The Multiscale Stochastic Cellular Automaton Model

1 Model Description

We construct a multiscale model based on an extension of the concept of cellular automata (CA) that uses probabilistic rules to govern dispersion process while a microscale physical process drives population growth. The interrelated processes at microscale and macroscale are connected by passing rescaled information back and forth. For simplicity, we utilize a birth-death process at the microscale, and leverage the stochastic CA model to describe dispersion at the macroscale. The model easily accommodates other microscale models.

At the macroscale, dispersion is modelled by a stochastic process of the movements of population density on a grid of polygons. The dynamics are generated by probabilistic rules for mortality, reproduction, immigration related to distance, and chance of successful colonization. At the microscale, we use a model of local population dynamics of individuals. For our application, we choose the well-known birth-death process. The model easily accommodates other microscale models.

We use a sequential approach to link the micro and macroscales. Information summarizing microscale behavior is passed to macroscale models by an "upscaling" process, while the macroscale model is processed to provide information for the microscale model. In particular, we convert the expected value of population size obtained from the microscale model into a cell density by using the ratio between population size to the carrying capacity. The values of these densities provide the initial conditions for macroscale stochastic process on the macroscale time step. After simulating the dispersion process at the macroscale, we convert the updated cell density back to population size in each cell. The updated population size is used as the new starting value for the birth-death process. To determine the appropriate number of microscale time steps for each macroscale time step, we use numerical experiments.

1.1 Domain

We use a tessellation of the domain comprised of hexagonal cells. The benefits of hexagonal cells include: 1. A higher fidelity description of geographic features; 2. It is easier to identify neighbors of a given cell; 3. The distance between centroids of hexagonal cells is the same in all six directions; 4. Hexagonal grids let movement occur in six directions on an equal basis.

The model depends on identifying the Moore neighbourhood of a cell, which is the cell and the six neighbouring cells.We let Z^2 denote the disjoint collection of $m \times n$ uniformly sized cells organized in m rows and n columns. Cell states are represented by normalized population densities with values in interval $\Pi = [0, 1]$, so the configuration space is $Q = \Pi^{Z^2}$. The density in a cell only potentially moves to its neighbours in the cell's Moore neighbourhood. We group all cells that potentially interact at a given time step in an "interaction neighbourhood". An interaction neighbourhood contains cells in which dispersion can occur. Any group of affected cells that are not separated by more than two nonaffected (zero density) cells along with all immediate neighbours form an interaction neighbourhood.

Transition rules specify how the density in the cells move between cells. The new state $\pi_{ij}^{(t+1)}$ of cell (i, j) depends on current states of the cells in the Moore neighbourhood of cell (i, j) according to the local transition rule F.

$$\pi_{ij}^{(t+1)} = \begin{cases} F\left(\pi_{ij}^{(t)}, \pi_{i-1,j}^{(t)}, \pi_{i,j-1}^{(t)}, \pi_{i,j+1}^{(t)}, \pi_{i+1,j-1}^{(t)}, \pi_{i+1,j}^{(t)}, \pi_{i+1,j+1}^{(t)}\right), & j \text{ even}, \\ F\left(\pi_{ij}^{(t)}, \pi_{i-1,j-1}^{(t)}, \pi_{i-1,j}^{(t)}, \pi_{i-1,j+1}^{(t)}, \pi_{i,j-1}^{(t)}, \pi_{i,j+1}^{(t)}, \pi_{i+1,j}^{(t)}\right), & j \text{ odd}. \end{cases} \tag{1.1}$$

In the model, we consider a density-dependent dispersion process in which population density drives individual movements. At each time step, the outcome of local cell transitions is specified by the *inclusion probability*, denoted by $p_{I,ij}^{(t)}$ of including a cell (i, j) in the interaction process;

$$p_{I,ij}^{(t)} = \left[\frac{\left(w_{ij}^{(t)}\right)^{\alpha_{ij}^{(t)}}}{1 + \left(w_{ij}^{(t)}\right)^{\alpha_{ij}^{(t)}}} d_{ij}^{(t)} \right]^{\beta_{ij}^{(t)}}, \tag{1.2}$$

where $w_{ij}^{(t)}$ is the inclusion weight for cells and $d_{ij}^{(t)}$ is the maximum of relative density difference for cells in the Moore neighborhood. We assume that the probability of a cell being included in dispersion increases as number of neighbouring affected cells increases. The inclusion weight $w_{ij}^{(t)}$ determines the likelihood of inclusion in dispersion for cell (i, j)

