Plant Population - Cell DEVS Model

Author: Owen Petersen (101233850), Saja Fawagreh (101217326)

Course Name: SYSC 5104 - Methodologies for Discrete Event Modelling and Simulation

Instructor: Dr. Gabriel Wainer

Date Submitted: 2025-04-26

GitHub URL: https://github.com/SajaFawagreh/PlantPopulation

1. Introduction

For our term project, we extended the work from Assignment 2 by building a more realistic and adaptable simulation of plant population dynamics using the Cell-DEVS formalism. Our motivation was to better understand how plants grow, compete, and adapt by simulating real-world conditions using real-world data.

To do that, we will QGIS to generate the simulation's input directly from a real-world map and visualize the model's output on the same spatial layout. To make this possible, we first enhanced the model to support new environmental factors and refined its rules to more accurately reflect ecological behaviour. We then customized an existing QGIS plugin to extract the required data and format it for our simulation.

2. Background

This model was originally implemented in CD++ by Sivaharan Thurairasa, Christy Gnanapragasam, and Vijay Mahendren in a previous year. It was based on the paper ***Simulation of Vegetable Population Dynamics Based on Cellular Automata*** by Stefania Bandini and Giulio Pavesi at the University of Milan-Bicocca. We re-implemented it in Cadmium V2 for our Assignment 2 while preserving the core idea, which is that the model simulates tree population dynamics in a spatial environment.

2.1 Overview of the Paper

This paper explores the dynamics of tree population growth and competition within an ecosystem. Environmental factors such as sunlight, water, and soil nutrients significantly impact the development of tree populations. The model represents the competitive nature of trees for resources, which influences their survival, reproduction, and overall distribution.

The paper presents a two-dimensional discrete Cellular Automata (CA) model designed to simulate the evolution of a heterogeneous tree population, similar to real forest ecosystems. In this model, each CA cell represents a specific portion of land containing multiple resources. If conditions are favourable, a cell can support a tree; otherwise, it remains empty. Trees grow, survive, reproduce, or die depending on resource availability. Trees growing close to each other compete for resources, potentially limiting their growth. The model also accounts for external environmental factors such as rainfall and animal interactions, which can influence resource distribution and tree survival.

Cell Representation

Each cell in the automaton represents a square portion of land with predefined dimensions. The information stored within a cell includes:

- Presence of a tree, including its size and height
- Available resources, such as sunlight, water, and fertilizers (Nitrogen and Potassium)

Update Rule for Cellular Automata

At each simulation step, trees absorb resources from their respective cells to sustain themselves, grow, and produce seeds. If excess resources are available, the tree stores them for future use. If resources are insufficient, the tree utilizes its stored reserves. If neither stored nor external resources are sufficient, the tree dies. Typically, a cell hosting a tree has fewer resources than neighbouring cells due to local consumption.

Tree Sustenance

A tree within a cell extracts a fixed amount of each available resource based on its species-specific requirements. Larger trees require more resources, which can intensify competition among neighbouring trees.

Tree Reproduction

A cell can contain either one tree or remain empty. Trees produce seeds, which may be dispersed within the same cell or neighbouring cells. Successful germination depends on resource availability and environmental conditions.

Resource Production and Flow

Each cell produces resources up to a maximum threshold. Resource distribution is not uniform; it follows a flow pattern where resources move from richer cells to poorer ones. This mechanism allows trees in resource-deficient areas to draw from neighbouring cells, promoting competition and variation in tree growth patterns.

This model provides insight into the complex interactions of tree populations in dynamic environments, allowing researchers to explore ecological behaviours such as competition, regeneration, and external environmental influences on forests. [1]

2.2 Original Model from Assignment 2

In Assignment 2, we implemented a Cell-DEVS model to simulate tree population dynamics. Each cell stored a tree's type and height, as well as available resources: **water**, **sunlight**, **nitrogen**, and **potassium**. The simulation followed rules for how trees survive, grow, and interact with neighbouring cells, all within a symmetric 10x10 grid using a Von Neumann neighbourhood.

