Event-Driven Simulation of Foraging Dynamics

Abstract.

We outline an event-driven simulation for the foraging model, which involves a resource that renews by logistic growth, as well as two classes of foragers—full and hungry. Full foragers reproduce at a fixed rate and are not vulnerable to mortality. However, a full forager can become hungry when resources are scarce; conversely, a hungry forager can become full when the resource is abundant. Hungry foragers do not reproduce and die at a fixed rate.

1. The Model

We assume that foragers can exist in two states—full and hungry. When a full forager F encounters a resource R, the resource is consumed with rate m so that the forager maintains its fullness. A full forager that does not encounter a resource as it wanders converts to a hungry forager H with rate equal to the product of the starvation rate σ and the density of non-resources. Full foragers also reproduce with rate λ. When a hungry forager encounters a resource, the resource is again consumed and the forager becomes full with rate ρ. A hungry forager dies with mortality rate μ, while full foragers have no mortality risk. We assume that in the absence of foragers the underlying resource undergoes logistic growth, with growth rate α and carrying capacity equal to 1.

Based on these processes and also under the assumption that the densities of full foragers, hungry foragers, and resources (also denoted by F, H, and R, respectively) are perfectly mixed, they evolve according to the rate equations:

$$\begin{aligned} \dot{F} &= \lambda F + \rho RH - \sigma(1-R)F\,, \\ \dot{H} &= \sigma(1-R)F - \rho RH - \mu H\,, \\ \dot{R} &= \alpha R(1-R) - R(mF + \rho H), \end{aligned} \tag{1}$$

where the overdot denotes time derivative. An important feature is the assumption that the carrying capacity is set equal to 1. This means that when the system is completely occupied by resources there can be no further generation of the resource. The approach described below can be generalized to carrying capacity less than 1, but we take the carrying capacity equal to 1 without loss of generality.

We now outline an event-driven algorithm that mimic these rate equations. Suppose that the system at a given time consists of N_F full foragers, N_H hungry foragers, and N_R resource units. The total number of particles is $N = N_F + N_H + N_R$. There are six elemental processes that are embodied by the rate equations (1):

(i) Reproduction: $F \to 2F$

(ii) Starvation: $F \to H$

(iii) Consumption: $F \to F$ and $R \to 0$

(iv) Death: $H \to 0$

(v) Recruitment: $H \to F$ and $R \to 0$

(vi) Growth: $R \to 2R$

For specificity, we define "consumption" as the consumption of the resource by a full forager and the maintenance of the forager in its full state, while "recruitment" is defined as the consumption of the resource by a hungry forager and conversion of the latter to the full state.

In an event-driven simulation, one of the above events is picked with the appropriate probability to be defined below, the selected event is implemented, and then the time is updated accordingly. Reading off from the steps (i)–(vi) listed above, the total rate for any event is proportional to

$$\mathcal{R} = F\big[\lambda + \sigma(1-R) + mR\big] + H\big[\mu + \rho R\big] + \alpha R(1-R)\,. \tag{2}$$

Here $F = N_F/V$, $H = N_H/V$, and $R = N_R/V$, where N_i is the total number of particles of type i, and V is the total number of lattice sites. Thus F, H, and R are the densities of each type of entity. The total rate is defined up to an overall constant, but this constant is immaterial because the time step is also proportional to this same constant.

In the most simple version of a simulation in the mean-field limit, one merely tracks the three variables N_F, N_H, and N_R and updates these numbers according to the steps outlined below. However, we want a simulation that can be directly adapted to treat foragers that diffuse on a lattice. The algorithm outlined below is based on populating a lattice with the three types of entities and updating the occupancy on the lattice sites according to the following event-driven simulation:

(i) Reproduction. With probability $\lambda F/\mathcal{R}$, create a new full forager anywhere in the system equiprobably, consistent with the absence of spatial structure.

(ii) Starvation. With probability $\sigma(1-R)F/\mathcal{R}$, pick a random full forager and change it to hungry. The factor $1-R$ is the fraction of resource-free space.

(iii) Consumption. With probability $mFR/\mathcal{R}$, pick and remove a random resource unit.

(iv) Death. With probability $\mu H/\mathcal{R}$, remove a random hungry forager.

(v) Recruitment. With probability $\rho HR/\mathcal{R}$, change a random hungry forager into a full forager and remove a random resource unit.