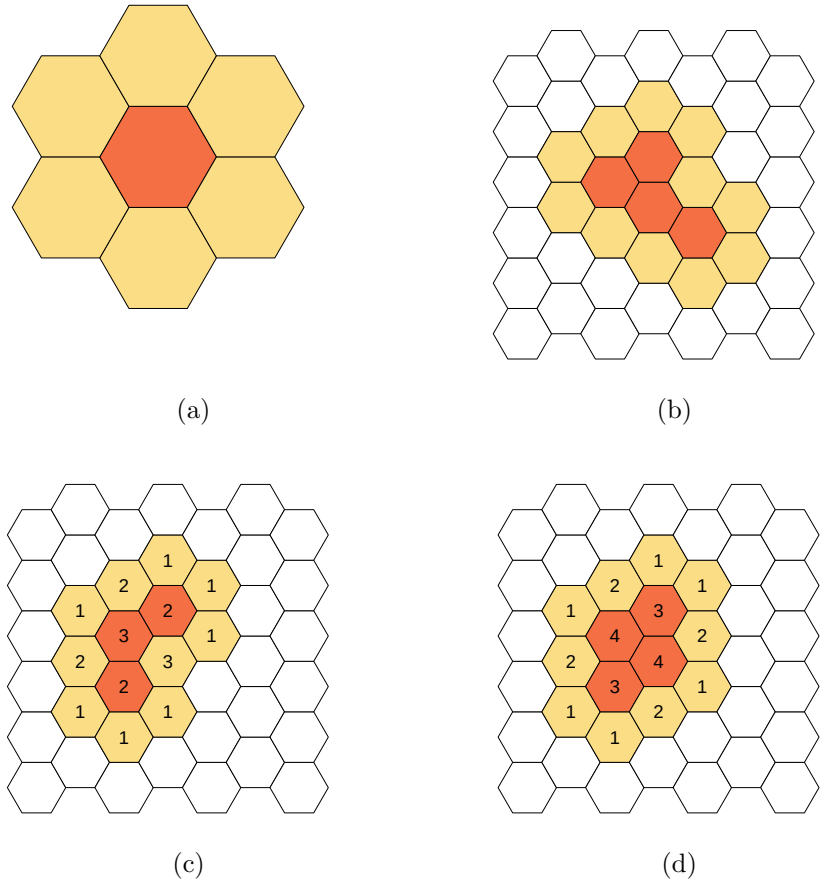
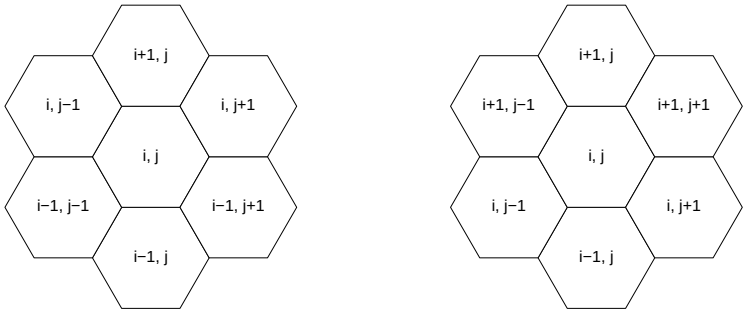


Figure 1: (a) The Moore neighbourhood of the central dark shaded cell. (b) An interaction neighbourhood. Population density can move within each interaction neighbourhood. The plots (c) and (d) present the inclusion weights $w_{ij}^{(t)}$ for cells in interaction neighbourhood at subsequent time steps. The number in each cell represents its inclusion weight. (c) An interaction neighbourhood with three affected cells. (d) The updated interaction neighbourhood at the next time step when one more cell becomes affected.



(a) (b)

Figure 2: The indexing for adjacent cells when the column index j is (a) odd and (b) even.

at time t. For non-affected cells, the inclusion weight $w_{ij}^{(t)}$ is calculated as the number of adjacent affected neighbours at each time step. Affected cells are more likely to be involved in dispersion process, so the inclusion weight $w_{ij}^{(t)}$ for affected cells is the number of its

adjacent affected neighbours plus 1.

