Environment and Planning A 2001, volume 33, pages 1445 – 1462

DOI:10.1068/a33210

Calibration of cellular automata by using neural networks for the simulation of complex urban systems

Xia Li¶, Anthony Gar-On Yeh
Centre of Urban Planning and Environmental Management, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, PR China;
e-mail: lixia@graduate.hku.hk, hdxugoy@hkucc.hku.hk
Received 22 November 2000; in revised form 14 May 2001

Abstract. This paper presents a new cellular automata (CA) model which uses artificial neural networks for both calibration and simulation. A critical issue for urban CA simulation is how to determine parameter values and define model structures. The simulation of real cities involves the use of many variables and parameters. The calibration of CA models is very difficult when there is a large set of parameters. In the proposed model, most of the parameter values for CA simulation are automatically determined by the training of artificial neural networks. The parameter values from the training are then imported into the CA model which is also based on the algorithm of neural networks. With the use of neural networks, users do not need to provide detailed transition rules which are difficult to define. The study shows that the model has better accuracy than traditional CA models in the simulation of nonlinear complex urban systems.

1 Introduction

Cellular automata (CA) are powerful spatial dynamic modeling techniques that have been widely applied to model many complex dynamic systems. Recently, a variety of urban CA models have been developed to simulate either artificial or realistic cities (Batty and Xie, 1994; Clarke et al, 1997; Li and Yeh, 2000; White and Engelen, 1993; Wu, 1998; Wu and Webster, 1998). Cities, like most geographical phenomena, are complex nonlinear systems involving spatial and sectoral interactions which cannot easily be modeled with the functionalities of current GIS software (Batty et al, 1999). CA-based approaches are useful in the study of urban and regional spatial structure and evolution. Modeling cities with CA is a new approach, although it has distant roots in geography in the work of Hägerstrand (1965) and Tobler (1979) (Clarke and Gaydos, 1998).

A critical issue in CA simulation is the provision of proper parameter values or weights so that realistic results can be generated. Real cities are complex dynamic systems that require the use of many spatial variables in CA simulation. Each spatial variable makes a contribution to the simulation and its influence is determined by its associated parameter or weight in the simulation. A variable associated with a larger parameter value usually indicates that it is more important than other variables with small parameter values. There are usually many parameter values to be defined in a CA model and the results of CA simulation are very sensitive to the parameter values (Wu, 2000).

Empirical data can be used to calibrate CA models to find suitable parameter values. Calibration is important to the generation of the best fit to actual urban development. Calibration procedures have been discussed in nonlinear dynamic spatial interaction models (Lombardo and Rabino, 1986). Unfortunately, at present there is no good CA calibration method because of the complexity of urban systems. There are only very limited studies addressing calibration issues in CA simulation. Wu and Webster (1998) use multicriteria evaluation (MCE) to define the parameter values for CA simulation heuristically. Calibration can also rely largely on repetitive runs of the

¶ Also at Guangzhou Institute of Geography, Guangzhou, PR China; e-mail: xlib@gis.sti.gd.cn

same model with different combinations of parameter values (Wu, 2000). Clarke et al (1997) consider visual tests to be useful to establish parameter ranges and to make rough estimates of these values. The impact of each parameter is assessed by changing its value and holding other parameters constant.

Clarke and Gaydos (1998) have provided a more elaborate calibration method by statistically testing the observed against the expected. The method is to find which set of parameter values can lead the model to produce the best fits. The set of parameter values with the best fits is then used for prediction. However, there are numerous possible combinations of parameter values. Their experiments have tried more than 3000 combinations which need a high-end workstation running several hundred hours to perform the calibration. This method is very time consuming because it needs to compare all possible combinations of parameter values. Another problem is that the combinations may be extraordinarily huge when there are many variables and it is hard to develop a sound search procedure for the calibration.

In this study we present a new method based on neural networks to deal with the complicated calibration issue in CA urban simulation. It has a number of advantages over other methods. The calibration procedure is simple and convenient because the parameter values are obtained automatically by using neural networks. The method is made more robust by using the well-developed procedure of back-propagation training. Moreover, the model is able to deal with complex interactions among variables. Variables are not required to be independent of each other because a neural network is used. The model structure is much simpler and more stable compared with traditional CA models. This paper will also demonstrate that a neural-network-based CA model can easily be developed within a GIS environment. Spatial variables can be conveniently retrieved from the GIS and used as inputs to the neural network for urban simulation.

2 Artificial neural networks (ANN)

Artificial neural networks (ANN) have been widely and seemingly extremely successfully applied in many disciplines with a high degree of difficulty (Openshaw and Openshaw, 1997). ANN are quite adapted to spatial problems that may involve wrong and poor data. The following are some of the advantages of neural networks that have been identified by Openshaw and Openshaw (1997) and Openshaw (1998):

(a) The structure of algorithms enables neural networks to be robust and noise resistant regardless of poor data.
(b) They can solve highly nonlinear problems in complex systems.
(c) The method is rather simple because no exact equations or expressions are required.
(d) The best level of performance can be obtained.
(e) There are no restrictions about using nonnumeric data.
(f) They adapt to nonnormal frequency distribution.
(g) Mixtures of measurement types can be used.
(h) They can use many variables some of which may be redundant.

