

## Transition rule complexity in grid-based automata models

W. Michael Childress\*, Edward J. Ryluel, Jr.\*\* , William Forsythe, Bai-Lian Li and Hsin-i Wu  
*Center for Biosystems Modelling, Department of Industrial Engineering, Texas A&M University, College Station, TX 77843-3131, U.S.A.*

Keywords: grid-based models, cellular automata, spatial automata, succession

### Abstract

Grid-based automata models have been widely applied in simulating ecological process and spatial patterns at all spatial scales. In this paper, we present methods for calculating the effects of number of states, size of the neighborhood, means of tallying neighborhood states, and choice of deterministic or stochastic rules on the complexity and tractability of spatial automata models. We use as examples Conway's Game of Life and models for successional dynamics in a mesquite savanna landscape in south Texas. The number of possible neighborhood state configurations largely determines the complexity of automata models. The number of different configurations in Life, a two-state, deterministic, voting-rule model with an eight-cell Moore neighborhood is 18. A similar model for the seven-state savanna system would have 21,021 different neighborhood configurations. For stochastic models, the number of possible state transitions is the number of neighborhood configurations times the number of possible cell states. A stochastic, unique neighbor model for the savanna system with a Moore neighborhood and seven possible states would have 282,475,249 possible neighborhood-based state transitions. Stochastic models with an eight-cell Moore neighborhood are probably most appropriate for ecological applications. The best options for minimizing the complexity of ecological models are using voting rather than unique neighbor transition rules, reducing the number of possible states, and implementing ecologically-based heuristics to simplify the transition rule table.

### Introduction

Grid-based models such as cellular automata have been widely applied in simulating ecological processes and spatial patterns at all spatial scales (*e.g.*, theoretical populations – Molofsky 1994; wild daffodils – Barkum and Hance 1982; grasses – Silverton *et al.* 1992; shrubs – van Tongeren and Prentice 1986; birch/beechn forest – Wissel 1991; animal populations in a landscape – Zhou and Liebhold 1995; metapopulations – Hanski and Gilpin 1991; landscapes – Turner *et al.* 1993). This simulation approach has great flexibility because each cell in the grid can represent any of a great variety of ecological entities from individual plants in community dynamics models to arbitrarily-large patches of ground such as the data pixels in satel-

lite images. In addition, the rules governing the state change behavior of each cell can be arbitrarily set for neighborhood size and interactions at the discretion of the modeller, or in accord with the nature of the spatial data available.

However, there is still considerable variability in terminology and details for grid-based models in the ecological literature. For example, 'cellular automata' has been widely used to characterize grid-based models with discrete or continuous state values, with stochastic or deterministic state transition rules, or with synchronous or asynchronous (*e.g.*, 'event-driven') state changes. The original formulation of 'cellular automata' was actually very narrowly defined to include a finite number of discrete states with deterministic and synchronous

transition rules (Hogeweg 1988). To avoid confusion between ecological applications and formal mathematical treatments of 'cellular automata' in the complexity and non-linear systems literature, we prefer to call this general class of ecological models 'spatial automata'. In fact, grid-based automata models are themselves a subset of a larger class of models called 'polyautomata', which includes such diverse formalisms as compartment models, network models, and neural networks (Smith 1976).

As for any modelling paradigm, it is important to understand as much as possible the general behavior of spatial automata as a mathematical construct. This is to make sure that ecologically-interesting behavior of applied models can be distinguished from artifact behavior arising from the mathematics of the model. One factor crucial to spatial automata models is the number of rules required for the state transition matrix, given the number of possible states, the size of the neighborhood, the means of tallying neighborhood states, and the choice of stochastic or deterministic rules. As number of states and the size of the neighborhood increase, the number of rules required to completely specify the system also increase, but how just how quickly?

In this paper, we present means for calculating the effects of these factors on the state transition rule matrix for different formulations of spatial automata, including, but not limited to, cellular automata. This presentation is semi-formal in that we introduce terminology and notation that we have found useful in understanding different automata models. However, our main objective is to demonstrate that while complete specification of a spatial automata model can become very complex very fast, there are a number of simplifications and tradeoffs to be made in selecting a particular formalism for application. This is to make sure that the model remains computationally feasible, and, for empirical applications, that the available data are adequate for parameterizing the model.

