Simulation of Development Alternatives Using Neural Networks, Cellular Automata, and GIS for Urban Planning

Anthony Gar-On Yeh and Xia Li

Abstract

This study integrates neural networks and cellular automata (CA) to simulate development alternatives for planning purposes. Most of the existing CA just focus on simulating realistic urban dynamics. This paper demonstrates that development alternatives can be simulated by incorporating planning objectives in CA. It is important to define appropriate parameter values for simulating development alternatives according to the planning objectives of planners and decision makers. Training neural networks can automatically yield the parameter values for urban simulation. GIS and remote sensing provide the training data for calibrating the model. However, the simulation can inherit past land-use problems if the original training data are used to calibrate the model. The original data should be assessed and modified so that the model can remember the past "failure" in land development. Planning objectives can thus be embedded in the model by properly modifying the training data sets. The training is robust because it is based on the well-defined back-propagation algorithm. Experiments were carried out by using the city of Dongguan, China as an example to test the model.

Introduction

The advantages of cellular automata (CA) have been well recognized because they have been used by researchers from a variety of disciplines (Goles, 1989; Batty and Xie, 1994; Batty and Xie, 1997; Clarke *et al.*, 1995). Cellular automata are spatial dynamic modeling techniques that have been widely applied to the simulation of complex dynamic systems, such as biological reproduction, chemically self-organizing systems, propagation phenomenon, and human settlements. Cellular automata can be developed to investigate the principals of dynamic systems and test ideas and assumptions for hypothetical applications (Couclelis, 1997; Batty *et al.*, 1999). The merits of CA are that very complex behaviors and global structures can emerge from some simple local actions or rules. A very successful example of CA applications in physics is to model complex lattice gases (Binder, 1989). Biological reproduction and evolution can also be modeled through the principles of "game of life" (Batty and Xie, 1994).

Urban CA have been developing rapidly for the simulation of complex urban systems since the late 1980s. Much interesting research has been documented in urban simulation (White and Engelen, 1993; Batty and Xie, 1994; Clarke and Gaydos, 1998). Urban systems involving spatial and sectoral interactions cannot be easily adapted to the functionality of current GIS software (Batty *et al.*, 1999). This can be overcome by the integration of GIS with CA-based approaches. Modeling cities with CA is a relatively new approach although it has distant roots in geography in the work of Hägerstrand (1965) and Tobler (1979) (Clarke and Gaydos, 1998).

There are at least three main types of urban CA. The first type of urban CA is to simulate urban dynamics that can be explained by urban theories. Cellular automata are used to test ideas and assumptions for hypothetical cities without using real data. For example, Webster and Wu (1999) present an interesting CA model to implement urban theories concerning developers' profit-seeking and communities' welfare-seeking behaviors, and the mediating effects of alternative systems of land-use rights. White and Engelen (1993) develop a CA model to generate fractal or bifractal land-use structures for urbanized areas. Another example is to simulate the formation of urban sub-centers by incorporating the combined forces of a series of local actions (Wu, 1998).

The second type of urban CA provides realistic urban simulation by using real data sets. Clarke and Gaydos (1998) applied CA to simulate and predict urban development in the San Francisco Bay region in California and the Washington/Baltimore corridor in the eastern United States. White *et al.* (1997) developed a CA model to simulate the land-use pattern of Cincinnati, Ohio. Li and Yeh (2000) simulated rapid urban expansion in the Pearl River Delta, China by the integration of CA and GIS. This type of urban CA can be used to predict the direction and pattern of future urban development if the current urban processes continue into the future. This is useful for planners to locate future urban growth areas, evaluate the impacts of urban expansion, and plan and provide infrastructure to support the development.

The third type of urban CA develops normative planning models to simulate different urban forms based on planning objectives. Yeh and Li (2001) used CA to simulate alternative urban forms, ranging from monocentric to polycentric urban development, by incorporating different planning objectives

Anthony Gar-On Yeh is with the Centre of Urban Planning and Environmental Management, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, P.R. China (hdxugoy@hkucc.hku.hk).

Xia Li is with the School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, P.R. China (xlib@gis.sti.gd.cn); also with the Guangzhou Institute of Geography, Guangzhou 510070, P.R. China and Centre of Urban Planning and Environmental Management, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, P.R. China (lixia@graduate.hku.hk).

Photogrammetric Engineering & Remote Sensing
Vol. 69, No. 9, September 2003, pp. 1043–1052.

0099-1112/03/6909–1043$3.00/0

that are specified by planners and decision makers. The model can explore development alternatives that can minimize development and environmental costs for sustainable development. Ward *et al.* (2000) also developed a constrained CA model which has been applied to an area in Gold Coast, a rapidly urbanizing region of coastal eastern Australia. They demonstrated that CA can simulate planned development as well as realistic development by incorporating sustainability criteria in the simulation. Their study shows that economic, physical, and institutional control factors can be incorporated to modify, constrain, and prohibit urban growth.

A problem for urban CA is how to define parameter values for producing realistic simulation. The simulation of complex urban systems needs to use numerous spatial variables. The contribution of each spatial variable to the simulation is quantified by some kind of functions. In these functions, a variable is usually associated with one parameter or weight. The parameter value is to address the importance of the variable. A variable associated with a larger parameter value usually means that it is more important than other variables. A CA may have many parameter values to be determined. These parameter values have crucial effects on the results of CA simulation. Studies indicate that urban simulation is very sensitive to parameter values (Wu, 2000).

In the simulation of realistic cities, calibration procedures should be employed to find suitable parameter values to produce the best fit of actual development. Unfortunately, there is no a universally applicable method of calibration due to the complexity of real urban development. So far there are only very limited studies in addressing the calibration issues in CA simulation. For example, Wu and Webster (1998) used multicriteria evaluation (MCE) to define parameter values heuristically. Clarke *et al.* (1997) consider that visual tests are useful to establish parameter ranges and to make rough estimates of parameter settings. The impact of each parameter is assessed by changing its value and holding other parameters constant. Clarke and Gaydos (1998) suggest that calibration can be done by statistically testing the observed against the expected. The method is to find which set of parameters can lead the model to produce the best fits. These parameters are then used for prediction. There are numerous possible combinations of different values of parameters. Their experiments have used more than 3,000 combinations, which need a high-end workstation to run several hundreds of hours for the calibration. A final set of parameters was chosen based on the best fit. The parameters were then used to run the model for the prediction of future development. The method is very time-consuming because it needs to compare all possible combinations of parameters. Another problem is that the combinations are infinite and a sound search procedure is hard to design.

