Agent-Based Modeling and Simulation

From Animations to Science

Dr. Alejandro Guerra-Hernández

Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana
Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,
Nuevo Xalapa, Xalapa, Ver., México 91097

mailto:aguerra@uv.mx
https://www.uv.mx/personal/aguerra/abms

Doctorado en Inteligencia Artificial 2024

Universidad Veracruzana

Credits

- These slides are based on the book of Railsback and Grimm [2], chapter 5.
- Any difference with this source is my responsibility.
- This work is licensed under CC-BY-NC-SA 4.0
- To view a copy of this license, visit:

  https://creativecommons.org/licenses/by-nc-sa/4.0/

Universidad Veracruzana

Using and documenting the model

- ABMS is not only about formulating and implementing models, real work starts after first implementation.
- Then, we use the model to find answers and solutions to the questions and problems we started our modeling project with.
- This almost always requires modifying the model formulation and software.
- Usually, the iterative modeling cycle is not documented, i.e., models are typically presented as static entities that were just produced and used.
- In fact, every model description is only a snapshot of a process.

Universidad Veracruzana

The NetLogo case

- The Models Library has a similar problem: it presents the models and givesn some hints of how to analyze them, but it cannot demonstrate how to do science with them.
- These models are very good at animation, letting us see what happens as their assumptions and equations are executed.
- But they do not show you how to:
  - Explore ideas and concepts;
  - Develop and test hypotheses; and
  - Look for parsimonious and general explanations of observed phenomena.

Universidad Veracruzana

The Buttefly model enhanced I

- How do we turn our Butterfly model into a scientific model?
- The purpose of this model was to explore the emergence of virtual corridors, *i.e.*, places where butterflies move in high concentrations even though there is nothing there attracting them.
- But our model does not tell anything about corridors and when and how strongly they emerge.
- Therefore, we will now produce quantitative output that can be analyzed, instead of just the visual display of butterfly movement.

Universidad Veracruzana

The Buttefly model enhanced II

- Then, we will also replace our very simple and artificial landscape with a real one read in from a topography file.
- In the exercises we suggest some scientific analyses for you to conduct on the Butterfly model.

Universidad Veracruzana

Learning Objectives I

- Understand the concept that using an ABM for science requires producing quantitative output and conducting simulation experiments; and execute your first simulation experiment.
- Learn to define and initialize a global variable by creating a slider or switch on the Interface.
- Develop an understanding of what reporters are and how to write them.
- Start learning to find and fix mistakes in your code.

Universidad Veracruzana

Learning Objectives II

- Learn to create output by writing to an output window, creating a time-series plot, and exporting plot results to a file.
- Try a simple way to import data into NetLogo, creating a version of the Butterfly model that uses real topographic data.

What is a corridor?

- We need a way to quantitatively observe the extent to which virtual corridors emerge. But how would we characterize a corridor?
- If all individuals followed the same path (as when they all start at the same place and q is 1.0) the corridor would be very narrow; or if movement were completely random we would not expect to identify any corridor-like feature.
- But we need to quantify how the width of movement paths changes as we vary things such as q or the landscape topography.

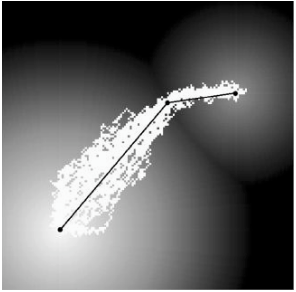
Universidad Veracruzana

Pe'er definitions revised

- Pe'Er, Saltz, and Frank [1] quantified corridor width by dividing the number of patches visited by all individuals during 1000 time steps by the distance between the start patch and the hill's summit.
- In our version of the model, different butterflies can start and end in different places, so we slightly modify this measure:
  - Assume that each butterfly stops when it reaches a local hilltop, *i.e.*, a patch higher than all its eight neighbor patches.
  - The width of a corridor is quantified as the number of patches that are visited by any butterfly divided by the mean distance between starting and ending locations, over all butterflies.
- This measure will be small (approaching 1.0) when all butterflies follow the same, straight path uphill; but should increase as they increasingly follow different paths.