## Cellular automata

We begin with a definition of cellular automata which, because it is a simple, narrowly-defined for-

malism, is useful as a starting point for considering other spatial automata formalisms. Cellular automata is a grid-based model in which each cell in the grid is an independently-behaving entity (an 'automaton') which at any moment in time assumes one of a set of possible discrete states. The grid is a regular tessellation of cells, most commonly an orthogonal two-dimensional grid with square cells. A common ecological application is to specify each cell as the possible location of a single plant, and the possible states for each cell as the presence or absence of a plant in that cell (*e.g.*, Molofsky 1994). The model would thereby simulate the population dynamics of the plant species within the spatial extent of the model grid. Cell behavior involves each cell deciding its state at the next time step in the model simulation. All cells make this decision and change states accordingly in synchrony. The decision is based on the current cell state and those of cells in a neighborhood of uniform number and spatial configuration for each cell in the grid. The particular configuration of states for the cell and its neighborhood of cells is called a *neighborhood Configuration*. For all cellular automata models, there is a set of rules which unequivocally specifies the next state of the cell given its neighborhood configuration, that is, cellular automata in the strict sense is based on *deterministic* rules for state changes. For any possible neighborhood configuration, there is a rule specifying the next state for the cell. In practice, there are a limited number of possible cell states, but a much larger possible number of neighborhood configurations, so many different configurations will indicate the same outcome state. Like the neighborhood size and shape, the state transition rules are identical for all cells in the grid.

The *neighborhood* of each cell can be defined in a number of ways, but two of the most common patterns in orthogonal grids with square cells are the von Neumann and Moore neighborhoods (Hogeweg 1988) (Fig. 1). The von Neumann neighborhood includes the four adjacent cells sharing a common edge with the central cell. The Moore neighborhood includes all eight adjacent cells touching the central cell along the edges or at the corners. Actually, because each cell considers its own current state as well as those of its neighbors in determining its next state, the *neighborhood*

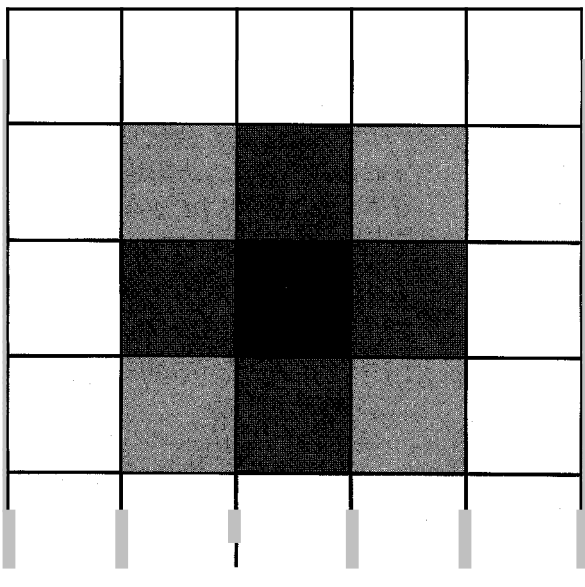


Fig. 1. A cell (black) and its neighborhood of cells (gray tones) in a grid-based model. The four dark gray neighbors adjacent along the cell edges comprise a von Neumann neighborhood. These and the four light gray neighbors at the cell corners comprise an eight-cell Moore neighborhood.

*configuration* for the von Neumann *neighborhood* considers the states for five cells rather than four, and that of the Moore *neighborhood* nine cells instead of eight. It is convenient to use the term *neighborhood* for the four or eight nearby cells, and *neighborhood configuration* for the neighbors and the central cell itself. Larger neighborhoods can also be specified by adding more cells in a roughly radially symmetric pattern, *i.e.*, in concentric rings of more and more cells around the central cell. However, as more cells are added to the neighborhood, the number of possible neighborhood configurations of state values also increases, so few models consider more than an eight-cell neighborhood.