Another method to calibrate CA is to use neural networks which can be trained with observation data (Li and Yeh, 2001). Training neural networks is very convenient by using the back-propagation algorithm. Li and Yeh (2001) have shown that this is quite effective in the simulation of real cities. This paper further extends the methodology and demonstrates that, with proper training, a neural-network-based CA can be developed to simulate development alternatives for urban planning. Generation and formulation of planning scenarios is a very common and important task in urban planning. It is possible to train the neural network so that development plans can be simulated and generated. Remote sensing and GIS data are used for training the neural network. If original training data are directly used, the model can simulate the development based on the past trends, i.e., the development pattern in the future when the present development trend continues without planning intervention. There is a need to assess the training data and modify them so that the neural network can be trained to remember the past "undesirable" development. Planning objectives, such as "smart" urban growth and sustainable development can be incorporated in the simulation. The training procedure is sound because of the use of the back-propagation algorithm. This proposed model has two main advantages over traditional CA methods. It can minimize the uncertainties in defining parameter values and solve the problem on how to define parameter values in generating development alternatives. The following sections will demonstrate that neural networks can be easily devised to simulate alternative urban growth according to different planning objectives that can be used for urban planning.

Neural Networks

Cities are open and non-linear complex systems. Mathematical equations have limitations in simulating such systems. Artificial neural networks (ANN) have the capability of mapping nonlinear features. Artificial neural networks can be used to recognize and classify patterns through training or learning process. Studies indicate that neural networks provide levels of performance superior to those of conventional statistical models because neural networks can handle well the uncertainties of spatial data (Openshaw, 1993; Fischer and Gopal, 1994). Geographical analysis is usually based on incomplete and inconsistent data due to the complexity of nature. Classical quantitative methods cannot be used to solve complex spatial decision problems which are highly assumption-dependent and application-specific. Neural networks have been widely and seemingly extremely successfully applied in many disciplines that have a high degree of hardness (Openshaw and Openshaw, 1997; Openshaw, 1998).

Neural networks have a very simple structure for processing data. The basic processing units in a neural network are so called neurons or nodes, which are organized in a couple of layers. A neural network usually has one input layer, one output layer, and no or some hidden layers between. A neuron in the input layer or a hidden layer is connected to all the neurons in the next layer for passing information. All the neurons, except those in the input layer, perform the two simple processing functions—collecting the activation of the neurons in the previous layer and generating activation as the input to the next layer. The neurons in the input layer only send signals to the next layer.

The functions are very simple for addressing the interactions between neurons. If p equals a sender neuron in the input layer and q is a receiver neuron in the next layer, the collection function is given as

$$net_q = \sum_p w_{pq} I_p \tag{1}$$

where I_p is the signal from neuron p of the sender layer, net_q is the collection signal for receiver neuron q in the next layer, and W_{pq} is the parameter or weight to sum the signals from different input nodes.

The receiver neuron creates activation in response to the signal net_q. The activation will become the input for its next layer. The activation is usually created in the form of a sigmoid function: i.e.,

$$\frac{1}{1 + e^{-net_q}}. \tag{2}$$

The activation will be passed to the next layer as the input signal, and Equations 1 and 2 will be used to process the signal again. These procedures continue until the final signals reach the output layer.

Parameters (weights) play a crucial role in determining the final signals. A back-propagation learning algorithm

(Rumelhart *et al.*, 1986; Foody, 1996) has been popularly used to obtain the optimal parameter values. The algorithm iteratively minimizes an error function over the network (calculated) outputs and desired outputs based on a training data set. The most apparent advantage of the back-propagation neural network is that the learning algorithm is not programmed, *a prior*, into the network (Hepner, 1990). The weights are first initially set by a random process. The errors computed as the difference between calculated and desired activation for the output neuron is propagated back through the network and used to adjust the weights. The process to adjust weight according to the errors will be repeated by many iterations until the error rate reaches an acceptable level.

Conventionally, the overall output error is defined as half the overall sum-of-the-squares of the output errors, which, for the k^{th} training pattern, is (Foody, 1996; Zhou and Civco, 1996)

$$E_k = \frac{1}{2} \sum_{q=1}^{m} (O_{kq} - D_{kq})^2 \tag{3}$$

where O_{kq} is the calculated network output, O_{kq} is the desired output for neuron q, and m is the number of neurons in the output layer of the network.

The accumulated error for all training patterns is

$$E = \sum_{k=1}^{l} E_k \tag{4}$$

where l is the total number of training patterns.

At each iteration, back-propagation recursively computes the gradient or change in error by an amount proportional to the accumulated $\partial E/\partial W$: i.e.,

$$\Delta W = -\eta \, \partial E/\partial W \tag{5}$$

where η is the learning rate.

The learning process in neural networks is able to make predictions that are as close as possible to the desired values for a set of training data. The prediction surface is distinctly non-linear, which is much superior to the linear surface of popular regression models (Lloyd, 1997) (Figure 1).

Simulating Development Alternatives for Urban Planning Using Neural Networks

An artificial neural network (ANN) can be devised to estimate development probability based on the inputs of site attributes. The advantages of an ANN are quite appealing because it can effectively reveal the complex relationships between site attributes and urban growth. In conventional urban CA models, development probability at a central cell is usually estimated according to its neighborhood conditions (attractiveness), such as developed quantity (density) and proximity variables. Development probability is proportional to the combined score of various types of attractiveness. Wu and Webster (1998) have provided an interesting example for the experiment using multicriteria evaluation (MCE) techniques.

The proposed model consists of two separate parts—training and simulation (Figure 2). The training is based on the back-propagation procedure which can generate optimal weights from a set of training data. Remote sensing and GIS data are used to provide the empirical data to reveal the relationships between site attributes and urban development. A number of spatial variables can be defined as the inputs (site attributes) to the neural network for estimating development probability. In the model, development probability is determined by the site attributes of each cell. It is very convenient to obtain the site attributes within a GIS. GIS functions can be employed to obtain the basic information for urban simulation. These site attributes usually include developed quantity in the neighborhood, and various types of proximity attractiveness (Batty and Xie, 1994; Wu and Webster, 1998).