Universidad Veracruzana

Graphically



- 50 butterflies stating at 85,95 with $q = 0.4$.
- The number of white patches is 1956.
- The mean distance between butterfly staring and ending points is 79.2 patches.
- The corridor width is 24.7 patches.

Universidad Veracruzana

Analysis

- We are going to produce a plot of corridor width vs. q.
- When analyzing a model, we need to have a clear idea of the kind of plot we want to produce from what output, because this tells us what kind fo simulation experiments we have to perform and what outputs we need the program to produce.

Universidad Veracruzana

Implementation

- Because it is now obvious that we need to conduct experiments by varying q and seeing its effect, create a slider for it on the Interface tab. Set the slider so that q varies from 0.0 to 1.0 in increments of 0.01.
- Change the `setup` procedure so that 50 individuals are created and start from the same position.
- Then vary q via its slider and observe how the area covered by all the butterfly paths changes.

Universidad Veracruzana

Some tips about using sliders

- Only global variables can be controlled by sliders, switches, and choosers; not turtle or patch variables.
- Once a controller is created for the global variable, it cannot still appear in the **globals** statement. The controller defines and initializes the variable.
- We recommend to comment the variable to remember it exists.
- The biggest potential problem is forgetting to remove statements setting the variable's value in the `setup` procedure. It will take always the same value.

Universidad Veracruzana

Stop moving criteria

- Modify the move procedure so that butterflies no longer move if they are at a hilltop, *i.e.*, when the butterfly is in a patch with elevation higher than all its neighbors.
- Add this at the beginning of move:

```
if elevation >= [elevation] of max-one-of neighbors [elevation] ;; already
    in a hilltop
  [stop]
```

- Observe that turtles can access the variables of its current patch.
- Observe the use of elevation as a reporter in max-one-of.

Universidad Veracruzana

New turtles and patch variables

- Add a new variable to the **patches—own** statement: `visited?`.
- This variable will be true if any butterfly has landed in that patch.
- Being a boolean variable, we follow the NetLogo convention of ending its identifier with a question mark.
- Add a variable called `start-patch` to the **turtles—own** statement.
- In the `setup` procedure, add statements to initialize these new variables: `visited?` is set as false; `start-patch` can use the primitive `patch-here`.
- In NetLogo new variables have zero as value.

Universidad Veracruzana

Keeping track of visted patches

- Add a statement to the `move` procedure that causes the butterfly, once it has moved to a new patch, to set the patch's variable `used?` to true.

Universidad Veracruzana

Calculating the corridor width

- In the go procedure's that stops the program after 1000 ticks, insert a statement that executes once before execution stops.
- This procedure uses the very important primitive `let` to create a new local variable `final-corridor-width` and give it the value produced by a new reporter `corridor-width`.
- Write the skeleton of a reporter procedure `corridor-width`, that reports the mean path width of turtles. Look up the keyword **to—report** in the NetLogo Dictionary and read about reporters in the Procedures section of the Programming Guide.

Universidad Veracruzana

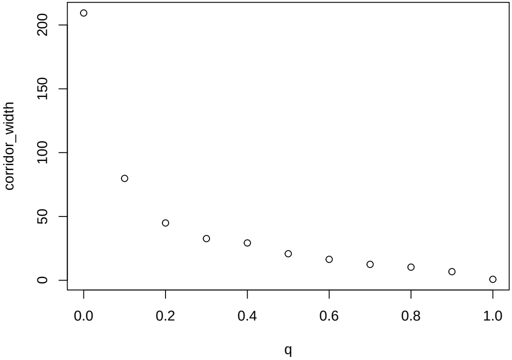
The corridor-width procedure

- Create a new local variable and set it to the number of visited patches (Hint: use the primitive `count`).
- Create a new local variable that is the mean, over all turtles, of the distance from the turtle's current patch and its starting patch. (Look at the primitives `mean` and `distance`).
- From the above two new local variables, calculate corridor width and report its value as the result of the procedure.
- In an output window added to the interface, print the message "Corridor width:" followed by the value of this variable at the end of each run.