The strict formulation of cellular automata has two attributes that distinguish it from other spatial automata. First, there is a finite number of discrete states for each cell. Second, the rules for state change are deterministic. The formal rule set for a cellular automata model would therefore consist of a two-column table in which each possible neighborhood configuration is listed in the left column

and the designated next cell state is listed in the right column. Because there is only one possible outcome for each neighborhood configuration, this formulation is considered a one-dimensional rule table. While most ecological automata models use discrete states, many use stochastic rules which specify probabilities for state change rather than deterministic rules. In a formal specification of the rules for a stochastic automata model, there is a column for each possible outcome state in the model, with a probability listed for each of these states for each neighborhood configuration. This is then a two-dimensional rule table.

A well-known cellular automata model is John Conway's Game of Life (Gardner 1970). In this simple simulation, each cell represents the potential location of a sessile microbe-like organism. Each cell can be either 'alive' or 'dead', that is, this is a two-state system. The neighborhood of each cell is an eight-cell Moore-type neighborhood. The simple deterministic rules for this system dictate under what conditions a dead cell is 'born' into a live cell, a living cell dies, and a dead or living cell remains a dead or living cell (Table 1). Note that there are only **six** rows in this table because neighborhood configurations with the same outcome have been pooled into single rows. In a more formal specification for the model there would be one row for each possible neighborhood configuration. These rules can also be stated in an equivalent stochastic rule table in which the probabilities for the outcome states are 1 and 0 (Table 2).

A model for successional dynamics in a mesquite savanna landscape at the La Copita Research Area in south Texas (Scalan and Archer 1991) will be used as a more complex example for the sections below. This model simulates changes in vegetation cover from 1960 to 1983 within 20 x 20 m cells in a grid superimposed on aerial photographs of the study area. This model is actually a Markovian matrix model in that the size of the neighborhood configuration is one, that is, the cell considers only its own state in determining its next state. In addition, the state transition rules are stochastic rather than deterministic (Table 3), so this model is not a strict cellular automata model. This model is of interest because there are seven possible states, and we would like to consider how complex the

**Table 1.** Deterministic state transition rule table for Conway's Game of Life, a strict cellular automata model. There are two possible state values for each cell, 'alive' or 'dead'.

Neighborhood configuration		
Current cell state	Number of 'alive' neighbors	Next cell state
Dead	0-2	Dead
Dead	3	Alive
Dead	4-8	Dead
Alive	0-1	Dead
Alive	2-3	Alive
Alive	4-8	Dead

model would become if the neighborhood configuration size was expanded from one (Markov) to five (von Neumann) or nine (Moore) cells.

### Basic notation and terminology

There are three basic numbers involved in calculating number of state transition rules for any spatial automata formulation:

- $s$  the number of possible discrete states of each cell,
- $n$  the number of cells in the specified neighborhood configuration, *i.e.*, the cell itself and all other cells in the neighborhood, and
- $c$  the number of possible neighborhood configurations of cell states.

The number of states  $s$  and the neighborhood configuration size  $n$  are specified in the model formulation, but  $c$  is calculated in different ways depending on how the neighborhood is polled for the transition rules. If each neighboring cell is uniquely identified by its spatial orientation with respect to the center cell (*e.g.*, north, south, east, or west), we term this model a 'unique neighbor' system. On the other hand, if the number of cells in each state are simply tallied without consideration for which state is in which cell, then we term it a 'voting' system. The roll of two dice is a voting system because we do not distinguish each die in tallying the result; all permutations of each outcome are considered identical. In the unique neighbor system, each permutation of states among cells in the neighborhood is considered a unique neighborhood configuration.

**Table 2.** Stochastic state transition rule table for Conway's Game of Life, a strict cellular automata model. There are two possible state values for each cell, 'alive' or 'dead'. Because all probabilities are either 1 or 0, this is still a deterministic model.

Neighborhood configuration		Next state probabilities	
Current cell state	Number of 'alive' neighbors	Alive	Dead
Dead	0-2	0.00	1.00
Dead	3	1.00	0.00
Dead	4-8	0.00	1.00
Alive	0-1	0.00	1.00
Alive	2-3	1.00	0.00
Alive	4-8	0.00	1.00

**Table 3.** State transition matrix for vegetation dynamics in a mesquite savanna at the La Copita Research Area in south Texas (Scanlon and Archer 1991). There are seven possible states for each 20 x 20 m cell. State changes are based on stochastic rules for a one-cell neighborhood. The vegetation notation for cell states are: W - woodland; Wm - woodland margin; C - coalesced cluster; Cm - coalesced cluster margin; M - mature cluster; P - pioneer cluster; and H - herbaceous.