The proposed simulation model is based on the algorithm of neural networks. A neural network is designed to estimate development probability at each iteration of a CA simulation. The neural network has three layers: one input layer one hidden layer, and one output layer. The input layer has n neurons to represent these site attributes: i.e.,

$$\mathbf{X} = [x_{ij1}, x_{ij2}, \ldots, x_{ijk}, \ldots, x_{ijn}]^{\mathrm{T}} \tag{6}$$

where x_{ijk} is the kth spatial variable at site ij, and T is the transposition function.

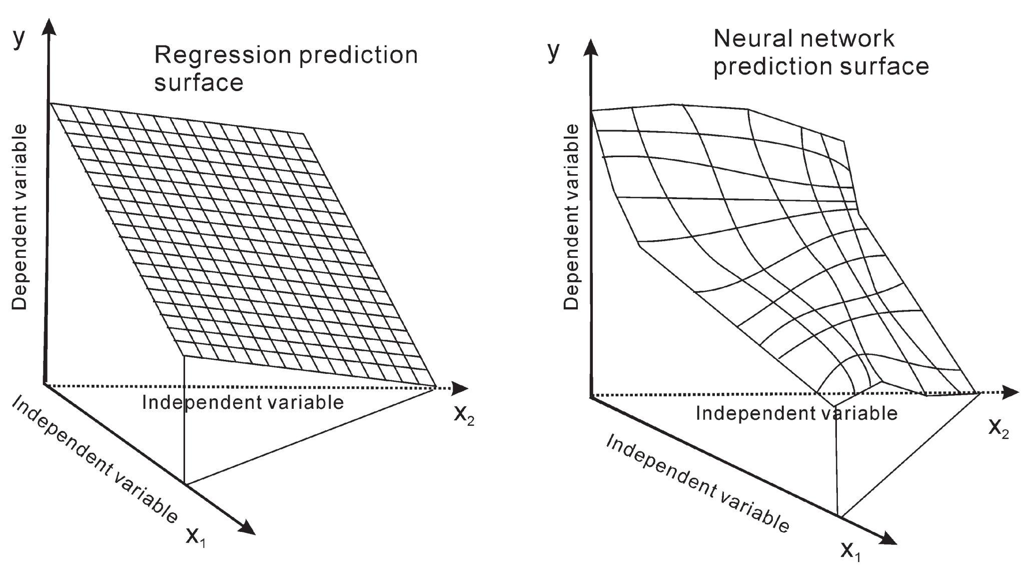


Figure 1. Comparison of prediction surfaces between regression models and neural networks.

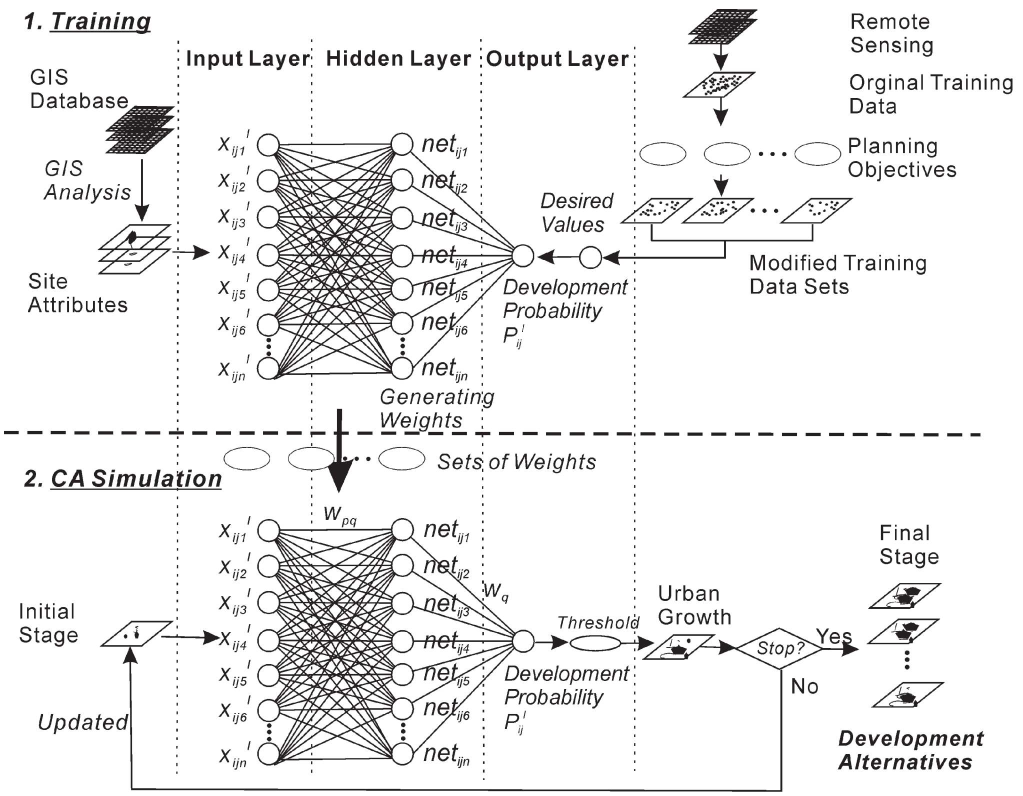


Figure 2. Simulation of development alternatives using neural-network-based CA.

The hidden layer may also have n neurons. The output layer has only one neuron which calculates development probability. At each iteration, the site attributes of a cell will be input into the first layer, and the neural network will determine its development probability at the output layer.

The original data are usually standardized by scaling them into the range of [0, 1] before they are input into neural networks (Gong, 1996). Scaling these variables can treat them as equally important inputs to neural networks and makes them compatible with a sigmoid activation function that produces a value between 0.0 and 1.0. The following linear transformation is used for the standardization:

$$x'_{ij} = (x_{ij} - x_{ijmin})/(x_{ijmax} - x'_{ijmin}). \tag{7}$$

The algorithm of the CA model is based on a neural network. In the neural network, neuron q in the hidden layer receives the signals from all the neurons in the first input layer. The received signal is calculated by

$$net_{ijq}(t) = \sum_p w_{pq} x'_{ijp}(t) \tag{8}$$

where ij is a cell, $net_{ijq}(t)$ is the signal received by neuron q in the hidden layer for cell ij at time t, and $x'_{ijp}(t)$ is the site attributes for variable (neuron) p in the input layer.