We assume that the relative density difference has an impact on the dispersion between neighbouring cells and dispersion is less likely to happen between neighbouring cells when their densities are close to each other. We set $d_{ij}^{(t)}$ equal to the maximum value of the density difference between the cell (i, j) and its six neighbouring cells. Finally, the parameters $\alpha_{ij}^{(t)}$ and $\beta_{ij}^{(t)}$ are used to increase or decrease the difference in inclusion probability between high density cells and low density cells. Note that all of the parameters can be varied for each cell (i, j) at each time step t. For an affected cell (i, j) at time t, we define the Bernoulli random variable $B_{I,ij}^{(t)}$ with

The inclusion probability increases as α increases and decreases as β increases. We assume that population tends to disperse into areas with a low population density. The inclusion probabilities increase as $\alpha_{ij}^{(t)}$ increases. We set $\beta_{ij}^{(t)} \in (0, 1)$ to increase the inclusion probability, and set $\beta_{ij}^{(t)} \geq 1$ to decrease the inclusion probability. The process of dispersion is as follows.

$$B_{I,ij}^{(t)} = \begin{cases} 1, & \text{with probability } p_{I,ij}^{(t)}, \\ 0, & \text{with probability } 1 - p_{I,ij}^{(t)}. \end{cases} \tag{1.3}$$

Only cells with $B_{I,ij}^{(t)} = 1$ are included in dispersion.

We divide the included cells into the contributing cells whose density is higher than the mean density of all the included cells in the same interaction neighbourhood and the non-contributing cells, which are the rest. The contributing cells are $\{\pi_{i_h}^{(t)}\}_{h=1}^{H^{(t)}}$, where $\pi_{i_h}^{(t)} \geq \bar{\pi}^{(t)}$, $\bar{\pi}^{(t)}$ is the average density of the inclusive cells in each interaction neighbourhood, and $H^{(t)}$ is the number of contributing cells at time step t. The non-contributing cells are $\{\pi_{i_k}^{(t)}\}_{k=1}^{K^{(t)}}$, where $K^{(t)}$ represents the number of non-contributing cells at time t. Then the total amount of density to be dispersed is,

$$\tilde{\pi}^{(t)} = \sum_{h=1}^{H^{(t)}} \gamma_1(\pi_{i_h}^{(t)} - \bar{\pi}^{(t)})$$

where the parameter γ_1 determines the percentage of population to be dispersed.

In the redistribution of density among the cells within the interaction neighbourhood, we balance the loss in the contributing cells and the gain in others. We allow the contributing cells to lose a proportion of the density above $\bar{\pi}^{(t)}$ so the temporary state for contributing cells becomes,

$$\pi_{i_h}^{(t)} = \pi_{i_h}^{(t)} - \gamma_1(\pi_{i_h}^{(t)} - \bar{\pi}^{(t)}), \quad h = 1, 2, \ldots, H^{(t)}.$$

We want to allow some of the dispersed population to be "lost". We assume the population survival rate is γ_2, and the background mortality rate $1-\gamma_2$ defines the population loss during the dispersion process. Assuming that the population disperses uniformly within the same interaction neighbourhood, the state for contributing cells after receiving population is,

$$\pi_{i_h}^{(t+1)} = \pi_{i_h}^{(t)} - \gamma_1(\pi_{i_h}^{(t)} - \bar{\pi}^{(t)}) + \frac{1}{H^{(t)}+K^{(t)}}\gamma_2\tilde{\pi}^{(t)}, \quad h = 1, 2, \ldots, H^{(t)}.$$

The state for non-contributing cells becomes,

$$\pi_{i_k}^{(t+1)} = \pi_{i_k}^{(t)} + \frac{1}{H^{(t)}+K^{(t)}}\gamma_2\tilde{\pi}^{(t)}, \quad k = 1, 2, \ldots, K^{(t)}.$$

The dispersion process is performed for all interaction neighbourhoods simultaneously.