Current cell state	Final cell state						
	W	Wm	C	Cm	M	P	H
W	0.970	0.027	0.003	0.000	0.000	0.000	0.000
Wm	0.497	0.429	0.000	0.042	0.000	0.011	0.027
C	0.373	0.090	0.328	0.119	0.030	0.060	0.000
Cm	0.132	0.126	0.137	0.423	0.058	0.095	0.021
M	0.044	0.065	0.196	0.522	0.109	0.065	0.000
P	0.088	0.121	0.036	0.170	0.104	0.447	0.036
H	0.019	0.086	0.019	0.216	0.111	0.475	0.074

The distinction between these two systems is important in calculating the number of possible neighborhood configurations. For the unique neighbor system:

$c = s^n$  the number of possible neighborhood configurations for a spatial automata based on 'unique neighbor' rules, where

$c = s^1$  for a Markov neighborhood configuration ( $n=1$ ),

$c = s^5$  for a von Neumann neighborhood configuration ( $n=5$ ), and

$c = s^9$  for a Moore neighborhood configuration ( $n=9$ ).

For voting rules, different permutations of state values are considered identical, so many neighborhood configurations which are considered unique

in the ‘unique neighbor’ system are considered identical in the ‘voting’ system. However, the state of the central cell is still considered a unique cell, while all the other cells in the neighborhood are indistinguishable. Therefore there are fewer possible neighborhood configurations for a ‘voting’ system, and  $c$  is calculated as:

$$c = s \times \frac{(n+s-2)!}{(n-1)!(s-1)!}, \text{ where}$$

$$c = s \times \frac{(1+s-2)!}{(1-1)!(s-1)!} = s \text{ for a Markov neighborhood configuration,}$$

$$c = s \times \frac{(s+3)!}{4!(s-1)!} \text{ for a von Neumann neighborhood configuration, and}$$

$$c = s \times \frac{(s+7)!}{8!(s-1)!} \text{ for a Moore neighborhood configuration.}$$

The Life model has  $s = 2$  and  $n = 9$ , and is based on voting rules to determine whether the cells are dead or alive at the next time step. The number of possible neighborhood configurations is therefore:

$$c = 2 \times \frac{(2+7)!}{8!(2-1)!} = 18.$$

This is reasonable because there are nine different combinations of On-Off counts for the eight neighbors (*i.e.*, 0-On and 9-Off through 9-On and 0-Off), and two possible initial states for the cell itself, for a total of  $2 \times 9 = 18$ . If this model instead used a unique neighbor system, then  $c = s^n = 2^9 = 512$ .

The original La Copita model has a neighborhood of only one cell, so for either the unique neighbor or voting rule formulations,  $c = s = 7$ . If this were a voting rule model with a Moore neighborhood as for the Life model, then:

$$c = 7 \times \frac{(7+7)!}{8!(7-1)!} = 21,021.$$

The number of neighborhood configurations obviously increases dramatically with an increase in the number of possible states from 2 (Life) to 7 (La Copita).

Before considering the number of rules that are required for these different spatial automata formulations, it is necessary to carefully define terms relative to state changes and neighborhoods:

A *transition* is the change in state ( $x_1$  to  $x_2$ ) of a cell from time  $t_1$  to time  $t$ . This can also be called a *state transition*.

$s$  is the number of possible *transitions* for a cell in any particular initial state  $x_1$ .  $s$  is also the number of states and therefore the number of possible final states.

$s^2$  is the number of possible *transitions* for all possible combinations of initial and final states  $x_1$  and  $x_2$ , for both *deterministic* and *stochastic* systems. This is simply the product of the number of possible initial and final states.

A *neighborhood-based transition* is the cell state change to a particular final state based on an initial *neighborhood configuration*, part of which is the initial cell state. There are a number of possible *neighborhood configurations* for each possible initial cell state, so there are more possible *neighborhood-based transitions* than there are possible *state transitions*.

A *transition rule* is a rule which dictates *state transitions* for a cell given a particular initial *neighborhood configuration*.