The activation of the received signal in the hidden layer is

$$\frac{1}{1 + e^{-net_{ijp}(t)}}. \tag{9}$$

The development probability $P_{ij}(t)$ for cell ij is then calculated by the output neuron: i.e.,

$$P_{ij}(t) = \sum_q w_q \frac{1}{1 + e^{-net_{ijp}(t)}}. \tag{10}$$

This simulation is loop-based. At each iteration, development probability is calculated by the neural network. The development probability decides whether a cell is converted or not for urban development. A stochastic disturbance term can be added to represent unknown errors in calculating the probability. This can allow the simulated patterns to be more like realistic. The error term (RA) is given by the following equation (White and Engelen, 1993):

$$RA = 1 + (-\ln \gamma)^{\alpha} \tag{11}$$

where γ is a uniform random variable within the range of [0, 1] and α is a parameter to control the size of stochastic perturbation. The development probability is revised as follows:

$$\begin{aligned} P'_{ij}(t) &= \mathrm{RA} \times \sum_q w_q \frac{1}{1 + e^{-net_{ijp}(t)}} \\ &= (1 + (-\ln \gamma)^{\alpha}) \times \sum_q w_q \frac{1}{1 + e^{-net_{ijp}(t)}}. \end{aligned} \tag{12}$$

A cell with a higher value of development probability will be more likely to be developed during the simulation. A predefined threshold value can be used to decide whether a cell is developed or not at each iteration. If a cell has the probability greater than the threshold value, it will be converted for development. The amount of already developed cells in the neighborhood is recalculated, and the site attributes are updated at the end of each iteration. The iterations will stop when the desired number of developed cells is reached.

The proposed neural-network-based CA planning model is capable of simulating urban development based on current prevailing trends. More importantly, it is also capable of simulating development alternatives through the incorporation of planning objectives in preparing the training data. These two purposes can be achieved by using different training data sets to train the neural network. First, for the simulation of urban development based on current prevailing trends, original data can be directly used to train the neural network for generating realistic simulation without planning intervention. This assumes that urban growth continues along the path of historical trends.

Second, for the simulation of development alternatives based on planning objectives, it is expected that the model should have the ability to adjust itself from past experience. This can be done by modifying the original data set to create alternative training data sets based on different planning objectives. Original data can be assessed and modified so that the past "undesirable" development would not enter the training process. This can allow the model to generate future land development based on the planning objectives of planners and decision makers. "Desirable" and "undesirable" development points can be identified in terms of their development costs and benefits. Some criteria can be used to facilitate the assessment. However, there are no unique ways for defining these criteria. In most situations, these criteria are related to planning objectives.

A simple way for data modification is to redefine some "undesirable" points according to the planning objective to create a new data set. For example, in the original training data, there are some developed sites converted from good-quality agricultural land. They should not be developed under the planning objective of agriculture-conservation. If the original data are used to train the neural network, the land-use problem will be inherited in the simulation. Therefore, these developed cells should be adjusted to non-developed cells in the modified training data set under this planning objective. For this planning objective, all developed cells in the original training data set with agricultural suitability score greater than a certain threshold (e.g., 0.8) will be considered as unsuitable for development. The development value of their cells will be adjusted from 1 to 0 accordingly, where 1 indicates developed cell and 0 undeveloped cell. This adjustment can allow the neural network to obtain the proper adjusted weights for the simulation according to this planning objective. Adjusted weights are obtained when the modified training data set is used to train the neural network. The simulation can then avoid the encroachment on good agricultural land in the simulation of development alternatives for future urban growth. The modification of the training data set through the incorporation of planning objectives can help the neural network to minimize urban development on land which is suitable for agricultural development and prevent the same problems from happening again in the simulation. The whole procedure is simple and robust because the parameter values are not arbitrarily determined.

Urban planning usually requires the preparation of various development alternatives under different assumptions. Wu and Webster (1998) propose a method for generating alternative development patterns by integrating multicriteria evaluation (MCE) techniques with CA. Parameter values can be adjusted corresponding to various planning objectives. There are uncertainties in this method because the determination of parameter values is quite relaxed. The proposed model is more robust because the strict back-propagation algorithm is used for the training. The parameter values can be automatically obtained through the proper training of neural networks.

Applications

Study Area and Site Attributes

The neural-network-based CA planning model was applied to a fast growing city, Dongguan, for the simulation of development alternatives in the region. Situated in the eastern part of the Pearl River Delta, China, the city has a total land area of 2,465 km^2. It has had a large scale of urban expansion in recent years according to the monitoring of remote sensing (Yeh and Li, 1997; Li and Yeh, 1998). The existing development patterns are unacceptable for sustainable development because of many land-use problems. This study is to simulate development alternatives under various planning objectives. The simulation of possible development alternatives is useful for urban and regional planning. It can help to prevent existing land-use problems from happening again in the future through urban planning.

Remote sensing and GIS data were used to provide the basic inputs to the model. In this study, seven spatial variables are defined to represent the site attributes of each cell. These variables include

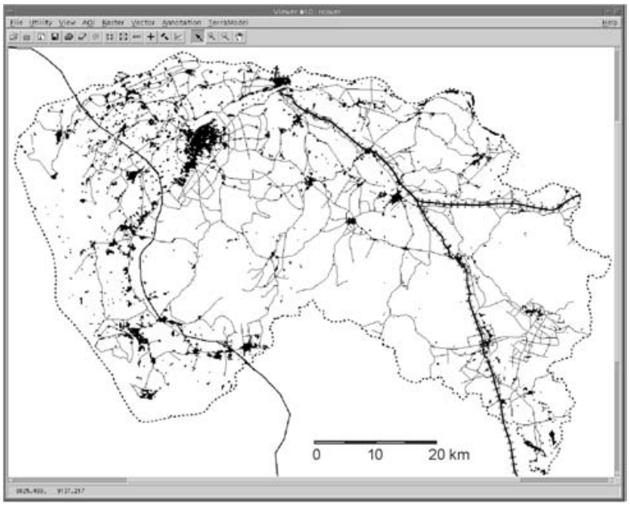
- Distance to the major (city proper) urban areas S_1,
- Distances to suburban (town) areas S_2,
- Distance to the closest road S_3,
- Distance to the closest expressway S_4,
- Distance to the closest railway S_5,
- Neighborhood development quantity (a window of 7 by 7 cells) S_6, and
- Agricultural suitability S_7.

