Simulation of Vegetable Populations Dynamics Based on Cellular Automata

Stefania Bandini and Giulio Pavesi

Dept. of Computer Science, Systems, and Communication
University of Milan–Bicocca
Milan, Italy
{bandini,pavesi}@disco.unimib.it

Abstract. Modeling the dynamics of vegetable populations is an extremely challenging problem. The evolution of a vegetable population, that is, of all the weeds, plants and trees that grow in a given area, is mainly influenced by the resources available on the territory (i.e. sunlight, water, substances present in the soil), and how the single individuals compete for them. Traditional models for this case study are continuous and based on differential equations. However, most of the data needed to provide reliable parameters for these models are usually scarce and difficult to obtain. The model we present is instead based on two–dimensional Cellular Automata, whose cells, arranged on a square grid, represent portions of a given area. Some resources are present on the area, divided among the cells. A cell can host a tree, represented in the model by a set of parameters defining its species, its size (that is, the size of its parts such as limbs, trunk, and roots), the amount of each resource it needs to survive, to grow, and/or reproduce itself (that is, produce fruits). The model has been applied to the simulation of populations consisting of robiniae (black locust), oak, and pine trees on the foothills of the italian alps, with encouraging results reproducing real conditions.

1 Introduction

Modeling the dynamics of vegetable populations, that is, of all the weeds, plants and trees living in a given area, is an extremely challenging problem [1]. The main difficulty lies in the acquisition of data for the definition of the parameters of the models, that must cover very long time periods, especially in the case of perennial plants. Such data must include the resources available on the territory, and those needed by plants to sprout, survive, grow, and reproduce themselves. In fact, the evolution of a vegetable population is mainly influenced by the resources available (i.e. sunlight, water, substances present in the soil), and how the different individuals compete for them. Traditional models are continuous and based on differential equations [2,3,4], and usually model the evolution of the system with global parameters such as the total number of trees and their overall biomass. More recently, Cellular Automata have been introduced to study this

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 202–209, 2002.

problem [5,6], but usually their application was limited to the evolution of single infesting species.

In this paper, we present a discrete model based on two–dimensional Cellular Automata, that allows to model and simulate the evolution of heterogeneous vegetable populations composed by different perennial species as in real woods and forests. The evolution of the system is thus modeled in a bottom–up fashion, that is, is the result of the interactions among single individuals and their competition for the resources available on the territory [7].

The cells of the CA represent portions of a given area. Each cell contains some resources, and if conditions are favorable, can host a tree. A tree is represented in the model by a set of parameters, defining its species, its size (that is, the size of its parts such as limbs, trunk, and roots), the amount of each resource it needs to survive, to grow, and reproduce itself (that is, produce fruits). A single tree has been "decomposed" in different parts in order to reproduce the effect of environmental influences. In fact, the environment and the resources available determine how the overall biomass of the tree is divided among the different parts composing it [8].

When a tree produces fruits, some seeds are scattered in the neighboring cells. A seedling can sprout in a cell when the latter contains a seed, no other tree, and a sufficient amount of each resource. In this case, a tree is born, and the state of the cell now comprises also all the parameters defining the tree present in it (otherwise set to zero, if no tree is present). Then, the cell has also to contain enough resources to sustain the growth of the plant. The quantity of resources needed varies according to the species of the tree and its size. When a tree starts growing, its increasing mass begins to need a larger amount of resources, that can also be taken from the neighbors of the cell where it is located. Thus, the sprouting or the growth of other trees in its proximity is negatively influenced, that is, the tree starts competing for resources with the others. Whenever a tree cannot find enough resources to survive, it dies.

The model has been applied to the simulation of populations consisting of robiniae (black locust), oak, and pine trees on the foothills of the italian alps, with encouraging results reproducing real conditions.