- There is a one-to-one correspondence between *neighborhood configurations* and *transition rules* in a complete, formal transition table.
- In *stochastic* systems, there may be multiple possible final states  $x_2$  from the same initial *neighborhood configuration*, and therefore multiple possible *transitions*. However, because these all have the same initial *neighborhood configuration*, they are all part of a single *transition rule*.
- A *transition rule* corresponds to a single row in a complete transition table/matrix for a spatial automata model. In cellular automata and other *deterministic* models, there is only a single column of final states  $x_2$ , but in *stochastic* models,

there are columns for each of the  $s$  possible states.

$s^c$  is the number of all possible *transition rules* for all  $s$  possible states and  $c$  *neighborhood configurations* for all conceivable *deterministic* systems. For *stochastic* systems, there are an infinite number of possible probabilities, and so an infinite number of possible *transition rules*.

A *transition rule set* is the specified set of *transition rules* for all possible *neighborhood configurations* in a single model formulation. The *transition rule set* corresponds exactly to the transition table/matrix.

$$s^n = c$$

is the number of possible *neighborhood-based transitions* for a given *deterministic* ‘unique neighbor’ *transition rule set*. This is the number of possible states raised to the number of cells in the neighborhood configuration, that is, the number of possible *neighborhood configurations*.

$$s^{n+1} = c \times s$$

is the number of possible *neighborhood-based transitions* for a given *stochastic* ‘unique neighbor’ *transition rule set*. This is the number of possible final states times the number of possible states raised to the number of cells in the neighborhood configuration, that is, the number of possible final states times the number of possible *neighborhood configurations*.

$$s \times \frac{(n+s-2)!}{(n-1)!(s-1)!}$$

is the number of possible *neighborhood-based transitions* for a given *deterministic* ‘voting’ *transition rule set*. This is the number of possible *neighborhood configurations*.

$$s^2 \times \frac{(n+s-2)!}{(n-1)!(s-1)!}$$

is the number of possible *neighborhood-based transitions* for a given *stochastic* ‘voting’ *transition rule set*. This is the number of possible final

states times the number of possible *neighborhood configurations*.

## Changes in complexity for different models

The overall complexity of a spatial automata model relates directly to the size of the state transition matrix for the particular model formulation. This is a critical concern in model development for two reasons. First, a more complex model can be computationally intractable. If the number of neighborhood configurations and the transition matrix grow too large, computation of next states for all cells in a large grid may be drastically slowed, even for fast computer systems. Second, development of empirical models requires adequate data to identify all possible state transitions and neighborhood configurations in the system. The more complex the model, the more data required to parameterize it.

The size of the transition matrix for any model formulation is dependent on 1) the number of possible states, 2) the size of the neighborhood, 3) the system for tallying neighbor states (*i.e.*, unique neighbor or voting), and 4) deterministic or stochastic rules. The choice of deterministic or stochastic rules determines whether the matrix will be one or two-dimensional, and the difference in number of possible state transitions is a factor of  $s$ . The choice between unique neighbor and voting rules also leads to a significant difference in the number of neighborhood configurations, with the voting system resulting in a smaller number. However, it is not quite so clear how important the number of states and the size of the neighborhood are to complexity of the models relative to the other factors.

The key determining factor in model complexity is the number of possible neighborhood configurations. Consider simple variations in the Life model formulation with respect to number of cells in the neighborhood and a choice between unique neighbor and voting rules (Table 4). There is a linear increase in number of configurations with increase in neighborhood size for voting rules, but a geometric increase for the unique neighbor rules. However, the linear increase for the voting rules is strictly an artifact of having  $s = 2$ , because the increase is much greater than linear for the La Copita system where  $s = 7$  (Table 5). The unique

Table 4. Number of possible neighborhood configurations in the Life cellular automata model for different neighborhood sizes and for 'unique neighbor' and 'voting' neighborhood tally systems. The number of possible states  $s$  in this model is two.

Number of cells in the neighborhood	Neighborhood tally system	
	Unique neighbor	Voting
1 (Markov)	2	2
5 (von Neumann)	32	10
9 (Moore)	512	18

Table 5. Number of possible neighborhood configurations in a deterministic spatial automata model for the mesquite savanna system under different neighborhood sizes and for 'unique neighbor' and 'voting' neighborhood tally systems. The number of possible states  $s$  in this model is seven.