The model simulated three types of trees: **Locust**, **Pine**, and **Oak**, each with different growth characteristics. Trees behaved according to the following rules:

- **Survive**, **grow**, or **die** based on available resources
- **Compete** for shared resources with neighboring cells
- **Reproduce** into nearby empty cells if they meet the conditions for spreading

To run the simulation, we created a custom JSON configuration file that defined the initial state of every cell, including whether a tree was present, its type and height, resource levels, and elevation. The simulation output was visualized using the Cell-DEVS Web Viewer, allowing us to observe how tree populations evolved.

3. Design Modifications for the Term Project

The objective of the term project was to expand the original plant population simulation by integrating real-world environmental data and introducing new ecological features. Our goal was to make the model behave more realistically by incorporating factors such as soil type, elevation, and water distribution, while also enabling the use of spatial data extracted from maps. These enhancements improved the accuracy, flexibility, and interpretability of the simulation.

3.1.1 QGIS Integration

Note: This was our initial plan for the project; however, some QGIS plugin features were removed due to time constraints. Please see Section 7 for details on these changes.

QGIS Overview

QGIS (Quantum Geographic Information System) is a free and open-source software used for viewing, editing, and analyzing geospatial data. QGIS provides tools to extract information from maps, apply spatial analysis, and visualize geographic layers with precision.

Role of QGIS in the Project

In our project, we will use QGIS to extract real-world features directly from map data. This allows us to automatically generate the input configuration for the simulation based on actual environmental conditions, rather than assigning values manually. After running the simulation, the output will be visualized on the map, providing an intuitive way to observe and interpret the results within the same spatial context.

Generating the Simulation's Input Configuration File

To achieve this integration, we will customize an existing QGIS plugin originally developed as part of this Capstone project. We will adapt it to extract the environmental data required by our simulation and use it to generate the input configuration file. Using this plugin, users will be able to select any region on the map and manually place tree seeds for each species, each starting with a height of zero. For every coordinate within the selected area, the plugin will extract the following information and create the configuration file needed as input to the simulator:

- Initial water level
- Initial sunlight level
- Tree type
- Soil type
- Eevation

Visualizing Simulation Output on the Map

The plugin will also support loading the simulation output directly onto the selected map area. After the simulation runs, a .csv file will be automatically loaded into QGIS by the plugin. This will allow users to easily observe how the tree population evolves within the real-world landscape, making spatial patterns and ecological dynamics easier to interpret.

3.1.2 Simulation Improvements

To improve realism and better reflect natural ecological processes, we will extend the original simulation with the following features:

- **Soil type compatibility**: Each tree species now grows only in specific soil types - for example, *Locust* in dry soil, *Pine* in both, and *Oak* in clay. This introduces realistic environmental constraints into the simulation.
- **Elevation-based seed spreading**: Seeds are more likely to spread to lower elevation cells, making elevation a key factor in tree propagation.
- **Water regions**: Certain grid areas act as lakes. These cells continuously provide water to their neighbours but cannot host any tree growth themselves.
- **Enhanced seeding rules**: Trees can only spread if both elevation and soil conditions are suitable, resulting in more selective and realistic growth behaviour.
- **Species-specific behaviour**: Each tree species now behaves differently, reflecting unique growth and survival characteristics, and contributing to more diverse and dynamic simulation outcomes.

4. Design Specification of the Plant Population

The model was updated to support new inputs and rules that reflect more realistic environmental interactions. Initial values, such as environmental features and tree properties, are now derived from spatial data using QGIS. The update rules were extended to support the new environmental features and ensure more realistic growth and interaction dynamics.

4.1 Initial Values

The configuration JSON file used to initialize the simulation is generated using QGIS. This file is then fed into the model, and the simulation output is visualized directly on the map. The config file will contain the following information, representing the initial values of each cell in the model:

- Tree existence

  Indicates whether a tree is present in the cell. A value of 0 means no tree, while 1, 2, and 3 represent Locust, Pine, and Oak, respectively. A value of 4 indicates a water cell, which does not contain a tree but is included to support visualization in the simulation. Tree placement is determined manually using QGIS based on the selected map area.

