Modeling Complex Software Systems Assignment 2

Lien ZHOU 1097463
Fanyu Meng 1026364
Zixin Ye 1101188

Total word count: 1168

Daisyworld Model Overview

Daisyworld is a computer simulation that explores the Gaia hypothesis. It focuses on the interaction between living and nonliving components. In the Daisyworld model, daisies act as the living components and temperature the non-living components.

There are 2 types of daisies in Daisyworld, black and white, with different albedo. White daisy has a higher albedo, which reflects more light, so that absorbed energy is relatively lower, causing the area to cool down. Black daisy however has a lower albedo, hence it absorbs more energy, causing the area to heat up. Both types of daisies have the same life span, and for a daisy to reproduce a certain temperature range must be met. A daisy will die once it reaches its life span.

When the absorbed energy by the ground is lower than the optimal growth temperature, the local temperature where the black daisies grow is higher because of the lower albedo. Since the local temperature is closer to the optimal growth temperature, black daisies will propagate better than the white daisies, where local temperature is lower. As the black daisies increase its population, the overall temperature will increase putting it beyond the optimal growth temperature. And the white daisies gain advantage. Ultimately this creates a feedback loop, leading to self regulation in the suitable temperature range. It is possible for the temperature to be too extreme so that none of the daisies could survive.

This project could be used as a subroutine for a larger program to generate results of the Daisyworld simulation. Or a student wants further understanding of ABM in a Java implementation.

The design of NetLogo Daisyworld

In NetLogo the agents are represented by patches and daisies. A patch’s state is determined by 2 components, temperature and daisy which could be black, white or empty (surface). Temperature in a patch represents the local temperature, and the average temperature of all patches represents the global temperature.

Temperature is calculated in each state and is determined by the absorbed energy using the following formula, and not dependent on the previous state.

temperature = (72*ln(absorbed-luminosity) + 80) (degree)

And the absorbed-luminosity is determined by the albedo and the solar-luminosity using the following formula.

absorbed-luminosity = (1 - albedo) * solar-luminosity

The albedos and solar-luminosity is adjustable. By default, surface albedo is 0.4, black daisy 0.25 and white daisy 0.75. Once updated, Once updated, each patch then dissipates 50% of its energy to the neighboring patches.

For each daisy, the reproduction/seeding possibility is determined by the following formula.

Possibility = (0.1457 * Temperature) – (0.0032 * Temperature^2) – 0.6443

The monte-carlo method is applied to determine if the seeding is successful. Once successful, a daisy of its kind will spawn in the empty (surface) neighboring patch. A daisy has a lifespan of 25, each iteration adds 1, and when it exceeds its life span its patch becomes empty.

The design of our Java implementation

In our system design, we would have only two classes, which are Patch and DaisyWorld.

Patch class would take the responsibility to save all related information on each patch. It would have attributes include:

- patch-number (to identify patches)
- An array of 8 Patch variables (its neighbors)*
- A daisy variable (0 for none, 1 for black, 2 for white)
- albedo (changed by daisy)
- age (identify current daisy age and trigger the seeding)
- solar-luminosity (the luminosity of the sun * （1-albedo reflect）+ neighbors diffusion)
- temperature (calculated by old temperature and current solar-luminosity)

And the DaisyWorld class is going to hold the system default value and work as the main class of this system.

It would have attributes (variables) include:

- start white.
- start black.
- albedo of white
- albedo of black
- scenario
- solar-luminosity
- albedo-of-surface

The working flow of our system to initial states would be:

1. After getting all the input value, create and give the default solar luminosity, ground albedo, 0 age to all the patches.
2. Calculate and check the temperature, and then plant daisy into these patches, which means to fill in the daisy variable and change the albedo variable according to what kind of daisy it is. The number of daisies would be an input.

3. Load all neighbor patch objects data into variables.
4. After plant daisies and setting the neighbor, we could calculate the actual temperature of every patch.

After step 4, the initial state should be settled.

When the system starts:

1. Check the blank space temperature and his neighbors. Then decide what new daisy should be planted there.
2. Check the daisy's age.
3. Load neighbors
4. Recalculate the temperature with new solar-luminate, albedo and diffuse.
5. Survive or die for the daisy and change the variables.

For the output of our system, we would create a CSV file and fill cells with daisy and temperature on that patch.

Experiment and result

In our project, we plan to run five experiments based on the five different scenarios: ramp-up-ramp-down, maintain current luminosity, low solar luminosity , our solar luminosity and high solar luminosity respectively. The starting percentage of white and black daisies will be both set to twenty percent. The albedo of white daisies will be set to 0.75 and the albedo of black daisies will be set to 0.25. The diffusion rate will be fixed at 0.5. The purpose of these experiments is to ensure our system is built as similar as the original system. All expected results will be listed below.

On ramp up ramp down scenario, the luminosity will be set at 0.8 before the simulation starts. Once the simulation starts, the luminosity will increase to 1.8 from 0.8 and stay at 1.8 for a while. The luminosity will then drop down and stay at 1.18. Under these conditions, the global temperature will increase to fifty-four degrees and stay stable eventually. In terms of the population of these two species of daisies, all black daisies will die out finally. Similarly, all white daisies will die out too finally. In a maintain-current-luminosity scenario, the solar luminosity will be fixed at 0.8 and will never be changed during simulation. The temperature will stay around 40 degrees. As for the population, the population of black daisies will fluctuate between 500 and 800. However, the white daisies will quickly die out. In low solar luminosity scenarios, the solar luminosity will be fixed at 0.6 and will never be changed during simulation. The temperature will stay at around 22 degrees and only black daisies will survive eventually. In our solar luminosity scenarios, the solar luminosity will be fixed at 1 and will never be changed during simulation. The temperature will stay at around 21 degrees and both species of daisies will survive and each of them will occupy half of the ground. In high solar luminosity scenarios, the solar luminosity will be fixed at 1.4 and will never be changed during simulation. The temperature will flacture between 5 degrees and 15 degrees. Only white daisies will survive in this case.

Appendix

Proposal

Lien Zhou: Daisyworld Model Overview & The design of NetLogo Daisyworld
Zixin Ye: The design of our Java implementation
Fanyu Meng: Experiment and result

Task breakdown

Lien Zhou: Working on any functions and classes related to daisy.

Zixin Ye: Working on any functions and classes related to patches of the ground.

Fanyu Meng: Working on any functions and classes related to simulation initialisation and input/output of simulation.

This breakdown is only our suggested plan and we will adjust accordingly if someone is struggling with their allocated tasks.

Reference List

- Novak, M. and Wilensky, U. (2006). NetLogo Daisyworld model. http://ccl.northwestern.edu/netlogo/models/Daisyworld. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- NetLogo. *NetLogo Help: diffuse.* http://ccl.northwestern.edu/netlogo/docs/dict/diffuse.html.