Universidad Veracruzana

Playing with q

- Use the slider to vary q over a wide range, write down the resulting corridor widths, and the plot the corridor width versus q:



Universidad Veracruzana

Surprises

- These results are not very surprising because they mainly show that with less randomness in the butterflies' movement decisions, movement is straighter and therefore corridor width smaller.
- It seems likely that a less-artificial landscape would produce more interesting results.

Plotting

- Now let's try something we often need to do: examine results over time as the simulation proceeds instead of only at the end of the model run.
- Plots are extremely useful for observing results as a model executes.
- However, we still need results written down so we can analyze them, and we certainly do not want to write down results for every time step from a plot or the output window. Instead, we need NetLogo to write results out in a file that we can analyze.

Universidad Veracruzana

Updating corridor width

- But before we can add any of these time-series outputs, we have to change the model so it produces results every time step.
- Currently we calculate corridor width only once, at the end of a simulation; now we need to calculate it each time step:

```
to go ; This is the master schedule
  ask turtles [ move ]
  plot corridor-width
  tick
  if ticks >= 1000 [ stop ]
end
```

Universidad Veracruzana

Plot

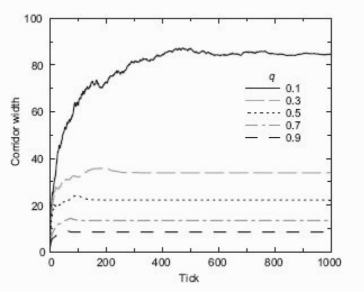
- The `plot` statement does two things:
  - It calls the `corridor-width` reporter to get the current value of the corridor width, and then
  - It sends that value to be the next point of a plot of the Interface tab.
- Add a plot named “Corridor width” to the Interface tab.
- The code for plotting can be written in the plot object, but it is preferable to put in the code tab, where it is visible and easily modifiable.
- Test-run the model with several values of q.

Exporting data

- Looking at the plot gives us an idea of how the corridor width changes with q and over time, but we need to export data:

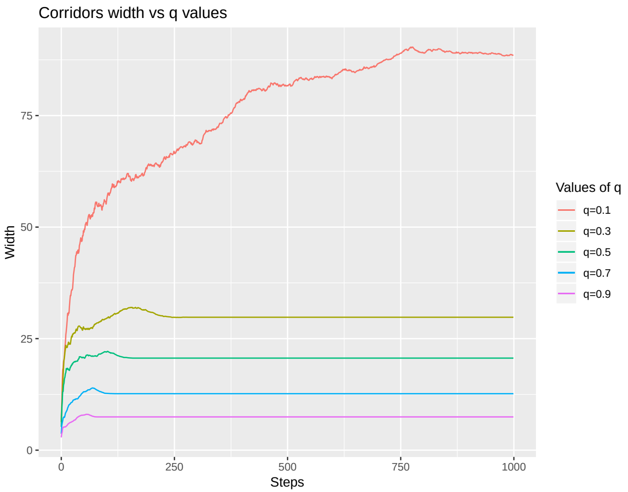
```
export-plot "Corridor width" word "Corridor-output-for-q-" q
```

- With a spreadsheet you can graph the changes over time for several values of q:



Universidad Veracruzana

My graph with R



Universidad Veracruzana

R Code

```r
qs <- read.csv(file = "corridor-width-qs.csv", header = TRUE, sep = ",")

summary(qs)

ggplot(qs,aes(step,q.0.1)) +
  geom_line(aes(color="q=0.1")) +
  geom_line(data=qs,aes(step,q.0.3,color="q=0.3")) +
  geom_line(data=qs,aes(step,q.0.5,color="q=0.5")) +
  geom_line(data=qs,aes(step,q.0.7,color="q=0.7")) +
  geom_line(data=qs,aes(step,q.0.9,color="q=0.9")) +
  labs(title="Corridors width vs q values", color="Values of q", x = "Steps",
       y = "Width")
```

Universidad Veracruzana

Grid-based Spatial Data

- Importing topographies of real landscapes, or any other spatial data, is straightforward –once the data are prepared for NetLogo.
- On the book's web site, a plain text file that you can import to NetLogo is provided: `ElevationData.txt`
- The file contains the mean elevation of 25-meter patches.
- By grid-based, we mean data records that include x-y coordinates and a variable value (elevation or anything else that varies over space).