- Tree height

  All trees start with a height of zero, as no growth has occurred at the beginning of the simulation.

- Soil Type

  Specifies the type of soil in each cell. 0 = dry soil, 1 = clay soil, and 2 = water (used for lakes or rivers). This value is extracted from the map using QGIS.

- Elevation

  Represents the vertical height of the cell from sea level, which is used in seed spread logic. Elevation data is extracted using QGIS.

- Resource level

  Initial amounts of each resource in a cell, retrieved from the map (Except for the fertilizers, which start at 0 - cells begin producing them during the simulation):

  - Water
  - Sunlight
  - Nitrogen
  - Potassium

4.2 Rules

Rules define how the plant population evolves by updating the state of each cell at every simulation step. These rules govern resource consumption, diffusion, and tree growth, ensuring that the model accurately represents environmental interactions.

- Update-Water Rule
  - The water level in a cell after an update depends on three factors: the water stored in neighbouring cells, the amount produced by the cell itself, and the amount already stored in the cell. Water diffuses from areas of higher concentration to lower concentration. This diffusion process is modelled using the following equation:

$$W'_{0,0} = 0.25 * [(W_{1,0} - W_{0,0}) + (W_{0,1} - W_{0,0}) + (W_{-1,0} - W_{0,0}) + (W_{0,-1} - W_{0,0})] + W_P + W_{0,0}$$

The updated water value cannot exceed the maximum allowed for the cell, any excess is discarded. From this new value, water is then consumed depending on whether the tree survives or grows, based on its specific resource requirements. If the tree dies, the water remains in the cell and becomes available for neighbouring cells to absorb.

- Update-Sunlight Rule
  - This rule follows the same process as the Update-Nitrogen Rule but applies to the sunlight resource instead.
- Update-Potassium Rule
  - This rule follows the same process as the Update-Nitrogen Rule but applies to the potassium resource instead.
- Update-Nitrogen Rule
  - This rule follows the same process as the Update-Water Rule but applies to the nitrogen resource instead.
- Update-Height Rule
  - If a cell contains a tree, it will grow as long as sufficient resources are available, based on its species-specific growth requirements. Each tree type has a maximum height it can reach, after which it stops growing even if resources remain sufficient. If the available resources fall below the minimum required to survive, the tree dies immediately, and its height is reset to zero.
- Reproduction Rule
  - If a cell does not have a tree and there is at least one neighbouring tree of sufficient height (based on species), the tree may attempt to spread into the empty cell. If multiple neighboring trees try to spread into the same cell, the following priority rules are applied:
    - Soil compatibility: Checks first if the tree can grow in the soil of the target cell.
    - Elevation: If multiple trees can grow, the one from the lowest elevation has priority.
    - Species priority: If multiple trees are at the same elevation, the following order is used: Locust → Pine → Oak (from lowest to highest priority).

Note: All update rules apply to every cell except those designated as water. Water cells, which represent lakes or rivers, do not support tree growth and only apply the Update-Water Rule, where they act as persistent water sources and share water with neighbouring cells.

5. Model Structure and Coupling Scheme

This section defines the structure of the atomic and coupled models in the Plant Population System, following the Cell-DEVS modelling framework. The system consists of one atomic model and one coupled model.

5.1 Atomic Cell-Devs

$$IDC = \langle X,\ Y, S, \theta,\ I, N, d, \tau, \delta_{int}, \delta_{ext}, \lambda, D \rangle$$

- X (Input Set):

  The input set **X** is represented as a vector containing the following values:

  1. **water** (*int*): Amount of water available in the neighbouring cell
  2. **sunlight** (*int*): Amount of sunlight available in the neighbouring cell
  3. **potassium** (*int*): Amount of potassium available in the neighbouring cell
  4. **nitrogen** (*int*): Amount of nitrogen available in the neighbouring cell
  5. **height** (*int*): Tree height of the neighbouring cell
  6. **soil_type (int):** Type of soil in the neighbouring cell
     a. **0** – Dry
     b. **1** – Clay
     c. **2** – Water
  7. **elevation (int):** Elevation value of the neighbouring cell
  8. **tree_type** (*int*): Indicates the presence and type of tree in the neighbouring cell
     a. **0** – No tree
     b. **1** – Locust
     c. **2** – Pine
     d. **3** – Oak
     e. **4** – Water
- Y (Output Set):
  1. **water** (*int*): Amount of water available to share with neighbouring cells after consumption
  2. **sunlight** (*int*): Amount of sunlight available to share with neighbouring cells after consumption
  3. **potassium** (*int*): Amount of potassium in the soil available to share with neighbouring cells after consumption
  4. **nitrogen** (*int*): Amount of nitrogen in the soil available to share with neighbouring cells after consumption

5. **height** (*int*): Height of the tree in the cell (0 if no tree is present)
6. **soil_type (int):** Type of soil in the cell
    a. **0** – Dry
    b. **1** – Clay
    c. **2** – Water
7. **elevation (int):** Elevation value of the cell

8. **tree_type** (*int*): Indicates the presence and type of tree sharing the resources
    a. **0** – No tree
    b. **1** – Locust
    c. **2** – Pine
    d. **3** – Oak
    e. **4** – Water

- S (State Set):
    1. **Vector defining the amount of resources stored in the cell:**
        a. **water** (*int*)
        b. **sunlight** (*int*)
        c. **nitrogen** (*int*)
        d. **potassium** (*int*)
    2. **Vector defining the maximum amount of each resource:** Same as above
    3. **Vector defining the amount of each resource produced at each update:** Same as above
    4. **Vector defining the minimum amount of each resource required to survive:** Same as above
    5. **Vector defining the minimum amount of each resource required to grow:** Same as above
    6. **soil_type (int):** Type of soil in the cell
        a. **0** – Dry
        b. **1** – Clay
        c. **2** – Water
    7. **elevation (int):** Elevation value of the cell
    8. **tree_type** (*int*): Indicates the presence and type of tree in the cell
        a. **0** – No tree
        b. **1** – Locust

c. **2** – Pine

d. **3** – Oak

e. **4** – Water

- τ (Local Computing Function): Please refer to the updated rules in section 4.2 Rules

Note: The maximum resource capacity of a tree, the amount of each resource it produces per update, and the minimum amounts required to survive or grow are now hardcoded for each tree type in the simulation. These values can be found in the [plantSpeciesInfo.hpp]() file located in the repository. Previously, this information was pulled from the configuration file, but this approach prevented us from assigning different resource configurations to each tree type, especially when a tree reproduced into an empty cell, as it would inherit the default resource values rather than species-specific ones. Hardcoding the values per species now allows each tree type to have its distinct resource behaviour.

5.2 Coupled Cell-Devs

$$CM = \langle I, X, Y, Xlist, Ylist, \eta, N, \{m,\ n\}, C, B, Z, select\rangle$$

- I (Input Interface): $< P_x, P_y >=< \emptyset,\ \emptyset >$
- X (Set of Input Values): same as X (Input Set) in 5.1 Atomic Cell-Devs
- Y (Set of Output Values): same as Y (Output Set) in 5.1 Atomic Cell-Devs
- Xlist (list of external inputs): Ø
- Ylist (list of external outputs): Ø
- n (Neighbourhood Size): 5 (Von Neumann Neighbourhood)
- N (Relative Positions of Neighboring Cells): {(0,0),(−1,0),(1,0),(0,1),(0,−1)}
- {m, n} (Grid Size {row, column}): {10, 10}
- C (Set of All Cells): $\{C_{ij}\ /\ i\ \in\ [1, 10],\ j\ \in\ [1, 10]\}$
- B (Border Condition): Ø
- Z (Neighbouring Cells Exchange Information):
  - $P_{ij}^{Y_1} \rightarrow P_{i,j-1}^{X_1}$ $\quad P_{i,j+1}^{Y_1} \rightarrow P_{ij}^{X_1}$
  - $P_{ij}^{Y_2} \rightarrow P_{i+1,j}^{X_2}$ $\quad P_{i-1,j}^{Y_2} \rightarrow P_{ij}^{X_2}$
  - $P_{ij}^{Y_3} \rightarrow P_{i,j+1}^{X_3}$ $\quad P_{i,j-1}^{Y_3} \rightarrow P_{ij}^{X_3}$
  - $P_{ij}^{Y_4} \rightarrow P_{i-1,j}^{X_4}$ $\quad P_{i+1,j}^{Y_4} \rightarrow P_{ij}^{X_4}$
  - $P_{ij}^{Y_5} \rightarrow P_{ij}^{X_5}$ $\quad P_{ij}^{Y_5} \rightarrow P_{ij}^{X_5}$
- Select (tie-breaking function): Ø