The basic processing units in a neural network are the so-called neurons or nodes which are organized in a couple of layers. All the neurons, except those in the input layer, perform two simple processing functions—collecting the activation of the neurons in the previous layer and generating an activation as the input to the next layer. The neurons in the input layer only send signals to the next layer.

The functions for addressing the interactions between neurons are very simple. If i equals a sender neuron in the input layer and j is a receiver neuron in the next layer,

the collection function is given as:

$$\text{net}_j = \sum_i W_{i,j} I_i \,, \tag{1}$$

where I_i is the signal from neuron i of the sender layer, net_j is the collection signal for receiver neuron j in the next layer, and $W_{i,j}$ is the parameter or weight to sum the signals from different input nodes. The receiver neuron creates activation in response to the signal net_j. The activation will become the input for its next layer. The activation is usually created in the form of a sigmoid function:

$$1 + \frac{1}{\exp(-\text{net}_j)}. \tag{2}$$

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 will continue until the final signals are obtained by the output layer.

The crucial part of neural networks is to determine the adaptive weights which are used to address the strengths of network interconnection between associated neurons. The values of the weights are not set by the users but rather are determined by the network during training. One of the most popular methods is a back-propagation learning algorithm which iteratively minimizes an error function over the network (calculated) outputs and desired outputs on the basis of a training data set (Foody, 1996; Rumelhart et al, 1986). The clearest advantage of the back-propagation neural network is that the learning algorithm is not programmed, a priori, into the network (Hepner, 1990). The weights are initially set by a random process. The error, 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 of adjusting the weights according to the errors is repeated over many iterations until the error rate is minimized or reaches an acceptable level.

Once the optimized weights have been obtained from the training data set, the network is ready for classification or prediction. Classification or prediction is based on the activation level in the output layer. The activation level of a neuron lies on a scale of 0 to 1 which reflects the variation from extremely low to extremely high strength of membership. In the classification, for example, a case will be allocated to the class associated with the output neuron with the highest activation level.

3 The artificial neural-network-based CA model (ANN-CA)

In this model, the transition rules of CA simulation are represented by the neural network calibrated by the empirical data. The essential part of the transition rules in general CA models is to estimate the conversion probability between states. For example, a simple urban CA model can be expressed by the following neighborhood-based transition rules (Batty, 1997):

if any cell $\{x \pm 1, y \pm 1\}$ is already developed,

then $P_\text{d}\{x, y\} = \sum_{ij \in \Omega} P_\text{d}\{i, j\}/8$, and

if $P_\text{d}\{x, y\} >$ some threshold value,
then cell $\{x, y\}$ is developed with some other probability $\rho\{x, y\}$,

where $P_\text{d}\{x, y\}$ is the urban development probability for cell $\{x, y\}$, and cell $\{i, j\}$ is the set of all the cells which are from the Moore neighborhood Ω including the cell $\{x, y\}$ itself. The probability of a cell being developed is decided by the number of

already developed cells in the neighborhood. Usually, there is a higher chance of a cell being developed if it is surrounded by more developed cells.

Simulation based on a single factor of developed cells in the neighborhood cannot address complicated urban systems. More factors have been incorporated in CA models to improve simulation performance for either hypothetical or realistic applications. Distance, direction, density thresholds, and transition or mutation probabilities have been included in the transition rules of various CA models (Batty and Xie, 1994; 1997). Various types of constraints based on site features can also be used to regulate development patterns for land-use planning (Li and Yeh, 2000; Yeh and Li, 2001a). An increase in the number of variables will result in an increase of the number of parameters in CA models. There are few concerns about the exact values of parameters if CA models are used only for hypothetical studies. However, when CA models are applied to the simulation of real cities, suitable parameter values have to be determined through some calibration procedures.

MCE techniques are frequently required to handle multiple factors in CA simulation. For example, a linear weighted combination of a series of spatial factors (such as various distances to urban centers and transportation lines, and the amount of development in the neighborhood) have been used to estimate development probability for CA simulation (Wu, 1998; Wu and Webster, 1998). MCE may be useful when weights cannot be determined by using empirical data. There are uncertainties because the weights are heuristically or subjectively defined in most situations. No strict calibration procedure is involved in the method. Moreover, MCE will face difficulties in determining the values of the weights when there are many factors. The method also cannot represent complicated nonlinear surfaces. It has been suggested that traditional methods of overlay and MCE can be replicated or even replaced by neural networks (Zhou and Civco, 1996).

The ANN-CA model consists of two separate parts—using the neural network to obtain the parameter values automatically based on training data, and using the neural network to carry out CA simulation based on these parameter values (figure 1). The first part involves the calibration (training) procedure to obtain optimal weights using empirical data. Remote sensing data are used to provide empirical data about urban development. GIS analysis is needed to obtain site attributes (attractiveness) for each cell. Neural networks can be used to reveal the relationships between development probability and site attributes. The training procedure should be outside the simulation process for computation efficiency. The parameters obtained from the training will be input to the CA model. The second step is to carry out CA simulation which is also based on the algorithm of neural networks. At each iteration, neural networks will determine the development probability which is subject to the input of site attributes and weights.