2 The Cellular Automaton

We now give a more formal description of the model. The state of each cell of the CA is defined by a flag denoting whether or not it contains a tree, the amount of each resource present in the cell, and a set of variables defining the features of the tree (possibly) growing in it. The update rule of the automaton mainly depends on the presence of a tree in a cell. In case a tree is present, part of the resources present in it (and in the neighboring ones, if the tree is large enough) are absorbed by the tree. Every cell also produces at each update step a given amount of each resource (that in any case cannot exceed a maximum threshold value). The production of resources in the cells is determined by a set of global parameters, and reproduces environmental factors such as rain, presence of animals in the area, and so on. The effect of the presence of a tree in a cell on the neighboring ones has been modeled by making resources flow from richer cells to poorer ones (that possess less resources since a part of them is consumed by the tree). The resources we explicitly included in the model are water, light, nitrogen, and potassium. Both von Neumann and Moore neighborhoods have been considered in the simulations.

The CA can be thus defined as:

$$\mathbf{CA} = \langle R, N, Q, f, I \rangle$$

where:

1. $R = \{(i,j)|1 \leq i \leq N, 1 \leq j \leq M\}$ is a two–dimensional $N \times M$ lattice;
2. H is the neighborhood, that can be either the von Neumann or Moore neighborhood;
3. Q is the finite set of cell state values;
4. $f : Q \times Q^{|H|} \rightarrow Q$ is the state transition function;
5. $I : R \rightarrow Q$ is the initialization function.

2.1 The Cells

Each cell of the automaton reproduces a square portion of terrain with a side ranging from three to five meters. As mentioned before, each cell contains some resources, and can host a tree. Thus, the possible states of a cell must define:

1. The type of terrain the cell reproduces;
2. The resources present in the cell;
3. The amount of resources the cell produces at each update step, and the maximum amount of resources it can contain, according to its type;
4. Whether a tree is present in the cell, or not;
5. If a tree is present:
   a) the size of the tree;
   b) the amount of each resource it needs at each update step to survive and grow;
   c) the amount of each resource stored by the tree at previous update steps;
6. Seeds scattered by trees living in the area.

If we assume that k types of resource and l different tree species are present in the area, the finite set of states Q can be defined as follows:

$$Q = \{\mathbf{R}, \mathbf{M}, \mathbf{P}, T, \mathbf{Z}_T, \mathbf{N}_T, \mathbf{U}_T^G, \mathbf{U}_T^S, \mathbf{R}_T, \mathbf{M}_T, \mathbf{G}_T, \mathbf{S}\}$$

where:

1. $\mathbf{R} = \{r_1, \ldots, r_k\}$ is a vector defining the amount of each resource present in the cell;
2. $\mathbf{M} = \{m_1, \ldots, m_k\}$ is the maximum amount of each resource that can be contained by the cell;

3. $\mathbf{P} = \{p_1, \ldots, p_k\}$ is the amount of each resource produced by the cell at each update step;
4. T is a flag indicating whether a tree is present in the cell or not;
5. $\mathbf{Z}_T = \{z_T^r, z_T^t, z_T^l, z_T^f\}$ is a vector defining the size of the different parts of the tree (in our model, roots, trunk, leaves, and fruits);
6. $\mathbf{N}_T = \{n_T[1], \ldots, n_T[k]\}$ are the amounts of each resource the tree takes from the cell at each update step (depending on its size);
7. $\mathbf{U}_T^G = \{u_T^G[1], \ldots, u_T^G[k]\}$ is the vector defining the amount of each resource needed at each update step by the tree to *grow*;
8. $\mathbf{U}_T^S = \{u_T^S[1], \ldots, u_T^S[k]\}$ is a vector defining the minimum amount of each resource the tree needs at each update step to *survive*; for each i, $1 \leq i \leq k$, we have $u_T^S[i] < u_T^G[i] < n_T[i]$;
9. $\mathbf{R}_T = \{r_T[1], \ldots, r_T[k]\}$ is the amount of each resource stored by the tree at previous update steps;
10. $\mathbf{M}_T$ is a vector of threshold values for different parameters defining the tree, such as maximum size, maximum age, minimum age for reproduction, maximum number of seeds produced for each mass unity of fruits, and so on. These threshold values can be fixed or picked at random in a given range when a new tree is created;
11. $\mathbf{G}_T = \{g_T^r, g_T^t, g_T^l, g_T^f\}$ is a vector defining the *growth rate* of each of the parts of the tree, that is, how much each part of the tree grows when enough resources are available;
12. $\mathbf{S} = \{s_1, \ldots, s_l\}$ is a vector defining the number of seeds present in the cell for each of the l species growing in the territory.

