Wolf Sheep Predation (NetLogo models library)

Uri Wilenski

Mon Mar 25 10:49:43 2019

Contents

Global

Subfiles

Description

A subfile to the wolf sheep model is included. This demonstrates that nls files are supported by nldoc.

```
__includes["Wolf Sheep Predation Extra.nls"]
```

Global variables

Description

There is only one global variable that stores the max number of sheep allowed

```
globals [ max-sheep ]  ; don't let sheep population grow too large
```

Breeds

Description

There are two breeds: sheep and wolves

```
breed [ sheep a-sheep ]  ; sheep is its own plural, so we use "a-sheep" as the singular.
breed [ wolves wolf ]
```

Agent properties

Description

Sheep and wolves have a energy variable. Patches have a countdown variable to store the current state of the grass regrowth countdown.

```
turtles-own [ energy ]       ; both wolves and sheep have an energy
patches-own [ countdown ]
```

Procedures

Setup

Description

The setup procedure first resets the model. Depending on the chosen model version, grass patches are initialized. Finally, wolves and sheep are created.

Go

Description

This is the main procedure of the model. It iterates over sheep and wolve agents. These agents then move, forage and die if they dont have enough energy.

Move

Description

Turtles turn left and right at random and then move one patch forward.

```
to move  ; turtle procedure
  rt random 50
  lt random 50
  fd 1
end
```

eat-grass

Description

If the current patch contains grass, the sheep eats it and gains energy. Then the patch color is set to brown

```
to eat-grass  ; sheep procedure
  ; sheep eat grass, turn the patch brown
  if pcolor = green [
    set pcolor brown
    set energy energy + sheep-gain-from-food  ; sheep gain energy by eating
  ]
end
```

reproduce-sheep

Description

Under a defined probability, a sheep may hatch a new offspring and loses 50% of its energy.

```
to reproduce-sheep  ; sheep procedure
  if random-float 100 < sheep-reproduce [  ; throw "dice" to see if you will reproduce
    set energy (energy / 2)                ; divide energy between parent and offspring
    hatch 1 [ rt random-float 360 fd 1 ]   ; hatch an offspring and move it forward 1 step
  ]
end
```

reproduce-wolves

Description

Under a defined probability, a wolf may hatch a new offspring and loses 50% of its energy.

```
to reproduce-wolves  ; wolf procedure
  if random-float 100 < wolf-reproduce [  ; throw "dice" to see if you will reproduce
    set energy (energy / 2)               ; divide energy between parent and offspring
    hatch 1 [ rt random-float 360 fd 1 ]  ; hatch an offspring and move it forward 1 step
  ]
end
```

eat-sheep

Description

If a wolf meets a sheep, the sheep dies and the wolf gains energy.

```
to eat-sheep  ; wolf procedure
  let prey one-of sheep-here                    ; grab a random sheep
  if prey != nobody  [                          ; did we get one?  if so,
    ask prey [ die ]                            ; kill it, and...
    set energy energy + wolf-gain-from-food     ; get energy from eating
  ]
end
```

death

Description

If a turtle loses all of its energy it dies.

```
to death  ; turtle procedure (i.e. both wolf nd sheep procedure)
  ; when energy dips below zero, die
  if energy < 0 [ die ]
end
```

grow-grass

Description

The patch countdown timer is reduced for all brown patches. If the countdown of a brown patch is 0, it turns green.

```
to grow-grass  ; patch procedure
  ; countdown on brown patches: if reach 0, grow some grass
  if pcolor = brown [
    ifelse countdown <= 0
      [ set pcolor green
        set countdown grass-regrowth-time ]
      [ set countdown countdown - 1 ]
  ]
end
```

grass

Return

number of green patches

Description

Reports the number of grass patches

```
to-report grass
  ifelse model-version = "sheep-wolves-grass" [
    report patches with [pcolor = green]
  ]
  [ report 0 ]
end
```

display-labels

Description

Sets energy levels as labels of sheep and wolves

```
to display-labels
  ask turtles [ set label "" ]
  if show-energy? [
    ask wolves [ set label round energy ]
    if model-version = "sheep-wolves-grass" [ ask sheep [ set label round energy ] ]
  ]
end
```

reset-patches

Description

This function just resets all patches to green color

```
to reset-patches
  ask patches
  [
    set pcolor green
  ]
end
```

reset-turtles

Description

This function creates some turtles

```
to reset-turtles
  ask turtles [die]
  create-turtles 10
end
```

calc-sum

Parameters

a any number b any number

Return

sum of a and b

Description

This function calculates the sum of two provided numbers

```
to-report calc-sum [a b]
  report (a + b)
end
```