Site attributes determine development probability in CA simulation. These attributes include the amount of development in the neighborhood and various types of proximity attractiveness (Wu and Webster, 1998) and GIS is useful for obtaining this type of information. The integration of GIS and modeling can be especially useful in solving issues arising from environmental problems (Clarke and Gaydos, 1998; Li and Yeh, 2000). CA models can be easily integrated with GIS to generate realistic simulations. Operational CA models need access to real databases for better simulation performance. GIS provides powerful functions to store, retrieve, analyze, and display spatial data. Information about constraints can be retrieved from GIS databases and input into CA models for generating rational development patterns and achieving urban sustainability (Li and Yeh, 2000; White et al, 1997).

A series of spatial variables can be easily defined on the basis of the buffer and neighborhood functions of GIS software. A cell may have n site attributes (variables):

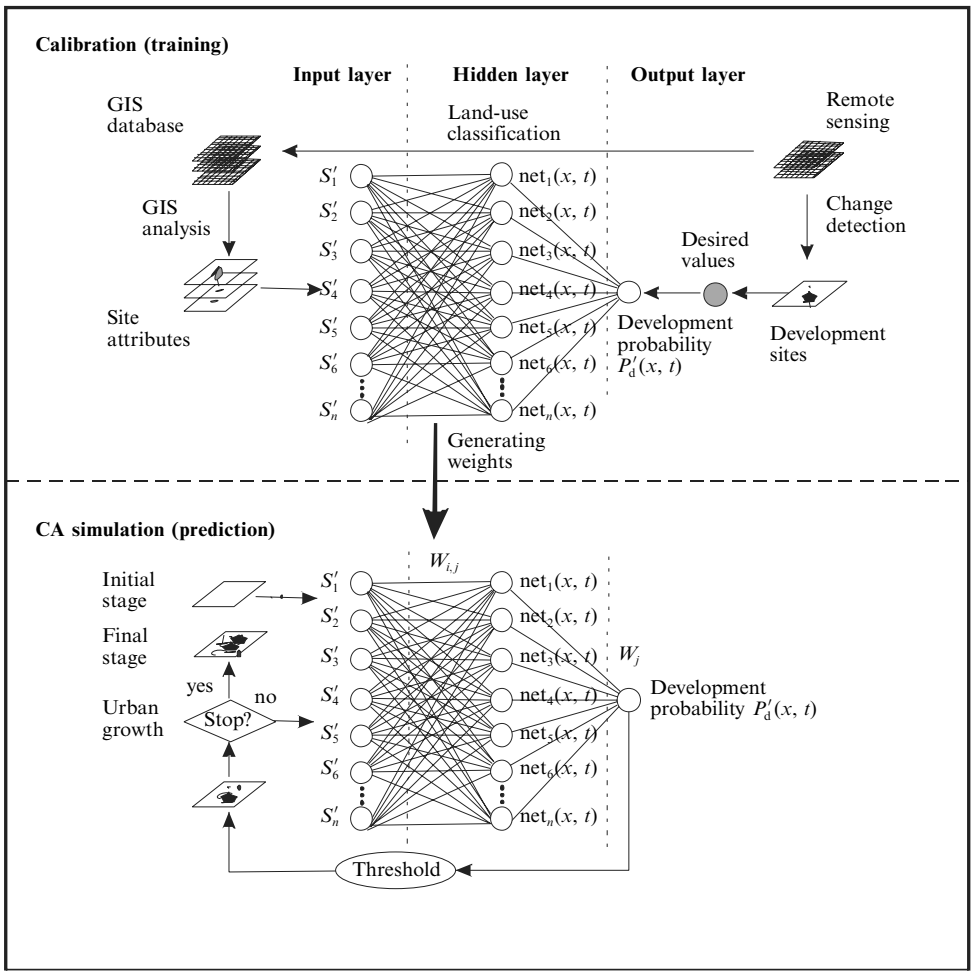


Figure 1. A dynamic cellular automaton (CA) model based on neural networks.

$$(S_1, S_2, S_3, S_4, S_5, S_6, ..., S_n)\,. \tag{3}$$

Urban growth is dependent on these variables which may include various types of proximities, amount of development, and site conditions. A regression model or MCE method may not be the best way to reveal relationships because urban systems involve complex nonlinear processes. Neural networks can be designed to estimate development probability at each iteration of the CA simulation. The neural network may have three layers: one input layer, one hidden layer, and one output layer. The input layer has n neurons corresponding to the n variables. The hidden layer may also have n neurons. The output layer has only one neuron which indicates 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.

Experiments indicate that it will be more appropriate for all original data to be converted into a range between 0 to 1 before they are input into neural networks (Gong, 1996). This is similar to data normalization in that it uses maximum and minimum values in scaling the original data set. Scaling variables treats them as equally important inputs to neural networks and makes them compatible with the

sigmoid activation function that produces a value between 0 and 1. The following linear transformation is used:

$$S_i' = \frac{S_i - \text{minimum}}{\text{maximum} - \text{minimum}}. \quad (4)$$

GIS and remote sensing can be used to obtain the training data set. An easy way is to obtain land-use change information from the classification of multitemporal satellite images (Li and Yeh, 1998). In the training data set, the desired (target) value in the output layer is recorded as 1 for a developed cell and 0 for a nondeveloped cell. The network output value will be obtained after a set of site attributes has passed through the network. The output value is expected to be close to the desired value according to the learning process by automatically adjusting the weights. The output value is within the range 0 to 1. When the network is used for prediction, the output value can be regarded as a development probability. During the prediction, an output value closer to 1 indicates higher development probability and one closer to 0 indicates lower development probability. The learning process can effectively let the network estimate development probability using a set of site attributes.