We model the potential loss using a survival probability for successful dispersion to low density cells. The motivation to allow for dispersed population to be lost is inspired by the field experiments conducted to investigate the dispersion of MPB described in [**?**]. The survival probability has two components. First, we use a parameter ϕ to quantify the effect of natural conditions and geographical properties of the non-contributing cells, e.g. wind, light and temperature. Second, we use the arctan function with two parameters ρ and ξ to quantify the density-related effect during population establishment in the new area. The survival probability is,

$$p_{S,ij}^{(t)} = \phi_{ij}^{(t)} + \rho_{ij}^{(t)} \arctan(\xi_{ij}^{(t)}\pi_{ij}^{(t)}), \tag{1.4}$$

where the shape parameters $\phi_{ij}^{(t)}$, $\rho_{ij}^{(t)}$ and $\xi_{ij}^{(t)}$ are used to adjust the survival probability for each non-contributing cell at each time step t. Then we define a Bernoulli random variable $B_{S,ij}^{(t)}$ each non-contributing cell,

$$B_{S,ij}^{(t)} = \begin{cases} 1, & \text{with probability } p_{S,ij}^{(t)}, \\ 0, & \text{with probability } 1 - p_{S,ij}^{(t)}. \end{cases} \tag{1.5}$$

Only the non-contributing cells with $B_{S,ij}^{(t)} = 1$ successfully receive density. The density is set to zero for non-contributing cells with $B_{S,ij}^{(t)} = 0$. Note that the survival probabilities tend to increase as the population density increases.

1.2 Boundary Conditions

We need to prescribe boundary conditions on the boundary of the domain. The formulation of interaction neighbourhoods are different at a boundary as well. In the SCA model, we impose absorbing boundary conditions. We create "auxiliary" adjacent cells for those boundary cells that have no neighbours and assume the auxiliary cells can receive density from the boundary cells, but cannot send density to other cells in the interaction neighbourhood. We note that most cells that are on the boundary have a lower density than the cells in the interior.

2 Application to Mountain Pine Beetle Infestations

In the application to the dispersion of MPB, we may use the microscale model to model population on a single tree. In that case, the SCA model describes dispersion of MPB between trees. Alternatively, we may use the microscale model to describe MPB populations on patches of trees, so each cell in the SCA represents a small patch of trees with similar infestation severity and the SCA models dispersion between patches. The microscale model is simulated multiple times, and the cell state is taken as the average of the microscale model output. The decision on these two alternatives depends on the tree density in the region.

2.1 Microscale process model

We use a birth-death process for the microscale to describe individual population in each cell. A birth-death process is a continuous-time Markov chain that counts the number of individuals in population over time. Let X_τ denote the population size at time τ, which can take values in the finite state space $\{0, 1, 2, \ldots, C\}$, where C is the maximum population size (capacity). The rate of births and deaths in a population of size n is determined by instantaneous birth rate b_n and instantaneous death rate d_n, respectively.

Given n individuals at time τ, the probability of a birth in the interval $(\tau, \tau + \Delta\tau)$, where τ si small, is

$$Pr(X_{\tau+\Delta\tau} = n + 1 | X_\tau = n) = b_n \Delta\tau + o(\Delta\tau). \tag{2.1}$$

Recall the notation $o(\Delta\tau)$ represents terms that have smaller order than $\Delta\tau$, i.e.,

$$\lim_{\Delta\tau \to 0} \frac{o(\Delta\tau)}{\Delta\tau} = 0.$$

The probability of a death in $(\tau, \tau + \Delta\tau)$ is

$$Pr(X_{\tau+\Delta\tau} = n - 1 | X_\tau = n) = d_n \Delta\tau + o(\Delta\tau). \tag{2.2}$$

Sin $\Delta\tau$ is small, the transition probabilities are

$$Pr(X_{\tau+1} = j | X_\tau = i) = \begin{cases} b_i, & \text{if } j = i + 1, \\ d_i, & \text{if } j = i - 1, \\ 1 - b_i - d_i, & \text{if } j = i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

The transition matrix P has the form,

$$P = \begin{bmatrix} 1 & d_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 - (b_1 + d_1) & d_2 & 0 & \dots & 0 & 0 \\ 0 & b_1 & 1 - (b_2 + d_2) & d_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 - (b_{N-1} + d_{N-1}) & d_N \\ 0 & 0 & 0 & 0 & \dots & b_{N-1} & 1 - d_N \end{bmatrix} \tag{2.4}$$