Universidad Veracruzana

Transformations

- Real data are typically in coordinate systems, *e.g.*, UTM, that cannot be used directly.
- NetLogo requires patch coordinates defined as integers and that the World includes an origin 0,0.
- Transformations could be made via NetLogo code as the data are read in, but it is safest to do it in software such as a spreadsheet that lets you see and test the data before they go into NetLogo.
- These steps will prepare an input file that provides the value of a spatial variable to each patch.

Universidad Veracruzana

Origin

- NetLogo requires the point 0,0 to be in the World.
- It is easiest to put 0,0 at the lower left corner by identifying the minimum x-and y-coordinates in the original data, and then subtracting their values from the x-and y-coordinates of all data points.

Universidad Veracruzana

Distances

- NetLogo patches are one distance unit apart, so the data must be transformed so points are exactly 1 unit apart.
- Divide the x- and y-coordinates of each point by the spatial resolution (grid size, the distance between points), in the same units that the data are in.
- Example: If the points represent squares 5 meters on a side and the coordinates are in meters, then divide the x- and y-coordinates by 5.
- As a result, all coordinates should now be integers, and grid coordinates should start at 0 and increase by 1.
- Save the data as a plain text file.

Universidad Veracruzana

The World

- In NetLogo's Model Settings window, set the dimensions to match those of your data set (or, in the `setup` procedure, use the primitive `resize-world`).
- Turn world wrapping off (at least temporarily) so that NetLogo will tell you if one of the points in the input file is outside the World's extent.

Universidad Veracruzana

Reading data

- The code is something like this:

```
file-open "ElevationData.txt"
while [not file-at-end?]
[
  let next-x file-read
  let next-y file-read
  let next-elevation file-read
  ask patch next-x next-y
    [set elevation next-elevation]
]
file-close
```

Universidad Veracruzana

Exercise

- Change the `setup` procedure so it reads in elevations from the file `ElevationData.txt`.
- Look at the elevation data file to determine what dimensions your World should have.
- Change the statement that shades the patches by their elevation so the color is scaled between the minimum and maximum elevation (See, `min` and `max`).

Universidad Veracruzana

Last Changes

- Comment out the statement causing butterflies to stop if they are at a local hilltop. They will stop too soon in the real landscape.
- In the `crt` block, assign `xcor` and `ycor` randomly in a 10 by 10 neighborhood, located somewhere you choose. Butterflies won't start any more at the same place.

Universidad Veracruzana

Small steps

- Modeling a system of multiple agents.
- Adding quantitative observations that we can analyse.
- Starting to use simulation experiments.
- Using real spatial data.

The Butterfly Model

- The model is extremely simple, but it already requires some difficult decision.
- Example: Several different ways you could define widths and corridor sizes, which likely would produce different results.
- It is thus important to learn how to analyse simple models before you turn to more complex ones.
- The model is good to start because people in all disciplines can understand it, even when it has been used in serious ecological research. But remember that ABMs are not restricted to organisms moving in lanscapes.

Universidad Veracruzana

Referencias I

[1] G Pe'Er, D Saltz, and K Frank. "Virtual corridors for conservation management". In: *Conservation Biology* 19.6 (2005), pp. 1997–2003.

[2] SF Railsback and V Grimm. *Agent-Based and Individual-Based Modeling*. Second. Princeton, NJ, USA: Princeton University Press, 2019.

Universidad Veracruzana