2.2 The Update Rule

At each update step of the automaton, the tree present in each cell (if any) takes the resources it needs from the cell itself and uses them to survive, grow (if enough resources are available), and produce seeds. If the resources available in the cell exceed its needs, the tree stores some resources. Conversely, if the resources available in the cell are not sufficient, the tree uses resources stored at previous update steps. If also the resources stored are not sufficient for the tree to survive, the tree dies. A newborn plant can sprout in a vacant cell, if the latter contains a seed of its species, and again enough resources.

Moreover, we defined the update rule in order to reproduce the increasing influence that a growing tree can have on neighboring cells. For example, its roots can extend beyond the limits of the cell hosting it. Or, when it gets taller, it shades an increasingly wider area around itself, thus having a negative influence on the growth of other trees in its neighborhood. We modeled the impact of a tree in a given position on its neighborhood by making resources flow from richer cells to poorer ones. In other words, a cell hosting a large tree is poor on resources, since the tree at each update step takes most (or all) of them. If the neighboring cells are vacant, their resources remain unused, and thus are richer than the one hosting the tree. Therefore, if we let resources flow from richer cells to poorer neighbors, the effect is that in practice a large tree starts to collect resources also from neighboring cells. Notice that if we include sunlight among the resources contained by a cell, we can model in this way also the “shade” effect. Seeds are also introduced in the model as a resource that moves from cell to cell. Thus, a tree can scatter its seeds in the neighboring cells.

Now, let $C(i,j)$ be the cell located at position (i,j) in the lattice. With $\mathbf{R}(i,j)$ we will denote the resource vector of cell $C(i,j)$, with $\mathbf{M}(i,j)$ the maximum resource values, and so on. The transition function can be divided in four sub–steps, defined as follows.

Tree sustenance. If a tree is present in cell $C(i,j)$, it takes from it a given quantity (defined by $\mathbf{N}_T(i,j)$) of each available resource $\mathbf{R}(i,j)$. If, for some resource i, the amount available $r_i(i,j)$ is lower than the corresponding value in $\mathbf{N}_T(i,j)$, then the tree takes the whole quantity $r_i(i,j)$. The amount of resources taken depends on the size of the tree $\mathbf{Z}_T(i,j)$. Then, if enough resources (those taken at this step, plus the resources stored at previous steps), are available, as defined by vector $\mathbf{U}_T^G(i,j)$, the tree grows, that is, each part grows according to the growth rate vector $\mathbf{G}_T(i,j)$ associated with the tree. Else, the resources might be just sufficient for the tree to survive (vector $\mathbf{U}_T^S(i,j)$). In this case, the tree parameters are left unchanged. In both cases, the tree “burns” an amount of each resource, as defined by vector $\mathbf{U}_T^G(i,j)$ or $\mathbf{U}_T^S(i,j)$. All the unused resources collected at this step are stored and added to vector $\mathbf{R}_T(i,j)$. Otherwise, if the overall amount (stored plus collected) of at least one resource is under the “survival threshold” of the tree, the latter dies. The tree also dies when it reaches its maximum age defined in vector $\mathbf{M}_T(i,j)$.

Tree reproduction. We have two cases to consider: a tree is present in the cell, or the cell is empty. In the former case, the tree may produce some seeds (if it is old enough, and according to the size of its fruits $z_T^f(i,j)$), that are used to update the corresponding variable in the seed vector $\mathbf{S}(i,j)$. Also, new trees cannot sprout from the seeds contained in the cell if a tree is already present. Instead, the cell can be vacant and contain some seeds. If the resources present in the cell are sufficient (quantities defined as global parameters for each tree species) a new tree is born. If seeds from different species are present in the cell, the winning species is chosen at random, with probability proportional to the number of its seeds.

Resource production. In the third sub–step, each cell produces a given amount of resources, according to its production vector $\mathbf{P}(i,j)$. In any case, the amount of each resource contained in the cell cannot exceed the corresponding maximum value defined by vector $\mathbf{M}(i,j)$.