6. Demonstrating Simulation without QGIS

We will use the Cell-DEVS Web Viewer to showcase our simulation's full features. This includes the interaction of environmental factors like elevation, soil type, and water regions and species-specific rules for tree growth, survival, and reproduction. The viewer allows us to visualize how trees compete for resources and spread under different conditions.

Note: This demonstration uses the basic version of the model available in the repository, which includes all features described in the report.

6.1 Sample Configuration File

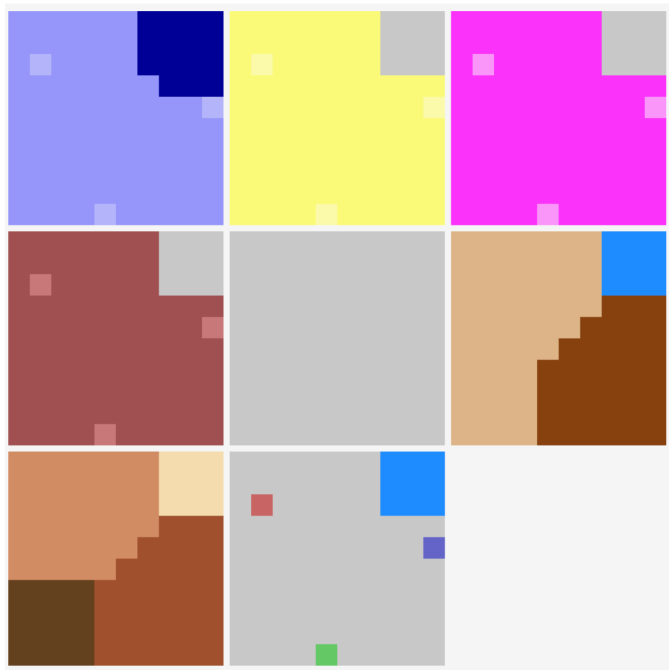
The configuration file shown here is **not generated using the QGIS plugin**. Instead, it was manually created to demonstrate all the simulation features in a controlled setup. It includes:

- **Tree placement** for Locust (1), Pine (2), and Oak (3) saplings. A value of 0 means no tree is present, and 4 is used for water cells.
- **Tree height** for each planted tree - set to 0 if no tree is present or if you want the tree to start as a seed.
- **Soil types**: dry (0), clay (1), and water (2) regions
- **Elevation values** for different parts of the grid
- **Current Resource levels**: water, sunlight, nitrogen, and potassium
- **Visualization settings** to map each of these fields to colours

You can view this simulation's full configuration JSON file, including the initial values, in our repository here.

6.2 Visual Guide to Simulation Output

The following figure represents the state of the grid at a given simulation step, using the Cell-DEVS Web Viewer output:



The grid is visualized in eight panels (left to right, top to bottom):

1. Water
2. Sunlight
3. Nitrogen
4. Potassium
5. Tree Height
6. Soil Type
7. Elevation: (Darker brown shades indicate higher elevation)
8. Tree Type

In the Soil Type panel:

- Light Brown = Dry
- Dark Brown = Clay
- Blue = Water

In the Tree Type panel:

- 🟥Red = Locust
- 🟩Green = Pine
- 🟪Purple = Oak
- 🟦Blue = Water

6.3 Simulation Video

You can watch a video of the simulation here. To view specific time steps, visit the simulation viewer website and upload the following relevant files, which are available in the repository:

- *plant_population_visualization_config.json* (found under the *config* folder)
- *grid_visualization_log.csv* (found under the *log_files* folder)

Execution Results and Model Analysis

This simulation begins with all resource levels, water, sunlight, nitrogen, and potassium, set to 0. Three trees are initially planted: one of each type (Locust, Pine, and Oak), each starting at height 0. A lake is positioned in the top-right corner, and the grid contains varying elevations and soil types. The goal is to observe how resources accumulate over time and how each tree type grows and spreads across the grid.