Number of cells in the neighborhood	Neighborhood tally system	
	Unique neighbor	Voting
1 (Markov)	7	7
5 (von Neumann)	16,807	1,470
9 (Moore)	40,353,607	21,021

Table 6. Number of neighborhood configurations for different number of states, unique neighbor and voting neighborhood tally systems, and different neighborhood sizes. These data are also the number of possible neighborhood-based transitions for a deterministic model.

Number of states	Neighborhood tally system					
	Unique neighbor			Voting		
	Neighborhood size			Neighborhood size		
	1	5	9	1	5	9
2	2	32	512	2	10	18
3	3	243	19,683	3	45	135
4	4	1,024	262,144	4	140	660
5	5	3,125	1,953,125	5	350	2,475
6	6	1,776	10,077,696	6	756	7,722
7	7	16,807	40,353,607	7	1,470	21,021

neighbor system clearly leads to extremely complex models as the neighborhood size or number of states increase. To consider the effect of using stochastic instead of deterministic rules, the number of neighborhood configurations for the Life and La Copita models should all be multiplied by 2 and 7, the  $s$  value for each model, respectively, to determine the number of neighborhood-based transitions or the number of entries in the state transition

matrix. In this case, the number of possible neighborhood-based transitions of the La Copita model with a Moore neighborhood and stochastic unique neighbor rules would be 282,475,249.

The alternatives for neighborhood tally system (*i.e.*, unique neighbor or voting) and deterministic versus stochastic systems are only two for either factor, so options are limited for reducing complexity. The size of the neighborhood is also rather limited for two-dimensional models. Ordinarily there should be a symmetric number of neighboring cells on each of the four sides of the center cell; this means that increases in number of neighbors would ordinarily be in increments of four (*e.g.*, Markov to von Neumann to Moore neighborhoods). Further, the two examples in Tables 4 and 5 show that neighborhood configurations with more than 9 cells are probably intractable except in very special cases. Number of states would seem to provide the greatest flexibility in controlling complexity. Table 6 indicates how rapidly the number of neighborhood configurations increases with number of states for the various other factors.

Data requirements for parameterizing these models is even more problematic, and for the seven-state La Copita system with a unique-neighbor Moore neighborhood essentially impossible. To exhaustively parameterize empirical spatial automata models, at least one instance of each neighborhood configuration must be present in the data, assuming a perfectly deterministic model. A number of instances of each configuration must be present to simply identify possible final states for a stochastic model, much less to calculate probabilities for each transition. Further, if some of the states are much less frequent than others in the data, then for a perfectly random distribution of states across the grid, neighborhood configurations including these states will be much less frequent than others, requiring an even larger data set to find examples for these rare configurations. And still further, if the states are not randomly distributed but tend to be either aggregated or dispersed, some configurations will be even rarer in the data.

The Markovian La Copita model was originally derived from 1,471 cell observations for the period 1960–1983, which coincidentally corresponds almost exactly with the number of possible neighborhood configurations (1,470) for a deterministic vot-

ing system with a five-cell von Neumann neighborhood configuration. However, the similarity of these numbers does not imply that this model could actually be easily calibrated from the observations. It is very unlikely that all possible configurations are represented in the observations; instead those configurations comprised of the more common cell states are frequently represented, and those including rarer states are not present at all. Theoretical models such as the Life model are not constrained by parameterization concerns because all rules are theoretically specified, but these models are still hindered by computational tractability as the transition table increases in size.

### Recommendations for reducing complexity

The objective of most spatial automata models is to address effects of interactions among the cells, whether they represent individual organisms, patches of different cover or community types, or light reflectance in satellite images. For this reason, constraining the nature of the interaction rules to meet computational or data limits defeats the purpose of the model. We should be free to implement exactly those interaction rules we consider most relevant to our ecological system, whether theoretical or real. Somehow, the model should be compromised to suit the system, rather than the reverse.

