A Model Based on Cellular Automata for the Simulation of the Dynamics of Plant Populations

Stefania Bandini [a], Giulio Pavesi[b]

[a]Dept. of Computer Science, Systems and Communication
University of Milano–Bicocca
Milan, Italy
bandini@disco.unimib.it

[b]D.I.Co.
University of Milan
Milan, Italy
pavesi@dico.unimi.it

Abstract: We present a Cellular Automata based model for the simulation of the dynamics of plant populations. The evolution of a plant population in a given area mainly depends on the resources available on the territory (in turn influenced by other factors like sunlight, rain, temperature, and so on) and how different individuals (plants) compete for them. Traditional methods used in this field are continuous and based on differential equations that model the global evolution of the system: unfortunately, 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 thought in a bottom–up fashion, and is based on a two-dimensional Cellular Automaton, whose cells, arranged on a square grid, represent portions of a given territory. Some resources are present on the area, divided among the cells. 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 resources it needs to survive, to grow, and/or reproduce itself (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 experimentally observed population trends.

Keywords: Cellular Automata; discrete models; plant population dynamics.

1 INTRODUCTION

Modeling the dynamics of plant populations living in a given area is a widely studied and extremely challenging problem, as described for example by M.G.Barbour et al. [1998] and J.Silverton and D.Charlesworth [2001]. 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 plant 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 like those introduced and employed (among many others) by J.L.Uso-Domenech et al. [1995]; Q.Zeng and X.Zeng [1996a, b]; C.Damgaard et al. [2002], 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 problem as for example by R.L.Colasanti and J.P.Grime [1993]; H.Baltzer et al. [1998], 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 the simulation of the evolution of heterogeneous plant 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, as for example discussed by D.Tilman [1994] and J.Ehrlen [2000]. In this paper, we show how 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. However, we believe that its generality and flexibility make it suitable for the simulation of different case studies and conditions, with no changes in its basic structure.

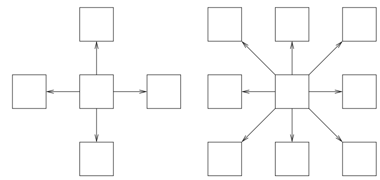

Figure 1: Von Neumann (left) and Moore (right) neighborhoods. In CA, the update of the state of a cell (the central one in the figure) depends on the state of the cell itself and on the state of cells defined as its neighbors.

2 CELLULAR AUTOMATA

A *Cellular Automaton* (CA) consists of a regular discrete lattice of *cells*, where each cell is characterized by a state belonging to a finite set of states. Each cell evolves (changes its state) according to a given *update rule*, which depends only on the state of the cell and a finite number of neighboring cells. All the cells of the automaton follow the same update rule. The automaton evolves through a sequence of discrete time steps, where all cells update their state simultaneously. For a more general introduction on CA we refer the reader to, for example, the book by E.Goles and S.Martinez [1990].

The idea of our model is that 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, as discussed by D.Tilman [1988].

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.

3 THE MODEL

We now give a more formal description of the model. The cells of the automaton are arranged on a two–dimensional square grid. The state of each cell is defined by a flag denoting whether or not it contains a tree, and a set of variables denoting the amount of each resource present in the cell, as well as 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 of a cell (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 that reproduce 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 usually contain 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 (shown in Fig. 1) have been considered in the simulations. The CA can be thus defined as:

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

where:

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

3.1 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:

- The type of terrain the cell reproduces;
- The resources present in the cell;
- The amount of resources the cell produces at each update step, and the maximum amount of resources it can contain, according to its type;
- Whether a tree is present in the cell, or not;
- If a tree is present:
  - the size of the tree;
  - the amount of each resource it needs at each update step to survive and grow;
  - the amount of each resource stored by the tree at previous update steps;
- 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:

- $\mathbf{R} = \{r_1, \ldots, r_k\}$ is a vector defining the amount of each resource present in the cell;
- $\mathbf{M} = \{m_1, \ldots, m_k\}$ is the maximum amount of each resource that can be contained by the cell;
- $\mathbf{P} = \{p_1, \ldots, p_k\}$ is the amount of each resource produced by the cell at each update step;
- T is a flag indicating whether a tree is present in the cell or not;
- $\mathbf{Z}_T = \{z_T^r, z_T^t, z_T^l, z_T^f\}$ is a vector defining the size of the different parts of the tree (in our model, roots, trunk, leaves, and fruits);
- $\mathbf{N}_T = \{n_T[1], \ldots, n_T[k]\}$ are the amounts of each resource the tree takes from the cell at each update step (in turn depending on its size $\mathbf{Z}_T$);
- $\mathbf{U}_T^G = \{u_T^G[1], \ldots, u_T^G[k]\}$ is the vector defining the amount of each resource needed at each update step by the tree to *grow*;
- $\mathbf{U}_T^S = \{u_T^S[1], \ldots, u_T^S[k]\}$ 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 $u_T^S[i] < u_T^G[i] < n_T[i]$;
- $\mathbf{R}_T = \{r_T[1], \ldots, r_T[k]\}$ is the amount of each resource stored by the tree at previous update steps;
- $\mathbf{M}_T$ 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;
- $\mathbf{G}_T = \{g_T^r, g_T^t, g_T^l, g_T^f\}$ 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;
- $\mathbf{S} = \{s_1, \ldots, s_l\}$ is a vector defining the number of seeds present in the cell for each of the l species growing in the territory.

3.2 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. Also, in this way it is possible to render more fertile areas located in the proximity of rivers or lakes, since water contained into them spills in neighboring terrain cells. Seeds are also introduced in the model as a resource that moves from cell to cell. Thus, a trees can scatter their seeds in the surrounding area.

Each update step of the automaton covers a given period of time. As shown here, the rules are suitable for one year updates (that is, each cell update reproduces the evolution of the population in one year), but they can nevertheless be further fine–grained in order to model for example single seasons, by changing the amount of resources produced by cells at each step (see the Resource Production rule below).

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 $\mathbf{M}_T(i,j)$. All the resources that are not absorbed by the tree can remain in a cell, or disappear.

Tree reproduction. We have two cases to consider: a tree is present in the cell, or the cell is vacant. 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 seeds contained in a cell if a tree is already present. Instead, a 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:

$$r'_h(i,j) = \frac{r_h(i,j)}{2} + \frac{r_h(i+1,j) + r_h(i-1,j)}{8} + \frac{r_h(i,j+1) + r_h(i,j-1)}{8}$$

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 $\mathbf{S}(i,j)$.

3.3 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).

4 The User Interface

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

- 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), rivers and lakes (containing only water) and so on;

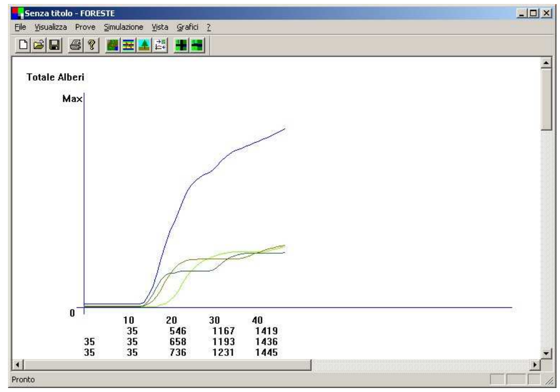

Figure 2: The user interface showing the total number of trees, and the number of trees of each species present in the area for the example shown in Fig. 4.

- 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. There is no upper bound on the number of species that can be defined;
- Additional resources, other than those shown in this article;
- 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. 2,3, and 4. In our experiments we could easily implement on standard desktop PCs CA consisting of thousands of cells, corresponding to several hectares of land, and a single update of the automaton (including the graphic layout) took a few seconds. Thus, the model seems to be suitable for the simulation of case studies of feasible size.

5 Conclusions

In this paper we presented a model based on CA for the simulation of the dynamics of plant populations. Our simulations, reproducing populations of black locusts, oaks and pine trees living on the foothills of the italian alps have shown results qualitatively similar to real case studies.