6.3.1 Resources

At the start of the simulation, resource levels appear low because each cell begins with zero and must produce its resources. This is why the initial colouring is light rather than empty. As the simulation progresses, resource levels gradually increase, but then quickly drop as the trees begin using the available resources. At first, trees only have enough to survive and not to grow.

Once the trees begin growing, they consume more resources. As they start to reproduce, you'll notice that the overall resource levels across the grid drop more quickly, since more trees are competing for the same limited supply. Eventually, the levels begin to rise again. This happens when trees reach their maximum height, so they no longer use extra resources to grow and only consume what is necessary to survive. Because trees in our model are designed to produce more resources than they need to survive, the levels eventually stabilize at their maximum.

If you observe the video closely, you'll also notice that resource levels drop less for Locust than for Pine or Oak. This is because Locust requires fewer resources to survive and grow compared to Pine and Oak. You'll also see that Locust reaches its maximum height faster, followed by Pine, then Oak. This is due to differences in how much each tree species produces - Locust produces the most, followed by Pine, and then Oak.

6.3.2 Water Feature

At the beginning of the simulation, you will notice that the lake has the highest water level. This is because it starts with more water than the surrounding cells. The other resources, such as sunlight, nitrogen, and potassium, appear grey, indicating a value of zero, and they stay at zero throughout the simulation since the lake only stores and shares water.

As the simulation progresses, especially in the early steps, the lake quickly distributes water to its neighbouring cells, much faster than other cells. Throughout the simulation, the water level at the lake remains high because it continuously produces water while sharing it with its surroundings. When neighboring trees begin using this water, the water's level drops temporarily but starts increasing again. Eventually, water levels across the grid rise as trees stop consuming water to grow and only use what they need to survive.

6.3.3 Tree Height

At the beginning of the simulation, the entire tree height panel appears grey, indicating that all trees start with a height of 0. Shortly after, light green patches appear at the positions where trees were initially planted. This marks the beginning of tree growth, as each tree starts producing resources and receiving additional support from neighboring cells.

As the simulation progresses, trees grow depending on both available resources and species-specific requirements. Locust trees are the first to reproduce and grow in new cells. This is because they require fewer resources to grow and a lower height threshold to reproduce compared to the other species. Pine trees come next, and Oak trees take the longest to grow and spread, since they have the highest height and resource requirements. The growth order is Locust, then Pine, then Oak.

Once reproduction begins, the height values in each cell continue to increase. The darker the green, the taller the tree in that cell. Locust trees spread and grow more quickly than Pine and Oak due to their lower requirements, while Pine trees spread faster than Oak for the same reason.

By the end of the simulation, the height panel displays a gradient of greens: cells with Locust trees appear lighter, followed by Pine, and then Oak. This reflects the fact that the maximum height defined in the model is lowest for Locust, higher for Pine, and highest for Oak.

6.3.4 Soil Type

You will notice that Locust trees only grew in dry soil, Oak in clay soil, and Pine in both, since each species only reproduces in its compatible soil type. You will also notice that none of the trees grew in the lake area, as it is only meant to act as a high-water resource zone for nearby cells and does not support tree growth.

6.3.5 Elevation

For elevation, a higher elevation area was placed in the lower-left part of the grid to observe how Pine and Locust trees would respond. Since Locust produces resources faster, it was placed slightly farther away from the elevated area. However, Pine was positioned closer, so it reached the elevated region first. Because the elevation difference between Pine and that area was smaller than it was for Locust, Pine had higher priority when spreading. As a result, Pine occupied more of the elevated area than Locust.