A new type of CA model can be derived based on the algorithm of neural networks. The advantage is that the parameters which have been obtained automatically from the training procedure can be applied directly to the ANN-CA simulation. Users do not need to define the parameters manually. This simulation is loop based. At each iteration, the site attributes are sent through the network to obtain the development probability of a cell. The development probability is then used to determine if the cell is selected for development or not according to a predefined threshold value.

The algorithm for the CA model is devised using a neural network. In the neural network, the signal received by neuron j of the hidden layer from neuron i of the first input layer for cell x is calculated from:

$$\text{net}_j(x, t) = \sum_i W_{i,j} S_i'(x, t), \quad (5)$$

where x is a cell, $\text{net}_j(x, t)$ is the signal received for neuron j of cell x at time t, and $S_i'(x, t)$ is the site attributes for variable (neuron) i. The activation of the hidden layer for the signal is:

$$\frac{1}{1 + \exp[-\text{net}_j(x, t)]}. \quad (6)$$

The development probability (P_d) for cell x is then calculated from:

$$P_\text{d}(x, t) = \sum_j W_j \frac{1}{1 + \exp[-\text{net}_j(x, t)]}. \quad (7)$$

A stochastic disturbance term can be added to represent unknown errors during the simulation. This can generate patterns that are closer to reality. The error term (RA) is defined as (White and Engelen, 1993):

$$\text{RA} = 1 + (-\ln \gamma)^\alpha, \quad (8)$$

where γ is a uniform random variable within the range 0 to 1, and α is the parameter controlling the size of the stochastic perturbation; α can be used as a dispersion factor in the simulation. The development probability is revised as:

$$P'_{\mathrm{d}}(x,t) = \mathrm{RA}\sum_j W_j \frac{1}{1+\exp[-\mathrm{net}_j(x,t)]}$$
$$= [1+(-\ln\gamma)^{\alpha}]\sum_j W_j \frac{1}{1+\exp[-\mathrm{net}_j(x,t)]}. \quad (9)$$

Finally, a predefined threshold value is used to decide whether a cell is to be developed or not according to the probability $P'_{\mathrm{d}}(x,t)$ at each iteration. If a cell has a probability greater than the threshold value, it will be converted for development. The number of already developed cells in the neighborhood is recalculated and the site attributes are updated at the end of each iteration. The number of iterations is determined by the total land consumption in a given time period. The simulation will stop when all the required number of cells have been converted into urban development.

4 Implementation and simulation results

4.1 Simulation platform

In this dynamic ANN-CA model, neural networks play two rather separate roles—calibration and simulation. The training or learning process was carried out outside the CA model. Training the network is most important for successful applications of neural networks. The connected strength (weight) between each pair of linked neurons, which are used in the CA model, has to be estimated from the training process. This was carried out by using a neural network package, ThinksPro,[1] which contains sophisticated algorithms and convenient interfaces for effective training and visualization.

After appropriate parameter values have been obtained, they are imported into the CA model for urban simulation. The simulation is still based on neural networks. The actual simulation model is implemented in a GIS platform by the integration of neural network, CA, and GIS. GIS facilitate the convenient access to the spatial data which are used as site attributes for the simulation. The model was programmed in ARC/INFO[2] GRID using the Arc Macro Language (AML). The GIS package also provides the buffering and overlay geoprocessing functions that are useful for CA simulation.

4.2 Remote sensing and GIS for obtaining site attributes

A real city, Dongguan, was selected to test the neural-network-based CA model. It has a land area of 2465 km^2, situated in the Pearl River Delta in southern China which has witnessed rapid urban expansion and land-use changes in recent years. The simulation of urban development and land-use change is useful for urban and regional planning. The model can be used to simulate future development for assessing the environmental impacts of urban development if the current dynamics and development pattern prevail in the future. Satellite images were used to provide basic urban development information to train and test the model.

Land-use classification was carried out on the satellite TM images of 1988 and 1993. The classification results were imported to ARC/INFO GRID in the grid format. These grids were used as the empirical data for the calibration of the CA model. Although the original TM images had a ground resolution of 30 × 30 m, the cell size was resampled to 50 × 50 m for faster simulation. Binary values were used to represent developed and nondeveloped areas in 1988 and 1993. A value of 1 represents developed (converted) cells and 0 represents nondeveloped. The urban areas of 1988 were used as the starting point of the simulation (figure 2, see over). The urban areas of 1993 (figure 3) were also obtained for testing the simulation by comparing the actual development with the result of simulated development.

[1] ThinksPro is a trademark of Logical Designs Consulting, Inc., La Jolla, CA.