In an orthogonal grid, the four edge cells in a von Neumann neighborhood are only marginally closer to the central cell than the four corner cells added in a Moore neighborhood. Although increasing the neighborhood configuration size from five to nine cells dramatically increases the complexity of any spatial automata model, for purposes of ecological interactions, it seems unreasonable to preclude interactions between the central cell and the four corner cells. The next larger symmetrical neighborhood configuration would include **13** cells, with one cell added in line with each of the four cells along the center cell edges. However, these additional cells would be twice the distance away from the center cell as the edge cells, so the intensity of the interactions would probably be qualitatively different from the adjacent cells in a real ecological system. The primary choice with respect to neighborhood configuration size in eco-

logical systems may be between the nine-cell Moore neighborhood configuration and a one-cell, no-interaction Markovian model. However, Zhou and Liebhold (1995) found that neighborhoods larger than the Moore in their two-state model of gypsy moth outbreaks performed better than Markovian, von Neumann, or Moore neighborhood versions.

Choices for the number of states implemented may also be limited if the objective is to build realistic ecological models. For example, the La Copita system has seven clearly-defined cover types. Reducing the number of states to meet parameterization or computation needs would defeat the objective of investigating observed cover-change dynamics. Models in which each cell represents an individual organism such as a tree often implement qualitatively different interaction rules for different ages of the organism. For example, a tree-cell may need to reach a certain age or size before it can reproduce and colonize an adjacent empty cell. In this case, restricting the states to 'alive' or 'dead' as in the Life model would defeat investigating effects of age structure and time delays on successional dynamics in a forest.

Nonetheless, some reductions in the number of states might be achieved by deriving alternative models which aggregate different subsets of the total list of states. For example, a landscape model might include, say, 20 different community types, a number of states clearly very intractable for most spatial automata models. Alternative models could focus on particular classes of states, while aggregating others. If all grassland and shrubland communities could be aggregated into single states for each class, then each forest type could be retained as a separate state to investigate forest dynamics across the landscape. An alternative model could then aggregate forest and grassland communities while retaining separate shrub communities, and so on. This divide-and-conquer approach could also identify which communities have the most ecologically interesting (*e.g.*, rapidly expanding or declining) behaviors, so that less-interesting communities could be aggregated in a final model.

The choice between deterministic and stochastic models can also strongly influence the complexity of spatial automata models. Theoretical ecological models can usually make good use of deterministic

rules to simplify the transition matrix by judicious selection of states and unique neighbor versus voting rules. However, most empirical ecological data indicate that different final states can result from the same initial neighborhood configuration. This means that empirical models should usually make use of a stochastic transition matrix. A large matrix does not necessarily mean a proportional increase in model complexity, however, because most of the probabilities in the table may be 0. The problem is that the data required to calculate the probabilities and confirm that most are 0 can be prohibitively large.

One approach to simplifying calibration of a large and potentially complex stochastic transition table is to set certain limits on the precision needed for calculation of probabilities. For example, all probabilities less than 0.10 could be converted to 0, and those greater than 0.90 could be set to 1.00. Alternatively, all probabilities could be rounded to the nearest 0.25. A series of models could be constructed with increasing precision in the probability estimates, and each model compared with the observed grid data to estimate how much increased precision actually improves model performance. It may well be that data needs for transition table calibration can be greatly curtailed if more data would not significantly improve replication of observed data.

The choice between unique neighbor and voting rules for tallying neighborhood states is fortunately straightforward. In most ecological applications, there is little need to uniquely identify each neighbor. Because all neighbors in the von Neumann and Moore neighborhoods are essentially the same distance from the central cell, the influence of two different neighbors with the same state should be identical. This means that in almost all cases, voting rules will be the preferred and much simpler formulation (Table 6). An exception would be models in which an environmental gradient of some sort is implemented in the model grid. For example, there may be prevailing winds or a slope across the grid which influences direction of dispersal. In this case, there may be a preferential influence of upwind or upslope cells, which could be implemented in the model by incorporating unique neighbor rules. Wissel (1991) implemented directionally in his model of forest dynamics be-

cause of potential solar radiation damage to beech trees. This was implemented by altering the transition probabilities for cells with empty neighbors to the south, east, and west, thereby considerably increasing the complexity of the transition rules. Analytical methods from digital image processing can be adapted specifically to identify significant gradients and asymmetrical cell interactions in dynamic systems (*e.g.*, Pratt 1991). Implementation of gradients across the model grid is itself a considerable complication in model design because the state transition rules might no longer be uniform for all cells in the grid.