The selection of these variables is based on knowledge and experience (Wu and Webster, 1998; Li and Yeh, 2002). These variables are important factors in deciding development probability. This can be confirmed by regression analysis. It is not difficult to obtain these variables by employing GIS analyses and remote sensing classification. The distance variables were calculated using the *Eucdistance* function of ARC/INFO GRID. These distance variables were dynamically updated during the simulation. The neighborhood development quantity was measured by counting the number of developed cells in the neighborhood of the 7 by 7 cells adjacent to the central cell. The *Focal* function of ARC/INFO GRID was used to count the number of developed cells at each iteration. This variable was dynamically updated during the simulation. The initial development quantity (the number of developed cells) in the neighborhood was calculated from the 1988 binary image.

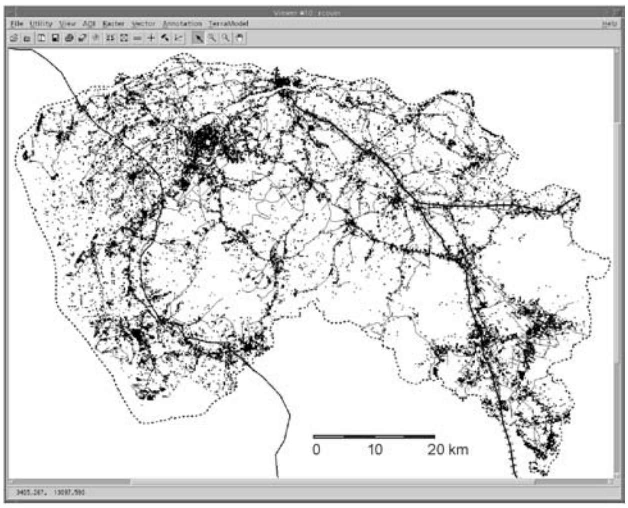
Training

The neural network must be trained to obtain the parameter values so that the simulation can be executed. The training data were from the classification of the 1988 and 1993 satellite TM images. The classification provides the empirical information about urban development in the period (Figures 3a and 3b). The classification results were imported to ARC/INFO GRID in grid format as training data. Although the original TM images had a ground resolution of 30 by 30 m, the cell size was reduced to 50 by 50 m by a resampling procedure for faster simulation. The total number of cells is 588 by 776. Data encoding was carried out for the training data. The overlay of the two images reveals where urban development has taken place. The urban areas were 16,234.6 ha in 1988, but they increased to 41,087.9 ha in 1993 according to the classification of the satellite TM images. The value of 1 was assigned to the developed (converted) cells and 0 to the non-developed cells.

An essential task is to design the network structure for urban simulation. The design of the network structure is quite relaxed because the numbers of layers and neurons in the layers can be rather subjectively determined. A three-layer neural network has been commonly used in applications. Kolmogorov's theorem suggests that any continuous function $\phi: X^n \to R^c$ can be implemented by a three-layer neural network which has n neurons in the input layer, $2n + 1$ neurons in the single hidden layer, and c nodes in the output layer (Wang, 1994). Two to three hidden layers may be useful for the cases of extreme non-linear features. However, the increase in



(a)



(b)

Figure 3. (a) Initial urban areas of Dongguan in 1988 from the classification of the satellite TM. Image. (b) Urban areas of Dongguan in 1993 from the classification of the satellite TM image.

the numbers of layers and neurons will drastically increase the computation time for the loop-based CA model. It is practical to limit the numbers of layers and neurons to as few as possible without severely compromising model accuracy. De Villiers *et al.* (1992) also suggest that a neural network with one hidden layer may be more preferable than one with two hidden layers in terms of learning speed and performance.

A three-layer network is most suitable for the CA model of many iterations. Practically, $2n + 1$ neurons in the single hidden layer may seem to be too much for actual applications. Experiments also indicate that a network of $2n/3$ neurons in the hidden layer can generate results of almost the same accuracy level but requires much less time to train than that of $2n + 1$ neurons (Wang, 1994). In this study, the input layer has seven neurons to represent the seven variables of the site attributes. The hidden layer also has seven neurons. The output layer has only one neuron to output the development probability. There are $7 \times 7 = 49$ weights to be determined for the links between the input layer and the hidden layer, and

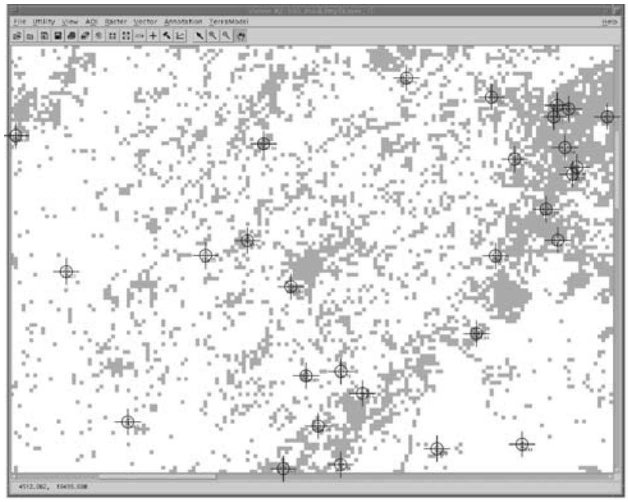


Figure 4. Random points of stratified sampling for obtaining training data sets.

seven weights between the hidden layer and output layer. A total of 56 parameters were used for the neural-network-based CA model.

The parameters' values were automatically determined by the training process. Training the network is done to find the optimal parameter values that can ensure the best fit between the calculated and expected values. The connected strength (weight) between each pair of linked neurons, which is used in the CA model, has to be estimated from the training process. The training was implemented by using a neural network package, THINKS PRO.[1]

Original training data were obtained by the overlay of the historical urban growth (1988–93) from remote sensing with the site attributes from the GIS. It is inappropriate to use the whole data set for training because the size is too large and the data may have spatial correlation. A random sampling procedure was carried out to reduce data volume and data redundancy. Figure 4 shows the examples of the random points using a stratified sampling procedure. These points were generated by the ERDAS IMAGINE[2] package. Their coordinates were then imported to ARC/INFO GRID for the retrieval of the site attributes using the *Sample* function. One-thousand random points were obtained as the training data to train the neural network.

In the training data, the desired value (the observed value) is associated with a number of site attributes for each cell. It is expected that the calculated value based on the site attributes should be as close as possible to the desired value. The training process was accomplished by the dynamic adjustment of the network interconnection strengths (weights) so that the error between the desired and the calculated values can be minimized. The whole set of sampling points was equally divided into two groups for the training process. One group was the training data set while the other group was the test data set. The training data set was used to obtain the weights for each link between a pair of neurons. The test data set was further used to verify the training results.