(vi) Growth. With probability $\alpha R(1-R)/\mathcal{R}$, create another resource unit anywhere in the system.

By explicit construction, the probabilities for the above events sum to 1. It is important to appreciate that one can perform this simulation either by populating a lattice with the three types of entities or merely be keeping track of the three variables N_F, N_H,

and N_R. A good test of the lattice-based simulation is that it should match that of simulating the above three variables.

Let's determine the change in the expected number of individuals of each type in a single event. For full foragers, the change in its number N_F is

$$\Delta N_F = \big[F\big(\lambda - \sigma(1-R)\big) + \rho H R\big]\,/\mathcal{R}\,. \tag{3a}$$

The term proportional to F describes processes in which a full forager is picked, while the term proportional to H describes processes in which a hungry forager is picked and converted to a full forager. Thus the change in the density of full foragers is

$$\Delta F = \frac{\Delta N_F}{V} = \big[F\big(\lambda - \sigma(1-R)\big) + \rho H R\big]\,/V\mathcal{R}\,. \tag{3b}$$

If we take the time step for each event to be $\Delta t = (V\mathcal{R})^{-1}$, the above reduces to the rate equation (1) for F. Thus for each microscopic update event, the time should be advanced by $\Delta t = (V\mathcal{R})^{-1}$.

In a similar fashion, the change in the expected density of hungry foragers in a single event is given by

$$\Delta H = \big[-H(\rho R + \mu) + \sigma F(1-R)\big]\,/V\mathcal{R}\,, \tag{4}$$

and the change in the density of individual resources in a single event is

$$\Delta R = \big[R\big(\alpha(1-R)\big) - (mF + \rho H)\big]\,/V\mathcal{R}\,. \tag{5}$$

The expressions for $\Delta F/\Delta t$, $\Delta H/\Delta t$, and $\Delta R/\Delta t$ reproduce the original rate equations (1) when the time step for an elemental update event is $\Delta t = (V\mathcal{R})^{-1}$. To summarize, pick an event according to the probabilities above, update the densities according (3b), (4), and (5), and then increment the time by $(V\mathcal{R})^{-1}$. A simulation that is based on the above steps should reproduce the predictions of the rate equations (1). In the three-variable simulation, keep only N_F, N_H, and N_R. In an update, pick an event according to the above probabilities, and then update N_F, N_H, and N_R and the time appropriately.

In implementing this algorithm for a lattice-based simulation, several points deserve emphasis. First, there is no constraint on the number of foragers on any site, but if we are describing a harsh environment, the parameters should be chosen so that this number is not large. Since the carrying capacity of the resource has been set to 1, N_R cannot exceed V. However, the number of resource units on any given site can exceed 1.

We now generalize the above approach to the situation where the foragers are diffusing on a lattice. Suppose that the full and hungry foragers diffuse with respective diffusion coefficients D_F and D_H. In this case, the rate equations (1) generalize to the

reaction-diffusion equations

$$\begin{aligned}
\frac{\partial F}{\partial t} &= \lambda F + \rho RH - \sigma(1-R)F + D_F\nabla^2 F\,,\\
\frac{\partial H}{\partial t} &= \sigma(1-R)F - \rho RH - \mu H + D_H\nabla^2 H\,,\\
\frac{\partial R}{\partial t} &= \alpha R(1-R) - R(mF+\rho H)\,.
\end{aligned} \tag{6}$$

Here the densities F, H, R are now functions of space and time, $F = F(\mathbf{r}, t)$ and similarly for H and R.

To construct an event-driven simulation that mimic these rate equations, we have to include two additional processes:

(vii) Diffusion of full foragers

(viii) Diffusion of hungry foragers

In an event-driven simulation, one of the eight possible events is picked with the appropriate probability (see below), the selected event is implemented, and the time is updated accordingly. The total rate for any event is

$$\begin{aligned}
\mathcal{R} =& \frac{1}{V}\sum_{\mathbf{r}} \Big\{F(\mathbf{r})\big[\lambda + \sigma(1-R(\mathbf{r})) + D_F\big] + H(\mathbf{r})\big[\mu + \rho R(\mathbf{r}) + D_H\big]\\
&\qquad + \alpha R(\mathbf{r})(1-R(\mathbf{r}'))\Big\}\\
=& F\big[\lambda + \sigma(1-R) + D_F\big] + H\big[\mu + \rho R + D_H\big] + \alpha R(1-R)\,,
\end{aligned} \tag{7}$$

in which the role of diffusion is now included. Notice also the presence of a two-particle correlation in the term $R(\mathbf{r})(1-R(\mathbf{r}'))$, where $\mathbf{r}'$ is a neighbor of $\mathbf{r}$. Thus going to the second line of the above equation is not quite kosher, but below I give a prescription to deal with this issue in a way that avoids using a correlation function.