Resource flow. In this step, resources are balanced among neighboring cells, in order to let resources flow from richer to poorer cells. Let $r_h(i,j)$ be the amount of resource h contained by cell $C(i,j)$, and assume that we are using the von Neumann neighborhood. $r'_h(i,j)$, the amount of resource i after this update sub–step, is defined as:

$$r'_h(i,j) = \frac{r_h(i,j) + \frac{r_h(i+1,j)+r_h(i-1,j)+r_h(i,j+1)+r_h(i,j-1)}{4}}{2}$$

In other words, we can see each cell as divided in four parts, each one containing the amount $r_h(i,j)/4$ of resource h, and corresponding to one of the neighbors. The amount of resource h contained in each part is balanced with the corresponding part of the neighbors. In case we adopt the Moore neighborhood, we can imagine the cells as split into eight portions. The effect is that, if cell $C(i,j)$ is richer on resource h than its neighbors, part of its content will spill into them. As mentioned before, $r'_h(i,j)$ cannot exceed the corresponding maximum value defined for the cell $(m_h(i,j))$. In this case, we set $r'_h(i,j) = m_h(i,j)$. The same rule is applied to each of the components of the seeds vector $\mathbf{S}(i,j)$.

2.3 The Initial Configuration

The initial configuration of the CA can be defined by the user, by setting appropriate resource parameters for each cell. Also, some trees might be already present on the territory, with all the variables defining them set. Or, the territory might be empty, with some seeds scattered here and there (clearly, if no tree and no seeds are present, nothing happens when the automaton is started).

3 The User Interface

The model has been implemented in C++ under Windows NT. The user interface permits to define explicitly:

1. Different types of cell, according to the maximum amount of resources the cell can contain and the amount of resources it produces, in order to resemble the features of different types of terrain. Moreover, it also possible to reproduce rivers (by setting high values for water content and production, and zero maximum content values for other resources), rocky terrain (with very low values for all the resources), roads (zero values for all the resources), and so on;
2. Different tree species according to the amount of resources needed at each update step, to the growth rate of the different parts, that is, how resources are distributed among the different parts, the quantity of seeds produced;
3. The initial configuration of the automaton.

The interface shows step–by–step the evolution of the system, giving a straightforward image of the growth of the trees. Moreover, it is possible to show the distribution of the resources on the territory at each step, and the overall results of the simulation (total number of trees, trees for each species, total biomass, biomass of each single species and single tree, and so on), as shown in Fig. 1, 2, and 3.

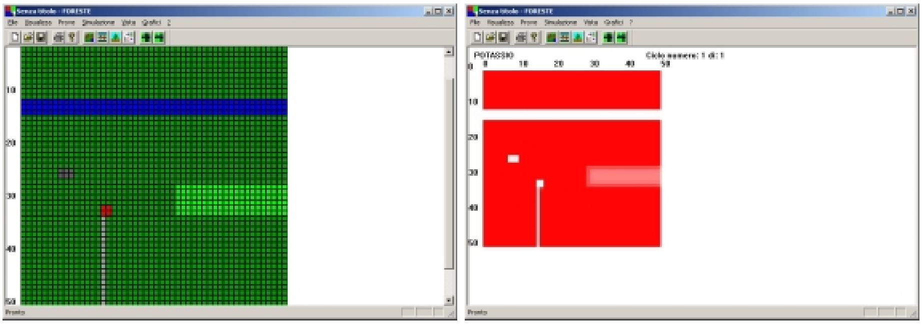

Fig. 1. An initial configuration of the automaton (left). The dark strip represents a river. The image to the right shows the initial distribution of potassium in the cells. Darker areas are richer on potassium.