We consider two birth rates. The first is related to the logistic growth model with capacity C and $b_n = nb(1-n/C)$. The second is $b_n = nb(1-n/C)+b_0 nb(1-n/C)\sin(2t\pi/m)$ which adds a periodic effect, where b_0 is a constant reflecting the size of periodic effect, t is the macroscale time point and m is a constant used to determine the time period of the birth rate. For each birth-death process within the affected cells, we generate random integers X_τ taking values -1, 0 or 1 for microscale time step $\tau = 1, \dots, 100$. The probabilities for generating each value depends on the cell density and the predefined birth and death rate:

$$X_\tau = \begin{cases} 1, & \text{with probability } b_{ij,\tau} = n^* b(1 - n^*/C)/N_t, \\ -1, & \text{with probability } d_{ij,\tau} = n^* d/N_t, \\ 0, & \text{with probability } 1 - b_{ij,\tau} - d_{ij,\tau}. \end{cases}$$

where b and d represent the predefined birth and death rate, respectively.

Assuming N_t microscale time steps for each macroscale time step, so $\Delta\tau = 1/N_t$, and the transition probabilities at each microscale time step can be written as

$$Pr(X_{\tau+\Delta\tau} = n + 1 | X_\tau = n) = \frac{b_n}{N_t}$$

and

$$Pr(X_{\tau+\Delta\tau} = n-1 | X_\tau = n) = \frac{d_n}{N_t}.$$

In our simulations, we set the population capacity $C = 100$ and $N_t = 100$. We relate the microscale and macroscale in the following way. Given a specified initial cell density π^*, we obtain the population size n^* by multiplying π^* by the capacity C in the cell. If $\pi^* C < 1$, then we set the population size to be 0. At the final microscale time step, we convert the microscale population to a density by dividing by C. If it is larger than 1, we set the density to be 1.

We assume there are multiple trees in each cell and each tree comes with a different density and birth-death process. Therefore, we simulate the birth-death process multiple times in each cell, and take the average of those processes to represent the average behaviour of those trees in the cell. In the simulation studies, we simulate the birth-death process for 20 realizations in each cell, and take the average of the final population size to obtain the updated cell density.

3 Implementation

We use the R programming language (version 4.1.1) to build a simulator that takes an initial cell configuration and a set of parameter values for the SCA transition rules and outputs the cell configuration evolution at a sequence of times. We set up a domain that is composed of $m \times n$ non-overlapping cells. Typically, $m = 30$ and $n = 30$ or $m = 50$ and $n = 50$. It turns out that handling the "merging" of two interaction neighbourhoods is computationally expensive, so we adopt a grouping scheme built on a grid-based clustering approach that has a faster processing time.

For cells on the boundary of the domain, we create auxiliary adjacent neighbours outside the domain that could receive population from the affected cells. We remove the auxiliary cells at the end of each dispersion iteration. The auxiliary cells only receive population and never send populations to other cells in the interaction neighbourhood.

Algorithm 1 Algorithm for creating interaction neighbourhoods

1: Determine all affected cells in the domain.
2: **for** each affected cell in the domain **do**
3: Randomly pick an affected cell as the starting point, where the cell has not been traversed before.
4: Create a two-cell radius shell around the picked cell, check if there is any affected cell(s) in the shell. If yes, include the affected cell(s) with the picked cell to form one interaction neighbourhood; otherwise, the starting point cell forms one separate interaction neighbourhood.
5: Repeat the step 3 and 4 until there are no affected cells in the two-cell radius shell.
6: **end for**
7: All cells will be associated with some interaction neighbourhood.

Algorithm 2 Algorithm for simulating the dispersion process

1: Set up the implementation grid of $m \times n$ cells, where $m = n = 50$.
2: Set the initial pattern for a small number of affected cells on the grid.
3: **for** macroscale time step $t = 1, 2, \ldots$ **do**
4: Simulate multiple realizations of birth-death process and compute the average number of individuals at the microscale within each cell for microscale steps to reach the macroscale time step.
5: Compute the ratio of the average number of individuals and the cell capacity at the microscale and use the ratio as the cell density for dispersion process at the macroscale.
6: Simulate the dispersion process at the macroscale.
7: Convert the density to population size by multiplying by capacity to produce an initial population for the microscale simulation in each cell.
8: Alternate the microscale and macroscale simulations for sufficient time steps to obtain the ending cell states.
9: During the simulation, compute the statistics that are used to describe the evolution.
10: **end for**