As discussed in the methodology section, there are two ways to use empirical data for training the network. First, original data can be directly used to train the neural network

[1]THINKS PRO is a trademark of Logical Designs Consulting, Inc.
[2]ERDAS IMAGINE is a trademark of ERDAS, Inc.

for generating a realistic simulation without planning intervention. This assumes that urban growth continues along the path of historical trends. Land-use problems will be inherited from the data because of the training. Second, it is expected that the model should have the ability to adjust itself from past experience. This can be done by providing the proper training data sets. Original data can be assessed and modified to avoid the influence of past land-use problems on the training process. The new set of modified training data can lead the training procedure to obtain new sets of parameter values. These new sets of parameter values will then be used by the neural network to generate development alternatives. This can help to rectify past land-use problems and lead to healthy urban growth in the future according to the planning objectives of planners and decision makers. The whole procedure is simple and robust because the parameter values are not arbitrarily determined.

Simulation Results

After appropriate weights had been obtained by the training, they were imported into the neural-network-based CA model for the simulation. The model was implemented by the integration of neural network and GIS. The model was developed in ARC/INFO[3] GRID using the Arc Macro Language (AML). The GIS package provides powerful spatial handling functions that are useful for CA simulation. It also allows an easy access to the GIS database for obtaining site attributes for the simulation.

The urban areas of 1988 were used as the starting point of simulation. It simulated development alternatives from 1988 to 1993. For comparison purposes, it was assumed that the same amount of land would be used in different development alternatives. This was set to 24,853.3 ha, which was equal to the actual total amount of land that had been used for urban development from 1988 to 1993. The simulating models would stop when this total amount was reached.

In each iteration, the output neuron of the network generates the development probability of each cell according to Equation 12. The developed probability is then compared with a threshold value to decide if a cell is developed or not. The threshold value was calibrated by actual land consumption. This is to guarantee that the simulated development will have the same amount of land consumption as that of actual development or planning development at the final output. The parameter α, the random disturbance, was set to 1 so that only a small amount of uncertainty was presented in the simulation.

It is easy to create a couple of alternative training data sets from the original data set based on the assessment of the past development. The modification is related to planning objectives. For example, the sites with a distance greater than 30 km from town centers are considered to be unsuitable for development under the town-based development objective. If a developed cell in the original sampling data set has a distance greater than the threshold, its desired value (development or not) should be adjusted from 1 to 0. For the agriculture conservation objective, all developed cells with an agricultural suitability score greater than 0.8 will be considered as unsuitable for development. Their desired values should be adjusted from 1 to 0 accordingly. Table 1 lists four typical planning objectives and their modification criteria (rules) that are used for preparing the alternative training data sets.

After these new data sets had been created based on the above modification criteria (rules), they were used to train the network. The training procedure was carried out by using the THINK PRO package. Normal weights were obtained using the original data according to Planning Objective 1. Adjusted weights were obtained using the modified data according to Planning Objectives 2, 3, and 4. Different sets of weights were then input to the CA to generate alternative urban growths, respectively.

Table 1. Rules for Creating Alternative Training Data Sets According to Different Planning Objectives

Planning Objectives	Modification Rules
(1) Urban growth according to past development trend	Use original data without any modification
(2) Promotion of monocentric development	Change the desired values from 1 to 0 for all the developed cells if $S_1 > 200$
(3) Promotion of polycentric development	Change the desired value from 1 to 0 for all the developed cells if $S_2 > 30$
(4) Promotion of agriculture-conservation development	Change the desired value from 1 to 0 for all the developed cells if $S_7 > 0.8$

The experiment indicates that training the network can be directly linked to the emergence of discriminated development patterns. Figure 5a is the simulation output using the normal weights from the original training data set. It is totally based on the past growth trend without any modification. As a result, the simulated pattern is very close to that of the actual development (Figure 3b). The effectiveness of the simulation result can be easily examined by using visual comparison of the simulated urban development (Figure 5a) with actual urban development (Figure 3b). Visual comparison may sometimes be more useful for validating CA because spatial patterns cannot be easily measured (Clarke *et al.*, 1997; Ward *et al.*, 2000). However, it is also possible to examine the goodness-of-fit by overlaying the simulated urban development (Figure 5a) on the actual urban development (Figure 3b) to produce the cross-tabulation. The overall accuracy is 0.79, which is pretty good.

The model is able to simulate various development alternatives by using the modified training data sets to train the network according to different planning objectives. Figure 5b shows the effects of promoting mono-centric development around the existing city proper. The simulation is based on the adjusted weights from the modified training data set under Planning Objective 2. Figure 5c is the simulation for promoting poly-centric development around the existing 29 town centers. The adjusted weights are obtained from Planning Objective 3. Figure 5d is the simulation results for protecting agricultural land resources. It can control the encroachment on the best agricultural land by applying the fourth set of adjusted weights according to Planning Objective 4. It can be found that land development can be well restricted in the fertile alluvial plain which is situated in the northwestern part of the region.

The simulation of development alternatives is important to urban and regional planning. There is very limited work on how to calibrate urban simulation, especially on how to derive development alternatives in an objective way. Many existing models use a heuristic approach to determine parameter values. The above experiment shows that, by training the neural network with modified training data sets according to different planning objectives, different development alternatives can be generated. Using this method, it is quite convenient to explore possible development alternatives under various planning objectives. Users are not required to define parameter values because they can be automatically obtained through proper training. The neural-network-based CA planning model can save much time in finding parameter values. In conventional CA, the determination of model structures, transition

[3]ARC/INFO is a trademark of Environmental Systems Research Institute, Inc.