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

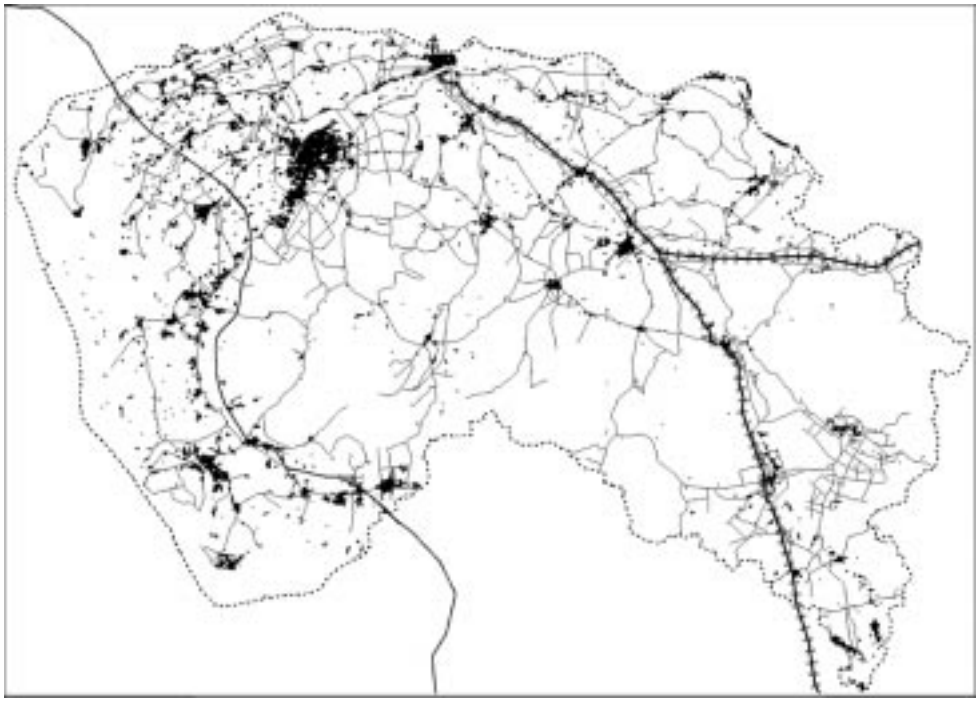


Figure 2. Initial urban areas of 1988 for simulation classified from satellite TM image.

Studies show that some distance-based variables are closely related to urban development in the region and can be used for CA simulation (Li and Yeh, 2000; Wu and Webster, 1998; Yeh and Li, 2001a). Satellite images can also indicate that development sites are usually located along major transportation lines or around an existing urban area. In this study, seven spatial variables were defined to represent the site attributes of each cell for the simulation of urban development:
(1) distance to the major (city proper) urban areas S_1;
(2) distances to suburban (town) areas S_2;
(3) distance to the closest road S_3;
(4) distance to the closest expressway S_4;
(5) distance to the closest railways S_5;
(6) amount of development in the neighborhood (a window of 7×7 cells) S_6;
(7) agricultural suitability S_7.

The distance variables were calculated with the Eucdistance function of ARC/INFO GRID. The simulation will be more precise when the distances are measured from existing urban areas rather than from urban centers, because growing urban areas will generate more infrastructure and additional centers to support further urban growth. These distance variables were dynamically updated during the simulation.

Amount of development was measured using the neighborhood of 7×7 cells adjacent to the central cell. The Focal function of ARC/INFO GRID was used for the calculation. This variable was also dynamically updated during the simulation. The initial amount of development (the number of developed cells) in the neighborhood was calculated from the 1988 binary image.

A problem is how to determine the weights or parameter values for these variables in CA simulation. Estimation becomes much more difficult when the number of spatial variables increases. However, the weights can be calculated automatically when a