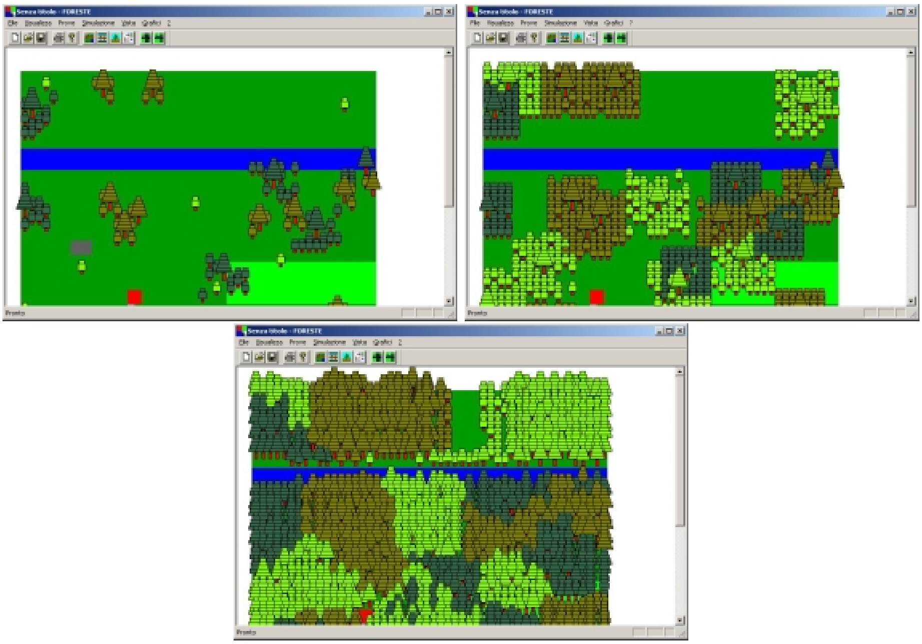

Fig. 2. Example of the user interface, showing three different stages of the evolution of a vegetable population composed by black locusts, oaks, and pine trees, starting from the initial configuration of Fig 1.

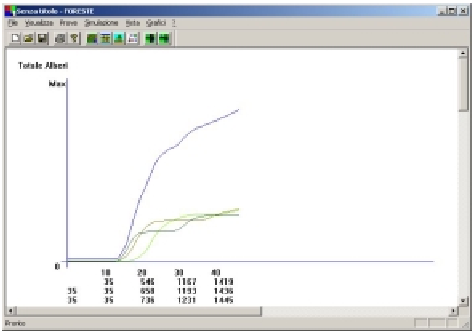

Fig. 3. The user interface showing the total number of trees, and the number of trees of each species present in the area for the example shown in Fig. 2.

4 Conclusions

In this paper we presented a model based on CA for the simulation of the dynamics of vegetable populations. Our simulations, reproducing populations of robiniae, oaks and pine trees living on the foothills of the italian alps have shown results qualitatively similar to real case studies. We believe that the flexibility of the model, that allows the user to define explicitly different types of terrain and tree species can provide an useful tool, not only for the simulation of real case studies, but also for a better comprehension of the main factors influencing the dynamics of vegetable populations.

References

1. M.G. Barbour, J.H. Burk, W.D. Pitts, F.S. Gilliam, M.W. Schwartz, Terrestrial Plant Ecology, Benjamin Cummings, 1998.
2. J.L. Uso–Domenech, Y. Villacampa–Esteve, G. Stübing–Martinez, T. Karjalainen, M.P. Ramo. MARIOLA: A Model for Calculating the Response of Mediterranean Bush Ecosystem to Climatic Variations. *Ecological Modelling*, **80**(1995), pp. 113–129.
3. Q. Zeng, X. Zeng, An Analytical Dynamic Model of Grass Field Ecosystem with Two Variables. *Ecological Modelling*, **85**(1996), pp. 187–196.
4. Q. Zeng, X. Zeng, Two Variables Dynamic Model of Grass Field Ecosystem with Seasonal Variation. *Ecological Modelling*, **85**(1996), pp. 197–202.
5. H. Baltzer, W.P. Braun, W. Köhler. Cellular Automata Models for Vegetation Dynamics. *Ecological Modelling*, **107**(1998), pp. 113–125.
6. R.L. Colasanti, J.P. Grime. Resource dynamics and vegetation processes: a deterministic model using two–dimensional cellular automata. *Functional Ecology*, **7**(1993), pp. 169–176.
7. D. Tilman. Competition and biodiversity in spatially structured habitats. *Ecology*, **75**(1994), pp. 2–16.
8. D. Tilman. Dynamics and Structures of Plant Communities. Princeton University Press, 1988.