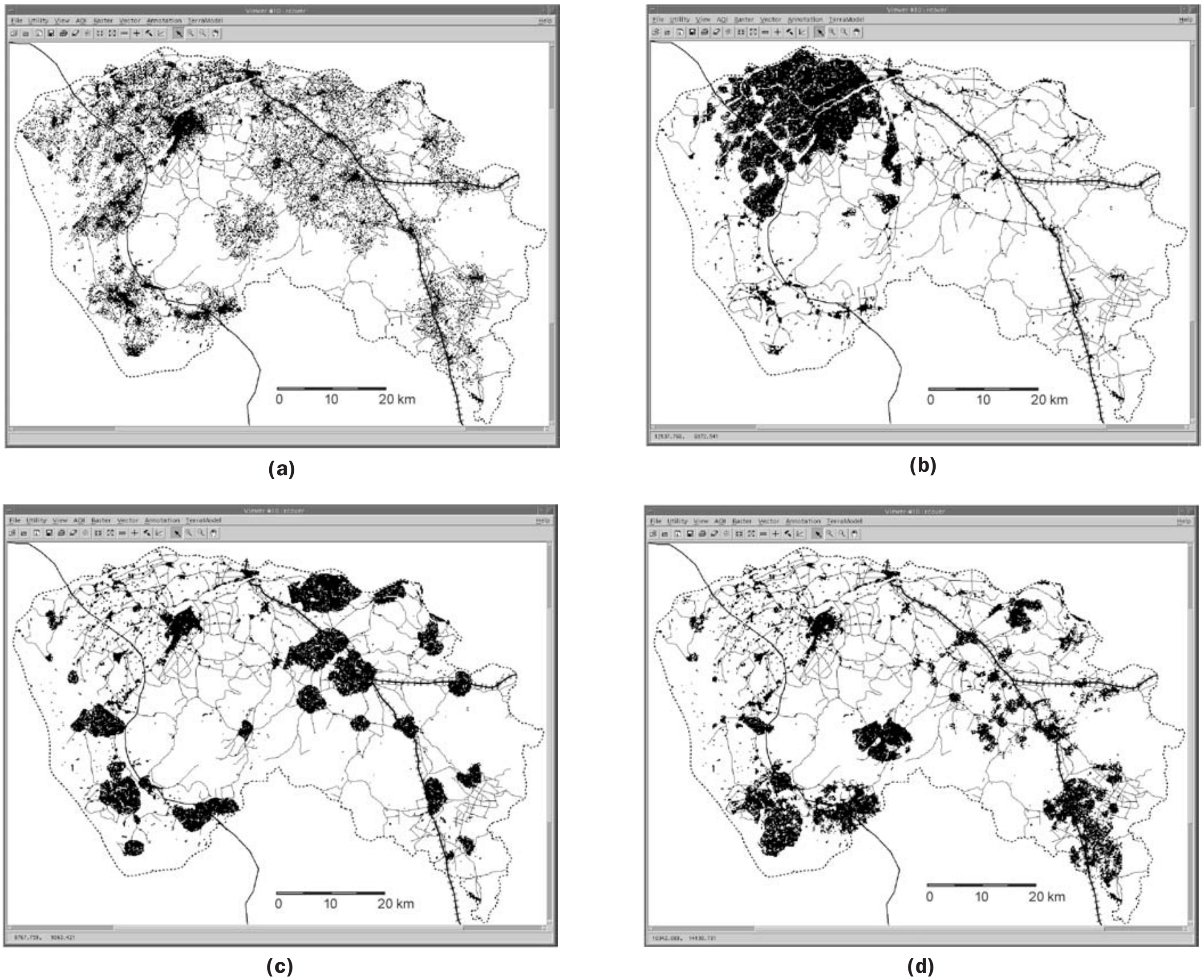


Figure 5. (a) Simulation of urban development based on the past development trend (Planning Objective 1) using normal weights. (b) Simulation of mono-centric development (Planning Objective 2) using adjusted weights. (c) Simulation of poly-centric development (Planning Objective 3) using adjusted weights. (d) Simulation of urban development for agriculture-conservation (Planning Objective 4) using adjusted weights.

rules, and parameter values is very time-consuming. These jobs may take a much longer time than the simulation itself.

Conclusion

This study proposes a new method to simulate development alternatives by the integration of neural networks, cellular automata, and GIS. The preparation of various development scenarios is important in urban and regional planning. The simulation can help planners and decision makers to test and compare what can be gained under different planning objectives, such as "smart" urban growth and sustainable urban development. Although CA models have been increasingly used in urban simulation, there are very limited studies in applying them to urban planning. The proposed method can conveniently generate development alternatives through appropriate training of neural networks according to different planning objectives.

The use of neural networks can greatly simplify the structure of CA models. The neural-network-based model significantly reduces the requirements for explicit knowledge for identifying relevant criteria, assigning scores, and determining criteria preference. The model can effectively map the non-linear features of urban systems because of the use of neural networks. Moreover, parameter values for urban simulation can be automatically obtained by properly training the network. The training procedure is simple because of the use of the back-propagation algorithm of neural networks. Training data can be obtained from the classification of remote sensing imagery and GIS analyses. However, it is inappropriate to use the original training data to train the network if the objective is to find a better development plan because past land-use problems can propagate through the training procedure. The assessment and modification of training data according to different planning objectives can avoid this problem. Land development can be assessed based on development costs and benefits.

In this study, the original training data set is assessed to identify which developed cells are "desirable" or "undesirable" in terms of development costs and benefits. Data modification is then carried out by using criteria (rules) which are defined according to planning objectives. New training data sets can be created for training the network to obtain different sets of parameter values. The parameter values are then input to the model to simulate development alternatives for future urban growth. The training procedure can accommodate interventions in the simulation process and lead to the formation of alternative urban growth.

The neural-network-based CA is not difficult to implement in a raster GIS environment. The use of a GIS can allow the model to gain access to a rich source of spatial information. Spatial variables retrieved from the GIS are used as the main inputs to the neural network for deciding development probability. These variables are dynamically updated during the simulation process. The simulation is also benefited by using the powerful functions of spatial analysis and visualization of GIS.

An inherent problem with neural networks is that they are black-box in nature. The meanings of the parameter values are difficult to explain because the relationships among neurons are quite complex. Moreover, the determination of network structure is also subject to user's preferences. It seems that there is no way to determine what is the best network structure. Further studies can be carried out to examine the sensitivity between network structure and simulation results.

There is another problem concerning the transport factor in urban simulation. Urban development patterns are always influenced by the layout of transport systems in a region. The dynamics of transport systems should be considered in urban CA. However, there is difficulty in predicting the future changes of transport systems. In this study, the simulation assumes that the transport system (railways and roads) in the region remained unchanged during the period of simulation. This is because the transport system in the study area has just been upgraded and should remain stable over the short term. Future studies should consider the possibility of transport changes and accommodate this factor in the simulation. There are substantial uncertainties in simulating the dynamics of transport systems because changes are more subject to planning and political factors. Therefore, it is more reasonable to tackle this issue by considering future transport development as an exogenous factor. A GIS provides a tool for planners to input the transport information into the urban simulation. It is better to treat the transport factor outside the simulation process instead of predicting the change within the model. CA can thus simulate various possible development alternatives for different transport network proposals.