The steps of an event-driven simulation are:

(i) Reproduction. With probability $\lambda F/\mathcal{R}$, pick a full forager at random and let it reproduce. The offspring should be placed close to the parent—either on the same site or on a neighboring site.

(ii) Starvation. With probability $\sigma(1-R)F/\mathcal{R}$, pick a random full forager that is on a resource-free site and change the forager from full to hungry. Note the two-particle nature of this rule: we need both the forager type and the resource state on a given site. To avoid keeping track of sites that are simultaneously resource free and occupied by a full forager, a simpler alternative is:

(ii') Starvation'. With probability $\sigma F/\mathcal{R}$, pick a random full forager. If the site is resource free, change the forager from full to hungry. If not, nothing happens.

(iii) Consumption. With probability $mFR/\mathcal{R}$, pick a random on a site that is also by a full forager and remove the resource. To avoid tracking sites that simultaneously contain a resource and a full forager, perform the following alternative:

(iii′) Consumption′. With probability $R/\mathcal{R}$, pick a random resource. If the site also contains a forager, remove the resource with probability m. Note the implicit assumption that the density of foragers is not large because the rate of resource removal does not increase if multiple foragers occupy the site. If the full forager density is large then the following works:

(iii″) Consumption″. With probability $mF/\mathcal{R}$, pick a random full forager. If the site also contains a resource, remove the resource. This prescription works if the forager density is small or large (as long as the resource density is less than 1) so it's best to use this last alternative.

(iv) Death. With probability $\mu H/\mathcal{R}$, pick and remove a random hungry forager.

(v) Recruitment. With probability $\rho HR/\mathcal{R}$, pick a random hungry forager on a site that also contains a resource and change the forager from hungry to full and remove the resource. This process also involves a two-body reaction and a simpler way to implement this two-body feature is:

(v′) Recruitment′. With probability $\rho H/\mathcal{R}$, pick a random hungry forager. If the site contains a resource, change the forager from hungry to full and remove the resource.

(vi) Growth. With probability $\alpha R(1-R)/\mathcal{R}$, pick a random neighboring pair of sites in which one contains a resource and the neighbor does not. Create a resource on the site without the resource. A computationally simpler alternative is:

(vi′) Growth′. With probability $\alpha R/\mathcal{R}$, pick a random resource unit. A new resource unit is created on one of the neighboring sites with probability $1-m/z$, where z is the number of nearest-neighbors and m is the number of these neighbors that are occupied by resource. If the local neighborhood is devoid of resources, growth necessarily occurs at one of these neighbors, while if the local neighborhood is full of resources, there is no growth. This rule ensures that the growth rate is proportional to $(1-R)_{\rm local}$.

(vii) Diffusion. With probability $D_F\,F/\mathcal{R}$, pick a random full forager and move it to a neighboring site.

(viii) Diffusion. With probability $D_H\,H/\mathcal{R}$, pick a random hungry forager and move it to a neighboring site.

By explicit construction, the probabilities for the above events sum to 1. Note also that the primed steps are computationally simpler than the corresponding unprimed steps, but have the possibility of a null step; this introduces some inefficiency. A true event-driven simulation avoids all null steps, but perhaps the computational tradeoff in bookkeeping simplicity is worth the loss of efficiency in allowing for null steps. This is somewhat a matter of user taste and testing.