6.3.6 Tree Type

At the beginning of the simulation, there are three trees on the grid, one of each type: Locust, Pine, and Oak, and a lake, shown in blue, where no trees can grow. As the simulation progresses, Locust begins to spread into nearby empty cells, followed closely by Pine, while Oak takes longer to begin spreading.

By the end of the simulation, both Locust and Pine have expanded across much of the grid, while Oak remains limited to fewer areas. This is because Locust reproduces faster, and Pine can grow in more types of soil. Since a cell can only be occupied by one tree type, species that spread earlier or into more environments have a greater chance of dominating the space.

7. Demonstrating Simulation with QGIS

This section explains how the simulation is integrated with QGIS to extract real-world environmental data and generate the input configuration. While we were able to extract all necessary data from maps using QGIS, run the simulation, and successfully produce the output .csv file, we were not able to get the QGIS visualizer to work due to an issue with compiling the simulator executable for a Windows environment.

As part of the QGIS integration, we re-implemented our model using asymmetric Cell-DEVS instead of the symmetric version initially planned. Due to this change and time limitations, we were unable to fully integrate multiple tree types and soil type compatibility into the QGIS version.

However, everything else in the model remains the same. The only differences are:

- The coupled model was switched from **GridCellDEVSCoupled** to **AsymmetricCellDEVSCoupled**.
- The QGIS plugin does not support loading different soil types from map data
- The QGIS plugin does not support planting different types of trees (only Locust trees can be planted).

All original features are still fully functional and demonstrated in Section 6 using the Cell-DEVS Web Viewer. This section is included to show that the full model works, even though some parts were not fully integrated with QGIS.

A demonstration video of the plugin working can be found here.

Note: This demonstration uses the advanced version of the model available in the repository, which includes the integration of QGIS.

7.1 Sample Configuration File

The configuration file shown here is **generated using the QGIS plugin**. It reflects real-world environmental data extracted from a selected map area and includes:

- **Tree placement**: 1 for Locust if a tree is placed, 0 if no tree, 4 for water cells
- **Soil types**: always 0 - dry soil by default
- **Elevation values** extracted from the map
- **Initial resource levels**:
  - **Water** and **sunlight** are extracted directly from map data
  - **Nitrogen** and **potassium** are initialized to 0, and will increase over time as cells begin producing their own resources
- **Neighbouring cell connections (asymmetric)**: defined for each coordinate

You can view a sample QGIS-generated configuration file here.

7.2 Sample Output File

This is a sample of the output file generated after running the simulation. It contains the logged state of each cell at every time step, showing how trees grow and how environmental conditions evolve.

Each row includes:

- **Current simulation time**
- The **model ID** and **cell coordinate**
- **Port**: empty or outputNeighborhood indicates if the entry shows the cell's state or shared output to neighbours.
- **A vector of values representing**:
  - Current water, sunlight, nitrogen, and potassium values
  - Tree Height
  - Soil Type: Always set to dry (0) or water (2)
  - Elevation
  - Tree type: 1 for Locust, 0 for empty, 4 for water

This output file is stored in .csv format and is automatically generated by the simulation. You can view a full sample output file here.

7.3 Generating the Config File and Running the Simulation

To use QGIS to extract map-based data, generate the configuration file, and run the simulation to obtain the output, please refer to the step-by-step instructions in the README file here for guidance.

8. Conclusion

In this project, we enhanced the original Cell-DEVS plant population model by introducing new features to make it more reflective of real-world ecosystems. We added support for soil types, elevation, and water regions to influence where trees can grow and how they spread. We also integrated QGIS to extract real environmental data from maps and generate the simulation input. Although we were unable to get the QGIS visualizer working, the simulation ran successfully and produced correct output based on real-world inputs. This demonstrates how tree populations evolve and compete depending on their environment, even with the simplified QGIS setup.

9. References

[1] S. Bandini and G. Pavesi, *"Simulation of Vegetable Population Dynamics Based on Cellular Automata,"* Dept. of Computer Science, Systems and Communications, University of Milan-Bicocca, Milan, Italy.