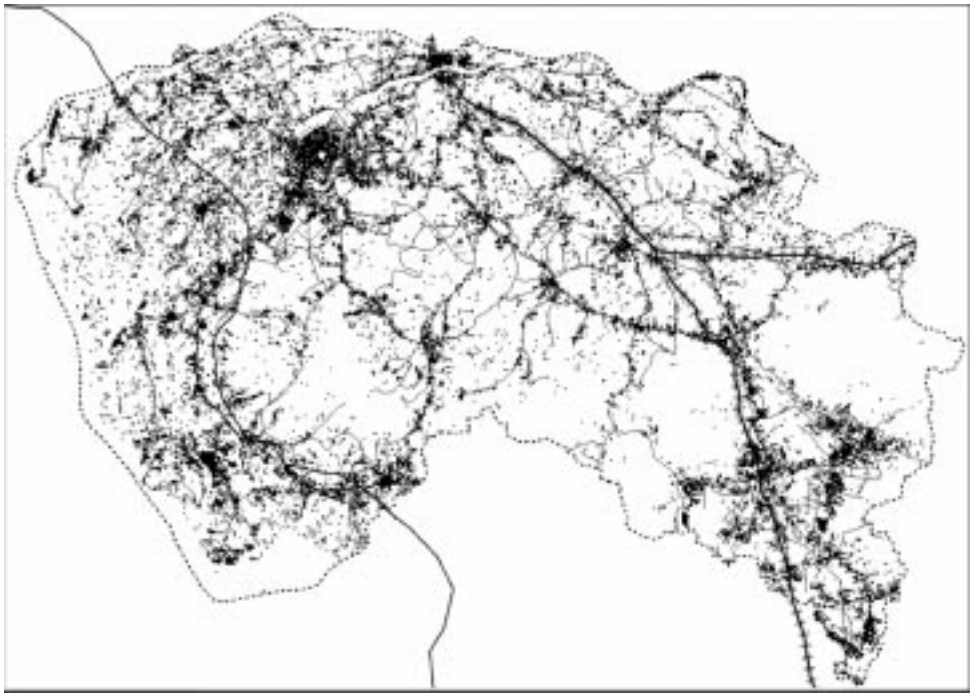


Figure 3. Actual urban areas of 1993 classified from satellite TM image that is used to verify the result of the ANN-CA simulation.

neural network is used. No specific attention should be paid to which variable should be selected in a neural network. The principle is to use all the spatial variables that are identified as related to urban development. A neural network can deal with correlated or redundant variables by itself.

4.3 Network structure

An essential task is to design the network structure for the CA 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. However, an increase in the numbers of layers and neurons will drastically increase the computation time for the loop-based CA model. The principle is to use as few layers and neurons as possible without severely compromising model accuracy.

Kolmogorov's theorem indicates that any continuous function ϕ: $X^n \rightarrow \mathbb{R}^c$ 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 (Hecht-Nielsen, 1987; Wang, 1994). De Villiers and Barnard (1992) also suggest that a neural network with one hidden layer may be preferable to one with two hidden layers in terms of learning speed and performance. 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 but requiring much less time to train than that of $(2n + 1)$ neurons (Wang, 1994).

Therefore, it is appropriate to use three layers of neural network for the CA simulation. The input layer has seven neurons corresponding to the seven site attributes chosen for the study. 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 7 weights between the hidden layer and output layer. A total of 56 parameters were used for the neural-network-based CA model. The values of these parameters were automatically determined by the learning process which is based on the back-propagation algorithm using the ThinksPro package.

4.4 Calibration (training)

GIS overlay analysis was carried out in ARC/INFO GRID to obtain the training data set. Land development in 1988–93 was overlaid with the seven layers of site attributes. The overlay provides the training data that can reveal the relationship between site attributes and development probability. However, the relationship is rather complicated because of the nonlinear development of urban systems. Neural networks are most suitable to predict the development probability based on site attributes for nonlinear systems.

It is inappropriate to use the whole data set for training. A sampling procedure was carried out by using a stratified sampling method (Congalton, 1991), which ensures that sampling effort can be distributed in a rational pattern so that a specific number of observations are assigned to each category to be evaluated. Figure 4 shows examples of the random stratified sampling points which were generated by the ERDAS IMAGINE[3] package. Their coordinates were then imported to ARC/INFO GRID for the retrieval of the site attributes that were associated with these sampling points using the Sample function. There were 600 random sampling points, which were used to train the neural network.

The training process was accomplished by the dynamic adjustment of the network interconnection strengths (weights) so that the error between the desired and calculated

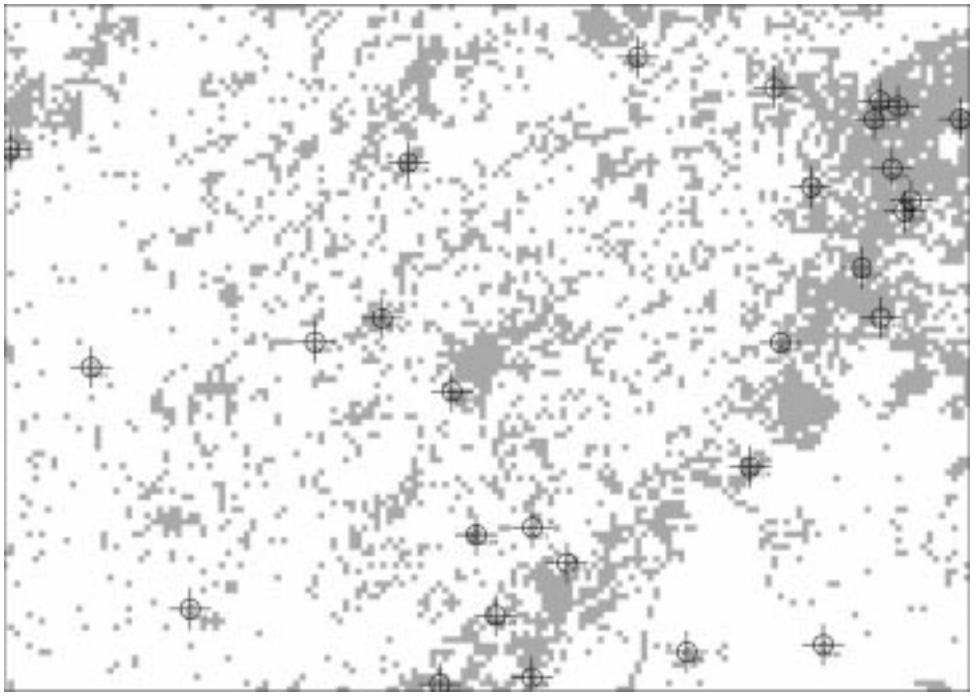


Figure 4. Random stratified sampling points for obtaining training data set.