We now determine the change in the expected number of individuals of each type at a given site $\mathbf{r}$ in a single event. For full foragers, the change in its number $N_F(\mathbf{r})$ at site $\mathbf{r}$ is

$$\Delta N_F(\mathbf{r}) = \Big\{F(\mathbf{r})\big(\lambda-\sigma(1-R(\mathbf{r}))\big)+\rho H(\mathbf{r})R(\mathbf{r})+D_F\big[\sum_{\mathbf{r}'}F(\mathbf{r}')-zF(\mathbf{r})\big]\Big\}/\mathcal{R}\,. \quad (8a)$$

The terms proportional to F accounts for processes in which a full forager is picked, while the term proportional to H accounts for processes in which a hungry forager is picked and converts to a full forager. The terms proportional to D_F account for nearest-neighbor hopping: $F(\mathbf{r})$ increases because full foragers at neighboring sites $\mathbf{r}'$ hop to $\mathbf{r}$, while $F(\mathbf{r})$ decreases because full foragers at $\mathbf{r}$ hop to one of the z neighboring sites. Thus the change in the density of full foragers at site $\mathbf{r}$ is (after taking the continuum limit for the nearest-neighbor hopping)

$$\Delta F(\mathbf{r}) = \frac{\Delta N_F(\mathbf{r})}{V} = \left[F(\mathbf{r})\big(\lambda - \sigma(1-R(\mathbf{r}))\big) + \rho H(\mathbf{r})R(\mathbf{r}) + \nabla^2 F(\mathbf{r})\right]/V\mathcal{R}\,. \tag{8b}$$

If we take the time step for each event to be $\Delta t = (V\mathcal{R})^{-1}$, the above reduces to the partial differential evolution equation (6) for $F(\mathbf{r})$. In each microscopic event of the model, the time should be advanced by $\Delta t = (V\mathcal{R})^{-1}$.

In a similar fashion, the change in the density of hungry foragers is

$$\Delta H(\mathbf{r}) = \left[-H(\mathbf{r})(\rho R(\mathbf{r}) + \mu) + F(\mathbf{r})\sigma(1-R(\mathbf{r})) + \nabla^2 H(\mathbf{r})\right]/V\mathcal{R}\,. \tag{9}$$

Finally, the change in the resource density is

$$\Delta R(\mathbf{r}) = \left[R(\mathbf{r})\big(\alpha(1-R(\mathbf{r}))\big) - (F(\mathbf{r}) + H(\mathbf{r}))\right]/V\mathcal{R}\,. \tag{10}$$

The equations for ΔF, ΔH, and ΔR would then reproduce the partial differential equations (6) when the time step for an elemental event is taken to be $\Delta t = (V\mathcal{R})^{-1}$.

In this algorithm, notice that single-particle and two-particle reactions are treated differently. In the former case, a particle of the right type is picked at random. In the latter case, a random *pair* of particles of the appropriate types is picked. Again, there is no constraint on the number of foragers on any site, but given that we are describing a harsh environment, the parameters should be chosen so that this number is not large. The carrying capacity of the resource has been set to 1, so that the number of individual resources at any site must be either 0 or 1.

It would be nice if there was a simple way to adjust parameters in the above lattice simulation so as to recover the mean-field limit. I describe a simple modification of the steps in the event-driven lattice simulation that should recover the mean-field limit. We use the same rate as in Eq. (7), but the actual update steps are slightly modified compared to the rules of the lattice simulation above. The differences are highlighted in italics. I'm also assuming that the simulation with steps (ii'), (iii"), (v'), and (vi') are being used.

(i) Reproduction. With probability $\lambda F/\mathcal{R}$, pick one of the full foragers at random and let it reproduce. *Place the offspring anywhere.*

(ii) Starvation. With probability $\sigma F/\mathcal{R}$, *change a random full forager to hungry with probability* $(1-R)$.

(iii) Consumption. With probability $mR/\mathcal{R}$, pick a random resource unit. If the site is also occupied by a full forager, remove the resource.

(iv) Death. With probability $\mu H/\mathcal{R}$, pick and remove a random hungry forager.

(v) Recruitment. With probability $\rho H/\mathcal{R}$, pick a random hungry forager. *Change it to full with probability R and also remove a random resource.*

(vi) Growth. With probability $\alpha R/\mathcal{R}$, pick a random resource. Create a resource anywhere at random with probability $1 - R$.

(vii) F Diffusion. With probability $D_F\, F/\mathcal{R}$, pick a random full forager and move it to a neighboring site.

(viii) H Diffusion. With probability $D_H\, H/\mathcal{R}$, pick a random hungry forager and move it to a neighboring site.

By explicit construction, the probabilities for the above events sum to 1.

It is important to appreciate that there are multiple ways to simulate the elemental reaction steps that involve a tradeoff between bookkeeping complexity and computational efficiency and still reproduce the same behavior as in the rate equations (1) or the reaction-diffusion equations (6) in the limit of a large system.