The most important means of controlling the complexity of spatial automata models is certainly implementation of heuristic information into the transition rules. In most ecological systems, certain state transitions would never or hardly ever be expected to occur. In successional systems, for example, a change from a severely disturbed site to mature forest in one time step would never be expected. In many dynamic ecological systems, there is a sequence of states that are usually followed, such as the seres in a succession sequence. For example, Turner *et al.* (1993) use a deterministic sequence of successional states in their model of forest dynamics in a landscape mosaic. In other cases, neighborhood configurations which are functionally identical can be aggregated into single rows in the transition table. In the Life model, for example, configurations with 4, 5, 6, 7, or 8 alive neighbors are all functionally identical, and these are aggregated in the transition table (Table 1). Wherever the pattern of final states are identical for different neighborhood configurations, these configurations can theoretically be aggregated. The key is to look for similarities among these configurations so heuristic rules can be derived for these aggregations.

One particularly useful heuristic approach is to construct a simple state transition table such as that for the La Copita system in Table 3. This matrix looks only at initial and final states, and does not consider neighborhoods (except for the one-cell Markovian neighborhood). Wherever the probability for a particular state transition is 0, this indicates that regardless of the neighborhood configuration, this transition will never occur. Therefore all possible neighborhood configurations which include this initial state will have a 0 probability for this

particular transition. This technique can considerably simplify calibration of a stochastic transition matrix.

The knowledge of the system on the part of the modeller is certainly the most important factor in controlling model complexity. Compromising among modelling objectives, ecological realism, and computational tractability is ultimately the responsibility of the ecologist. As for any other modelling paradigm, this is the main part of the art of ecological modelling.

### Acknowledgments

The authors are pleased to acknowledge support from NSF Grants BSR-91-09240 and DEB-93-06679 during the preparation of this paper.

### References

- Barkum, J.P. and Hance, C.E. 1982. Population dynamics of the wild daffodil (*Narcissus pseudonarcissus*). III. Implications of a computer model of 1000 years of population change. *Journal of Ecology* 70: 323–344.
- Gardner, M. 1970. Mathematical games. The fantastic combinations of John Conway's new solitary game 'life'. *Scientific American* 223(4): 120–133.
- Hanski, I. and Gilpin, M. 1991. Metapopulation dynamics: brief history and conceptual domain. *Biological Journal of the Linnean Society* 42: 3–16.
- Hogeweg, P. 1988. Cellular automata as a paradigm for ecological modelling. *Applications of Mathematics and Computing* 27: 81–100.
- Molofsky, J. 1994. Population dynamics and pattern formation in theoretical populations. *Ecology* 75: 30–39.
- Pratt, W.K. 1991. *Digital image processing*. 2nd Ed. John Wiley & Sons, New York.
- Scanlon, J.C. and Archer, S. 1991. Simulated dynamics of succession in a North American subtropical *Prosopis* savanna. *J. Veg. Sci.* 2: 625–643.
- Silverton, J., Holtier, S., Johnson, J. and Dale, P. 1992. Cellular automaton models of interspecific competition for space – the effect of pattern on process. *Journal of Ecology* 80: 527–534.
- Smith, A.R., III. 1976. Introduction to and survey of polyautomata theory. *In Automata, Languages, Development*. pp. 405–422. Edited by A. Lindemayer and G. Rozenberg. North-Holland, Amsterdam.
- Turner, M.G., Romme, W.H., Gardner, R.H., O'Neill, R.V. and Kratz, T.K. 1993. A revised concept of landscape equilibrium: Disturbance and stability on scaled landscapes. *Landscape Ecology* 8: 213–227.
- van Tongeren, O. and Prentice, I.C. 1986. A spatial simulation model of vegetation dynamics. *Vegetatio* 65: 163–173.
- Wissel, C. 1991. A model for the mosaic-cycle concept. *In The Mosaic-Cycle Concept of Ecosystems*. pp. 22–45. Edited by H. Remmert. Springer-Verlag, Berlin.
- Zhou, G. and Liebhold, A.M. 1995. Forecasting the spatial dynamics of gypsy moth outbreaks using cellular transition models. *Landscape Ecology* 10: 177–189.