Acknowledgments

This study is supported by the funding from the Hong Kong Research Grants Council (Contract No. HKU 7209/02H), Hong Kong SAR, PR China.

References

Batty, M., and Y. Xie, 1994. From cells to cities, *Environment and Planning B: Planning and Design*, 21:531–548.

———, 1997. Possible urban automata, *Environment and Planning B: Planning and Design*, 24:175–192.

Batty, M., Y.C. Xie, and Z.L. Sun, 1999. Modeling urban dynamics through GIS-based cellular automata, *Computer, Environment and Urban Systems*, 23:205–233.

Binder, P., 1989. Evidence of Lagrangian tails in a lattice gas, *Cellular Automata and Modeling of Complex Physical Systems* (P. Manneville, N. Boccara, G.Y. Vichniac, and R. Bidaux, editors), Springer-Verlag, Berlin, Germany, pp. 155–160.

Clarke, K.C., P. Riggan, and J.A. Brass, 1995. A cellular automata model for wildfire propagation and extinction, *Photogrammetric Engineering & Remote Sensing*, 60:1355–1367.

Clarke, K.C., L. Gaydos, and S. Hoppen, 1997. A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area, *Environment and Planning B: Planning and Design*, 24:247–261.

Clarke, K.C., and L.J. Gaydos, 1998. Loose-coupling a cellular automata model and GIS: Long-term urban growth prediction for San Francisco and Washington/Baltimore, *International Journal of Geographical Information Science*, 12(7):699–714.

Couclelis, H., 1997. From cellular automata to urban models: New principles for model development and implementation, *Environment and Planning B: Planning and Design*, 24:165–174.

De Villers, J., and E. Barnard, 1992. Backpropagation neural nets with one and two hidden layers, *IEEE Transactions on Neural Networks*, 4(1):136–141.

Fischer, M.M., and S. Gopal, 1994. Artificial neural networks: A new approach to modelling interregional telecommunication flows, *Journal of Regional Science*, 34:503–527.

Foody, G.M., 1996. Relating the land-cover composition of mixed pixels to artificial neural network classification output, *Photogrammetric Engineering & Remote Sensing*, 62:491–499.

Goles, E., 1989. Cellular automata, dynamics and complexity, *Cellular Automata and Modeling of Complex Physical Systems* (P. Manneville, N. Boccara, G. Y. Vichniac, and R. Bidaux, editors), Springer-Verlag, Berlin, Germany, pp. 10–20.

Gong, P., 1996. Integrated analysis of spatial data from multiple sources: Using evidential reasoning and artificial neural network techniques for geological mapping, *Photogrammetric Engineering & Remote Sensing*, 62:513–523.

Hägerstrand, T., 1965. A Monte-Carlo approach to diffusion, *European Journal of Sociology*, 6:43–67.

Hepner, G., 1990. Artificial neural network classification using a minimal training set: Comparison to conventional supervised classification, *Photogrammetric Engineering & Remote Sensing*, 56:469–473.

Li, X., and A.G.O. Yeh, 1998. Principal component analysis of stacked multi-temporal images for monitoring of rapid urban expansion in the Pearl River Delta, *International Journal of Remote Sensing*, 19(8):1501–1518.

———, 2000. Modelling sustainable urban development by the integration of constrained cellular automata and GIS, *International Journal of Geographical Information Science*, 14(2):131–152.

———, 2001. Calibration of cellular automata by using neural networks for the simulation of complex urban systems, *Environment and Planning A*, 33:1445–1462.

———, 2002. Urban simulation using principal components analysis and cellular automata for land-use planning, *Photogrammetric Engineering & Remote Sensing*, 68(4):341–351.

Lloyd, R., 1997. *Spatial Cognition: Geographic Environments*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 287 p.

Openshaw, S., 1993. Modelling spatial interaction using a neural net, *GIS, Spatial Modelling and Policy* (M.M. Fischer and P. Nijkamp, editors), Springer, Berlin, Germany, pp. 147–164.

———, 1998. Neural network, genetic, and fuzzy logic models of spatial interaction, *Environment and Planning A*, 30:1857–1872.

Openshaw, S., and C. Openshaw, 1997. *Artificial Intelligence in Geography*, John Wiley & Sons, Chichester, United Kingdom, 329 p.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams, 1986. Learning representations by back-propagating errors, *Nature*, 323:533–536.

Tobler, W.R., 1979. Cellular geography, *Philosophy in Geography* (S. Gale and G. Olssen, editors), Reidel, Dordrecht, The Netherlands, pp. 379–386.

Wang, F., 1994. The use of artificial neural networks in a geographical information systems for agricultural land-suitability assessment, *Environment and Planning A*, 26:265–284.

Ward, D.P., A.T. Murray, and S.R. Phinn, 2000. A stochastically constrained cellular model of urban growth, *Computers, Environment and Urban Systems*, 24:539–558.

Webster, C. J., and F. Wu, 1999. Regulation, land-use mix, and urban performance. Part 2: Simulation, *Environment and Planning A*, 31:1529–1545.

White, R., and G. Engelen, 1993. Cellular automata and fractal urban form: A cellular modelling approach to the evolution of urban land-use patterns, *Environment and Planning A*, 25: 1175–1199.

White, R., G. Engelen, and I. Uijee, 1997. The use of constrained cellular automata for high-resolution modelling of urban land-use dynamics, *Environment and Planning B: Planning and Design*, 24:323–343.

Wu, F., and C.J. Webster, 1998. Simulation of land development through the integration of cellular automata and multicriteria evaluation, *Environment and Planning B: Planning and Design*, 25:103–126.

Wu, F., 1998. An experiment on the generic polycentricity of urban growth in a cellular automatic city, *Environment and Planning B: Planning and Design*, 25:103–126.

———, 2000. A parameterised urban cellular model combining spontaneous and self-organising growth, *GIS and Geocomputation* (P. Atkinson and D. Martin, editors), Taylor & Francis, New York, N.Y., pp. 73–85.

Yeh, A.G.O., and X. Li, 1997. An integrated remote sensing and GIS approach in the monitoring and evaluation of rapid urban growth for sustainable development in the Pearl River Delta, China, *International Planning Studies*, 2(2):193–210.

———, 2001. A constrained CA model for the simulation and planning of sustainable urban forms using GIS, *Environment and Planning B: Planning and Design*, 28:733–753.

Zhou, J., and D. Civco, 1996. Using genetic learning neural networks for spatial decision making in GIS, *Photogrammetric Engineering & Remote Sensing*, 62:1287–1295.