[3] ERDAS IMAGINE is a trademark of ERDAS, Inc., Atlanta, GA.

can be minimized. The whole set of sampling points was divided into two equal groups for the learning process of neural networks. One group is the training data set whereas another group is 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 learning results. Table 1 shows examples of the training data set and the calculated development probability from the ThinksPro package. A set of weights was finally obtained from the training procedure.

One of the most important characteristics of trained neural networks is their ability to generalize from the training data. If the network simply memorizes or overfits the training data, it will generally perform poorly on the test data. ThinksPro can graphically show when a network is overtraining by the use of the 'test while training' option. The test data set provides the means of validating the network performance. The use of weight-minimization and network-growing methods available in ThinksPro greatly increases the ability of the network to generalize. It is important to decide the number of iterations so that the training can be stopped properly. If training process too long, the network may be overtrained and cause large prediction errors for the rest of the data. This problem was solved by stopping the training when the error for the test data began to increase.

4.5 Simulation

The simulation was based totally on neural networks. At each iteration, the development probability of each cell was obtained from the neural network. Cells with a

Table 1. Examples of site attributes, desired values (actual), and calculated development probability from artificial neural networks (ANN).

Site attributes							Desired value[a]	Output value from ANN (development probability)
S_1	S_2	S_3	S_4	S_5	S_6	S_7		
201	10	0	18	256	14	0.4	0	0.079
82	8	1	38	152	21	0.2	1	0.815
82	38	3	8	169	19	0.6	1	0.606
173	5	1	172	64	31	0.6	0	0.135
170	2	0	199	33	20	0.6	0	0.069
169	1	1	199	33	21	0.6	0	0.074
99	25	16	38	190	14	0.2	1	0.512
166	3	2	196	31	10	0.2	0	0.082
139	3	1	33	222	26	0.4	0	0.455
105	20	0	14	192	26	0.6	1	0.608
169	3	1	209	3	23	0.6	0	0.089
96	24	3	23	169	20	0.2	1	0.746
94	30	1	9	180	17	0.6	1	0.493
91	8	3	147	2	21	0.2	1	0.693
154	27	1	38	231	4	0.6	0	0.049
140	19	1	29	227	26	0.6	0	0.304
69	3	1	10	161	19	0.2	1	0.864
69	34	0	117	40	20	0.6	1	0.560

[a] 0 means not developed; 1 means developed.

development probability greater than a predefined value were converted into developed cells. It is straightforward to set the predefined value at 0.5 according to the training results (table 1). However, CA simulation usually involves many iterations to decide whether a cell is converted or not. It is better to raise the threshold value so that urban growth is fulfilled step-by-step. A value of 0.5 is too low because too many cells are converted at each iteration. Experiments indicate that a predefined value of 0.75 can

produce better simulation results but that higher values can significantly increase simulation time. At the end of each iteration, the amount of development in the neighborhood was then recalculated to update the site attributes.

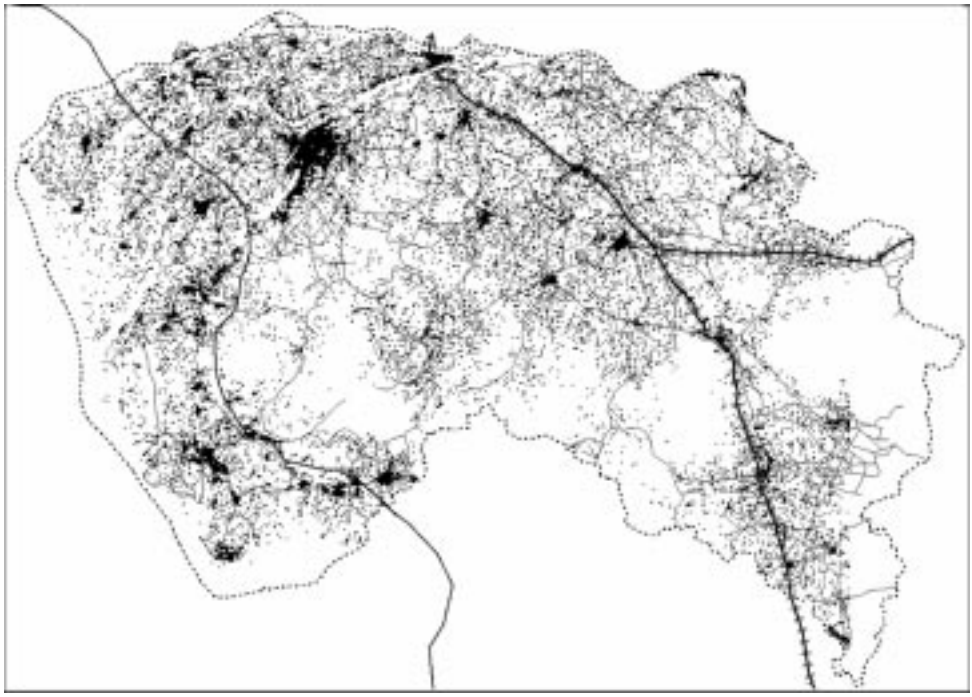
The first step was to simulate the urban development in 1988 – 93 using the proposed neural-network-based model. This can be compared with the actual urban development that is detected from remote sensing. The initial urban areas for the simulation were from the 1988 classified TM image. The simulation time was automatically determined to ensure that the amount of land conversion was finally equal to that of the actual development in 1988 – 93. A disturbance term can be used in CA simulation to represent stochastic perturbation (White and Engelen, 1993). Like other statistical models, the network cannot capture stochastic perturbation by training. There is no model that can produce the exact development pattern because of the uncertainties and complexities of the real world. However, different values of parameter α can be used to explore possible urban forms that are related to uncertainties. Higher values of parameter α can lead to the urban areas becoming more chaotic or dispersed in the simulation. It is found that the simulation can produce a pattern similar to the actual development in 1988 – 93 using a higher value of disturbance ($\alpha = 3$) [figure 5(a)]. The actual development is quite dispersed. As an alternative, a more compact pattern can be simulated using a lower value of the dispersion factor ($\alpha = 1$) [figure 5(b)]. This indicates that the actual land development in the region is influenced by some unexplained variables. It is also confirmed by other studies that the region is characterized by a chaotic development pattern because of severe land speculation (Yeh and Li, 2001b).