GUI elements

Name	Type	Value	Properties
initial-number-sheep	SLIDER	100	min: 0; max: 250; incr: 1
sheep-gain-from-food	SLIDER	4	min: 0; max: 50; incr: 1
sheep-reproduce	SLIDER	4	min: 1; max: 20; incr: 1
initial-number-wolves	SLIDER	50	min: 0; max: 250; incr: 1
wolf-gain-from-food	SLIDER	20	min: 0; max: 100; incr: 1
wolf-reproduce	SLIDER	5	min: 0; max: 20; incr: 1
grass-regrowth-time	SLIDER	30	min: 0; max: 100; incr: 1
show-energy?	SWITCH	TRUE	
model-version	CHOOSER	sheep-wolves	validvalues: c("sheep-wolves", "sheep-wolves-grass")

Info Tab

WHAT IS IT?

This model explores the stability of predator-prey ecosystems. Such a system is called unstable if it tends to result in extinction for one or more species involved. In contrast, a system is stable if it tends to maintain itself over time, despite fluctuations in population sizes.

HOW IT WORKS

There are two main variations to this model.

In the first variation, the "sheep-wolves" version, wolves and sheep wander randomly around the landscape, while the wolves look for sheep to prey on. Each step costs the wolves energy, and they must eat sheep in order to replenish their energy - when they run out of energy they die. To allow the population to continue, each wolf or sheep has a fixed probability of reproducing at each time step. In this variation, we model the grass as "infinite" so that sheep always have enough to eat, and we don't explicitly model the eating or growing of grass. As such, sheep don't either gain or lose energy by eating or moving. This variation produces interesting population dynamics, but is ultimately unstable. This variation of the model is particularly well-suited to interacting species in a rich nutrient environment, such as two strains of bacteria in a petri dish (Gause, 1934).

The second variation, the "sheep-wolves-grass" version explictly models grass (green) in addition to wolves and sheep. The behavior of the wolves is identical to the first variation, however this time the sheep must eat grass in order to maintain their energy - when they run out of energy they die. Once grass is eaten it will only regrow after a fixed amount of time. This variation is more complex than the first, but it is generally stable. It is a closer match to the classic Lotka Volterra population oscillation models. The classic LV models though assume the populations can take on real values, but in small populations these models underestimate extinctions and agent-based models such as the ones here, provide more realistic results. (See Wilensky & Rand, 2015; chapter 4).

The construction of this model is described in two papers by Wilensky & Reisman (1998; 2006) referenced below.

HOW TO USE IT

1. Set the model-version chooser to "sheep-wolves-grass" to include grass eating and growth in the model, or to "sheep-wolves" to only include wolves (black) and sheep (white).
2. Adjust the slider parameters (see below), or use the default settings.
3. Press the SETUP button.
4. Press the GO button to begin the simulation.
5. Look at the monitors to see the current population sizes
6. Look at the POPULATIONS plot to watch the populations fluctuate over time

Parameters: MODEL-VERSION: Whether we model sheep wolves and grass or just sheep and wolves INITIAL-NUMBER-SHEEP: The initial size of sheep population INITIAL-NUMBER-WOLVES: The initial size of wolf population SHEEP-GAIN-FROM-FOOD: The amount of energy sheep get for every grass patch eaten (Note this is not used in the sheep-wolves model version) WOLF-GAIN-FROM-FOOD: The amount of energy wolves get for every sheep eaten SHEEP-REPRODUCE: The probability of a sheep reproducing at each time step WOLF-REPRODUCE: The probability of a wolf reproducing at each time step GRASS-REGROWTH-TIME: How long it takes for grass to regrow once it is eaten (Note this is not used in the sheep-wolves model version) SHOW-ENERGY?: Whether or not to show the energy of each animal as a number

Notes: - one unit of energy is deducted for every step a wolf takes - when running the sheep-wolves-grass model version, one unit of energy is deducted for every step a sheep takes

There are three monitors to show the populations of the wolves, sheep and grass and a populations plot to display the population values over time.

If there are no wolves left and too many sheep, the model run stops.

THINGS TO NOTICE

When running the sheep-wolves model variation, watch as the sheep and wolf populations fluctuate. Notice that increases and decreases in the sizes of each population are related. In what way are they related? What eventually happens?

