

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 discrete states—full and hungry. Full foragers F are those that have just encountered and consumed a unit of resource R . On the other hand, 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 a parameter σ and the density of non-resources. Whenever a forager, either full or hungry, encounters a resource, one unit of the resource is consumed. If the forager was hungry, it turns into a full forager with rate ρ . During the time that a forager is hungry, it dies with mortality rate μ , while full foragers do not experience mortality risk. Furthermore, full foragers reproduce with rate λ . Finally, we assume that, in the absence of foragers, the underlying resource undergoes logistic growth, with growth rate α and carrying capacity equal to one.

According to 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:

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

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 one can 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 individual resources. The total number of particles $N = N_F + N_H + N_R$. There are six basic processes embodied by the rate equations (1):

- (i) Reproduction: $F \rightarrow 2F$
- (ii) Starvation: $F \rightarrow H$
- (iii) Recruitment: $H \rightarrow F$ and $R \rightarrow 0$
- (iv) Death: $H \rightarrow 0$
- (v) Growth: $R \rightarrow 2R$
- (vi) Consumption: $R \rightarrow 0$

Note that recruitment also involves consumption of a resource, so we must be careful to not double count. So I define “Recruitment” as the consumption of the resource by a hungry forager and “Consumption” as the consumption of the resource by a full forager.

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. The total rate for any event is proportional to

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

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 only up to an overall constant, but this constant is immaterial because the time step is also proportional to this same constant.

At the simplest level one merely keeps 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. Thus the algorithm outlined below is based on populating a lattice with the three types of entities and then updating the occupancy on the lattice sites according to the following event-driven simulation:

- (i) Reproduction. With probability $\lambda F/\mathcal{R}$, pick one of the full foragers and allow it to reproduce. The offspring should be placed anywhere in the system; this is consistent with there being no spatial structure.
- (ii) Starvation. With probability $\sigma(1 - R)F/\mathcal{R}$, pick a full forager and change it to a hungry forager. The factor $1 - R$ is the fraction of resource-free space.
- (iii) Recruitment. With probability $\rho HR/\mathcal{R}$, pick a hungry forager, turn it into a full forager, and remove the resource.
- (iv) Death. With probability $\mu H/\mathcal{R}$, pick a hungry forager and remove it.
- (v) Growth. With probability $\alpha R(1 - R)/\mathcal{R}$, pick a resource unit and create another unit of resource anywhere in the system.
- (vi) Consumption. With probability $(F + H)R/\mathcal{R}$, pick a resource unit and remove it.

By explicit construction, the probabilities for the above events sum to 1. Note also that a good test of the correctness of the above lattice-based algorithm is to compare it with the three-variable update rule that is mentioned at the start of this paragraph.

We now 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

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

The terms 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

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

Thus 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 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 expected number of hungry foragers in a single event is given by

$$\Delta N_H = [-H(\rho R + \mu) + F\sigma(1 - R)] / \mathcal{R}, \quad (4a)$$

so that the change in the density of hungry foragers simply is

$$\Delta H = [-H(\rho R + \mu) + F\sigma(1 - R)] / V\mathcal{R}. \quad (4b)$$

Finally, the change in the expected number of individual resources in a single event is given by

$$\Delta N_R = [R(\alpha(1 - R)) - (F + H)] / \mathcal{R}, \quad (5a)$$

so that the change in the resource density is

$$\Delta R = [R(\alpha(1 - R)) - (F + H)] / V\mathcal{R}. \quad (5b)$$

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 event is $\Delta t = (V\mathcal{R})^{-1}$. To summarize, pick an event according to the probabilities enumerated above and update the densities according (3b), (4b), and (5b), and then increment the time by $(V\mathcal{R})^{-1}$ after each event. In the three-variable simulation, we keep only N_F , N_H , and N_R . In an update, pick an event according to the above probabilities, update N_F , N_H , and N_R appropriately, and then increment the time.

There are several aspects of this algorithm that need to be specified for a lattice simulation. First, there is no constraint on the number of foragers on any site, but given that we are describing a harsh environment, it seems that the parameters should be chosen so that this number is not large. Since the carrying capacity of the resource has been set to 1, this means that N_R cannot exceed V . However, the number of resource units on any given site can exceed 1. With these details, a simulation that is based on the above steps should reproduce the predictions of the rate equations (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

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

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

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

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 one of the full foragers at random and let it reproduce. Since the process occurs on a finite-dimensional lattice, the offspring should be placed close to the parent. This could be either on the same site or 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. There is a way to get around tracking sites that are simultaneously resource free and occupied by a full forager. This leads to the alternative rule:

- (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) Recruitment. With probability $\rho H R/\mathcal{R}$, pick a random hungry forager on a site that contains a resource unit and change the forager from hungry to full and remove the resource. This process also involves a two-body reaction and perhaps a simpler way to implement this two-body feature is the following:
- (iii') Recruitment'. With probability $\rho H/\mathcal{R}$, pick a random hungry forager. If the site contains a resource unit, change the forager from hungry to full and remove the resource.
- (iv) Death. With probability $\mu H/\mathcal{R}$, pick and remove a random hungry forager.
- (v) Growth. With probability $\alpha R(1 - R)/\mathcal{R}$, pick a random neighboring pair of sites in which one contains a resource unit and the neighbor does not. Create a resource on the site without a resource. A computationally simpler alternative is the following:
- (v') 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. Thus 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 $(1 - R)_{\text{local}}$.
- (vi) Consumption. With probability $(F + H) R/\mathcal{R}$, pick a random resource unit on a site that is also occupied by a forager of either type and remove the resource. Alternatively:
- (vi') With probability R/\mathcal{R} , pick a random resource unit. If the site also contains a forager, remove the resource. Note the implicit assumption that the density of foragers is not large because the rate of resource removal does not increase if there are multiple foragers on the site. If this forager density is large then the following will work:
- (vi'') With probability $(F + H)/\mathcal{R}$, pick a random forager (either full or hungry). 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 perhaps it's best to use this last alternative.
- (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 steps (ii'), (iii'), (v'), and (vi'') are computationally simpler than the unprimed steps, but allow for the possibility of a null step. 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.

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

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

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)

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

Thus 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 $F(\mathbf{r})$. Thus 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

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

Finally, the change in the resource density is

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

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 $\Delta t = (V\mathcal{R})^{-1}$.

There are various aspects of the simulation that require discussion. First, single-particle and two-particle reactions must be 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 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 I think 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) Recruitment. With probability $\rho H/\mathcal{R}$, pick a random hungry forager and *change it to full with probability R .*
- (iv) Death. With probability $\mu H/\mathcal{R}$, pick and remove a random hungry forager.
- (v) Growth. With probability $\alpha R(1 - R)/\mathcal{R}$, pick a random neighboring pair of sites in which one contains a resource unit and the neighbor does not contain a resource unit. Create a resource on the site without a resource (see below for a simpler alternative).
- (vi) With probability $(F + H)/\mathcal{R}$, pick a random forager (either full or hungry). *With probability R remove a randomly chosen resource unit.*
- (vii) Consumption. With probability $(F + H)R/\mathcal{R}$, pick a random resource unit on a site that is also occupied by a forager of either type and remove the resource.
- (viii) Diffusion. With probability $D_F F/\mathcal{R}$, pick a random full forager and move it to a neighboring site.
- (ix) 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.