits and indifference in competitive foraging. PLoS computational biology. 2021;17(2):e1008734.

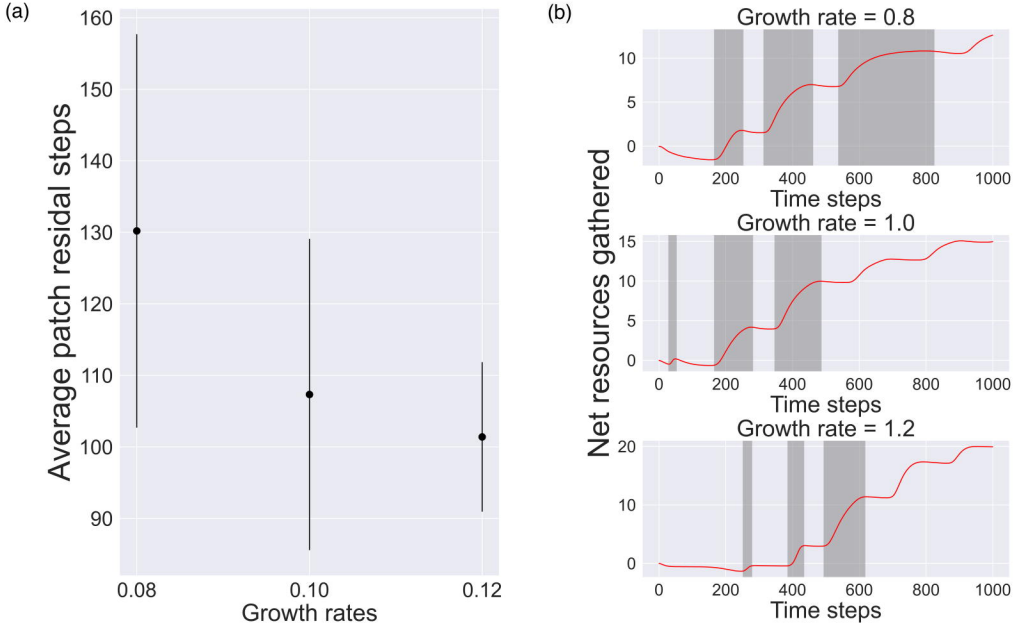
59. Sharma A, Czégel D, Lachmann M, Kempes CP, Walker SI, Cronin L. Assembly theory explains and quantifies selection and evolution. Nature. 2023;622:321–328. doi:10.1038/s41586-023-06600-9.

60. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings; 2010. p. 249–256.


(a)
Resource
Agent
Front
q_1
0
2π
Top view
q_2
0
2π
Front view
(b)
0
$\pi/2$
π
$3\pi/2$
π
$\pi/2$
q_2
q_1
Flattened torus


(a)
Migration
Position control near resources
Oscillatory migrations between resources
q_2
q_1
(b)
Value function: $J(\pi_\theta)$
Number of trials
(c)
Net resources gathered
Time steps
Resource values
Resource 1
Resource 2
$J(\pi_\theta) = -21.50$
$J(\pi_\theta) = 4.14$
$J(\pi_\theta) = 10.39$


(a)
Threshold
Decision variable
0
Time steps
(b)
Decision variable
0
Time steps
Threshold 1
Threshold 2
Threshold 3
Threshold 4
(c)
z for neuron 30
Time steps
Patch residal step: 115
Patch residal step: 137
Patch residal step: 138
Patch residal step: 152
Average threshold
(d)
z for neuron 1
z for neuron 24
Time steps
Patch residal step: 115
Patch residal step: 137
Patch residal step: 138
Patch residal step: 152
Thresholds


(a)
Average patch residal steps
160
150
140
130
120
110
100
90
0.08
0.10
0.12
Growth rates
(b)
Growth rate = 0.8
Growth rate = 1.0
Growth rate = 1.2
Net resources gathered
Time steps
0
200
400
600
800
1000