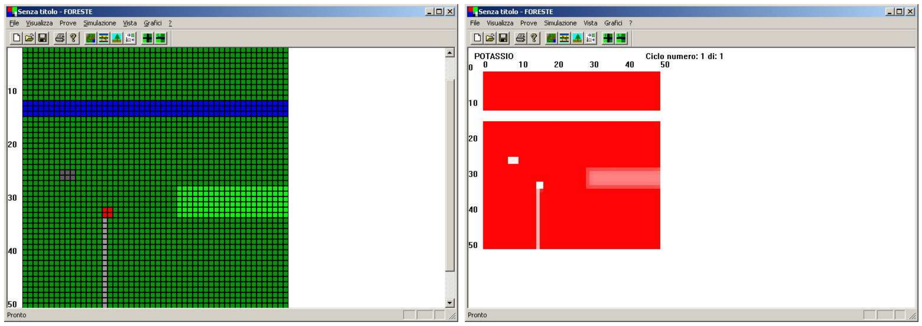

Figure 3: An initial configuration of the automaton (left). The dark strip represents a river. The image to the right shows the initial distribution of potassium in the cells. Darker areas are richer on potassium.

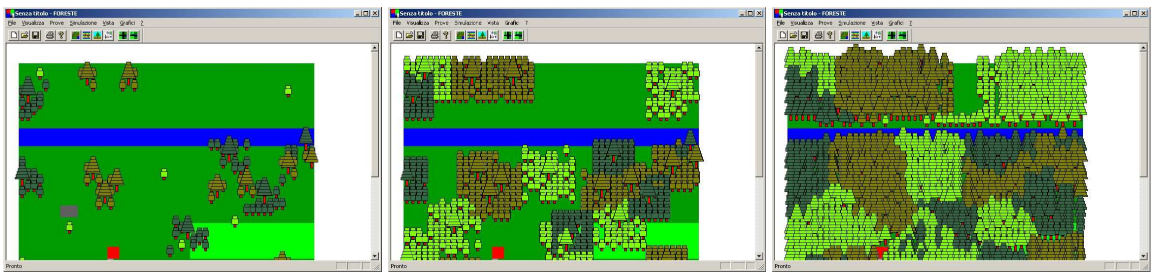

Figure 4: Example of the user interface, showing three different stages of the evolution of a plant population composed by black locusts, oaks, and pine trees, starting from the initial configuration of Fig 3.

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 for the simulation of real case studies and a better understanding of the main factors influencing the dynamics of plant populations.

References

C.Damgaard, J.Weiner, and H.Nagashima. Modelling individual growth and competition in plant populations. *Journal of Ecology*, 90:666–671, 2002.

D.Tilman. *Dynamics and Structures of Plant Communities*. Princeton University Press, 1988.

D.Tilman. Competition and biodiversity in spatially structured habitats. *Ecology*, 75:2–16, 1994.

E.Goles and S.Martinez. *Neural and Automata Networks*. Kluiwer Academic Publishers, 1990.

H.Baltzer, W.P.Braun, and W.Kohler. Cellular automata models for vegetation dynamics. *Ecological Modelling*, 107:113–125, 1998.

J.Ehrlen. The dynamics of plant populations: does the history of individuals matter? *Ecology*, 81: 1675–1684, 2000.

J.L.Uso-Domenech, Y.Villacampa-Esteve, G.Stubing-Martinez, T.Karjalainen, and M.P.Ramo. Mariola: A model for calculating the response of mediterranean bush ecosystem to climatic variations. *Ecological Modelling*, 80: 113–129, 1995.

J.Silverton and D.Charlesworth. *Introduction to Plant Population Biology*. Blackwell, 2001.

M.G.Barbour, J.H.Burk, W.D.Pitts, F.S.Gilliam, and M.W.Schwartz. *Terrestrial Plant Ecology*. Benjamin Cummings, 1998.

Q.Zeng and X.Zeng. An analytical dynamic model of grass field ecosystem with two variables. *Ecological Modelling*, 85:187–196, 1996a.

Q.Zeng and X.Zeng. Two variables dynamic model of grass field ecosystem with seasonal variation. *Ecological Modelling*, 85:197–202, 1996b.

R.L.Colasanti and J.P.Grime. Resource dynamics and vegetation processes: a deterministic model using two–dimensional cellular automata. *Functional Ecology*, 7:169–176, 1993.