In the sheep-wolves-grass model variation, notice the green line added to the population plot representing fluctuations in the amount of grass. How do the sizes of the three populations appear to relate now? What is the explanation for this?

Why do you suppose that some variations of the model might be stable while others are not?

THINGS TO TRY

Try adjusting the parameters under various settings. How sensitive is the stability of the model to the particular parameters?

Can you find any parameters that generate a stable ecosystem in the sheep-wolves model variation?

Try running the sheep-wolves-grass model variation, but setting INITIAL-NUMBER-WOLVES to 0. This gives a stable ecosystem with only sheep and grass. Why might this be stable while the variation with only sheep and wolves is not?

Notice that under stable settings, the populations tend to fluctuate at a predictable pace. Can you find any parameters that will speed this up or slow it down?

EXTENDING THE MODEL

There are a number ways to alter the model so that it will be stable with only wolves and sheep (no grass). Some will require new elements to be coded in or existing behaviors to be changed. Can you develop such a version?

Try changing the reproduction rules – for example, what would happen if reproduction depended on energy rather than being determined by a fixed probability?

Can you modify the model so the sheep will flock?

Can you modify the model so that wolves actively chase sheep?

NETLOGO FEATURES

Note the use of breeds to model two different kinds of "turtles": wolves and sheep. Note the use of patches to model grass.

Note use of the ONE-OF agentset reporter to select a random sheep to be eaten by a wolf.

RELATED MODELS

Look at Rabbits Grass Weeds for another model of interacting populations with different rules.

CREDITS AND REFERENCES

Wilensky, U. & Reisman, K. (1998). Connected Science: Learning Biology through Constructing and Testing Computational Theories – an Embodied Modeling Approach. International Journal of Complex Systems, M. 234, pp. 1 - 12. (The Wolf-Sheep-Predation model is a slightly extended version of the model described in the paper.)

Wilensky, U. & Reisman, K. (2006). Thinking like a Wolf, a Sheep or a Firefly: Learning Biology through Constructing and Testing Computational Theories – an Embodied Modeling Approach. Cognition & Instruction, 24(2), pp. 171-209. http://ccl.northwestern.edu/papers/wolfsheep.pdf .

Wilensky, U., & Rand, W. (2015). An introduction to agent-based modeling: Modeling natural, social and engineered complex systems with NetLogo. Cambridge, MA: MIT Press.

Lotka, A. J. (1925). Elements of physical biology. New York: Dover.

Volterra, V. (1926, October 16). Fluctuations in the abundance of a species considered mathematically. Nature, 118, 558â€“560.

Gause, G. F. (1934). The struggle for existence. Baltimore: Williams & Wilkins.

HOW TO CITE

If you mention this model or the NetLogo software in a publication, we ask that you include the citations below.

For the model itself:

Figure 1: CC BY-NC-SA 3.0

- Wilensky, U. (1997). NetLogo Wolf Sheep Predation model. http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Please cite the NetLogo software as:

- Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

COPYRIGHT AND LICENSE

This model was created as part of the project: CONNECTED MATHEMATICS: MAKING SENSE OF COMPLEX PHENOMENA THROUGH BUILDING OBJECT-BASED PARALLEL MODELS (OBPML). The project gratefully acknowledges the support of the National Science Foundation (Applications of Advanced Technologies Program) – grant numbers RED #9552950 and REC #9632612.

This model was converted to NetLogo as part of the projects: PARTICIPATORY SIMULATIONS: NETWORK-BASED DESIGN FOR SYSTEMS LEARNING IN CLASSROOMS and/or INTEGRATED SIMULATION AND MODELING ENVIRONMENT. The project gratefully acknowledges the support of the National Science Foundation (REPP & ROLE programs) – grant numbers REC #9814682 and REC-0126227. Converted from StarLogoT to NetLogo, 2000.

Behavior Space Experiments

wolf-sheep-experiment

Definitions

Setup	Go	Final	Time limit	Metrics	Measure each tick	repetitions	exit condition
setup	go	ca	500	count sheep; count wolves; grass	true	10	not any? turtles

Parameters

Parameter	Type	Values
show-energy?	constant	false
wolf-reproduce	enumerated	3, 5, 7
initial-number-wolves	constant	50
initial-number-sheep	constant	100
model-version	enumerated	“sheep-wolves”, “sheep-wolves-grass”
sheep-gain-from-food	constant	4
grass-regrowth-time	constant	30
sheep-reproduce	enumerated	3, 4, 5
wolf-gain-from-food	sequence	9from: 10; to: 30; by: 5