The second step was to predict possible future urban development in 1993 – 2003 for planning purposes. The initial urban areas were from the classified 1993 TM image. The prediction of future urban development assumed that the same growth trend will continue. The amount of land consumption for the simulation was calculated on the assumption that the average annual growth rate of the urban areas in 1993 – 2003 was the same as that in 1988 – 93. The same set of parameter values was used to predict future urban development. Figure 6(a) (over) is the projection of the future urban areas of Dongguan in 2003 using the same set of parameter values as those for the simulation of the similar pattern of 1988 – 93 ($\alpha = 3$). It was found that the urban areas are quite dispersed. There is a need to promote more compact development. Figure 6(b) (over) is the result of using a smaller value for the dispersion factor ($\alpha = 1$).

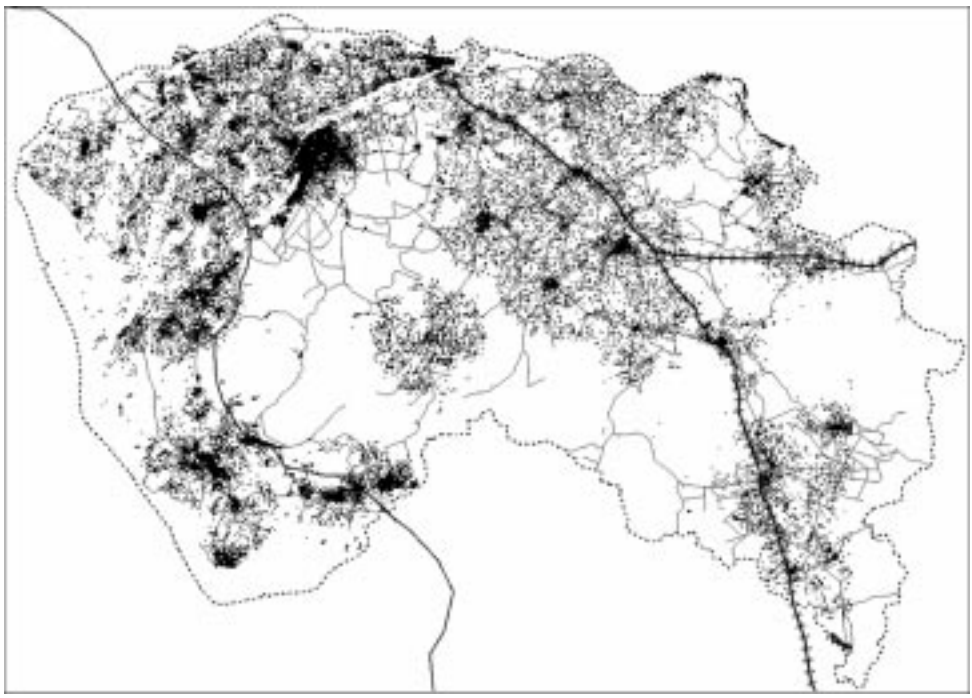
The model provides a tool directly linking training to the simulation of realistic urban development. It is also possible to simulate future development and generate alternative patterns for planning purposes. The model is much more convenient than other CA models because users do not need to define most of the parameter values for simulation. The simulation can provide useful information and implications for land-use planning and management.

One problem with this prediction is that the changes in infrastructure such as transportation networks have not been considered because of uncertainty. For example, it is not easy to forecast where a new road will be built. The changes in transportation layout can significantly influence development patterns. There is a need to accommodate these changes although there are difficulties. The problem also occurs in other CA models. One possible solution is to consider these changes as exogenous factors that can be imported into the model from outside rather than simulating them inside. For simplicity this model does not consider these exogenous factors at this stage. Further studies are needed to develop a planning model based on neural networks which can deal with a variety of uncertainties and exogenous factors.

The study demonstrates that the neural-network-based CA model is not difficult to implement. The algorithm may be slightly more complicated than a traditional CA

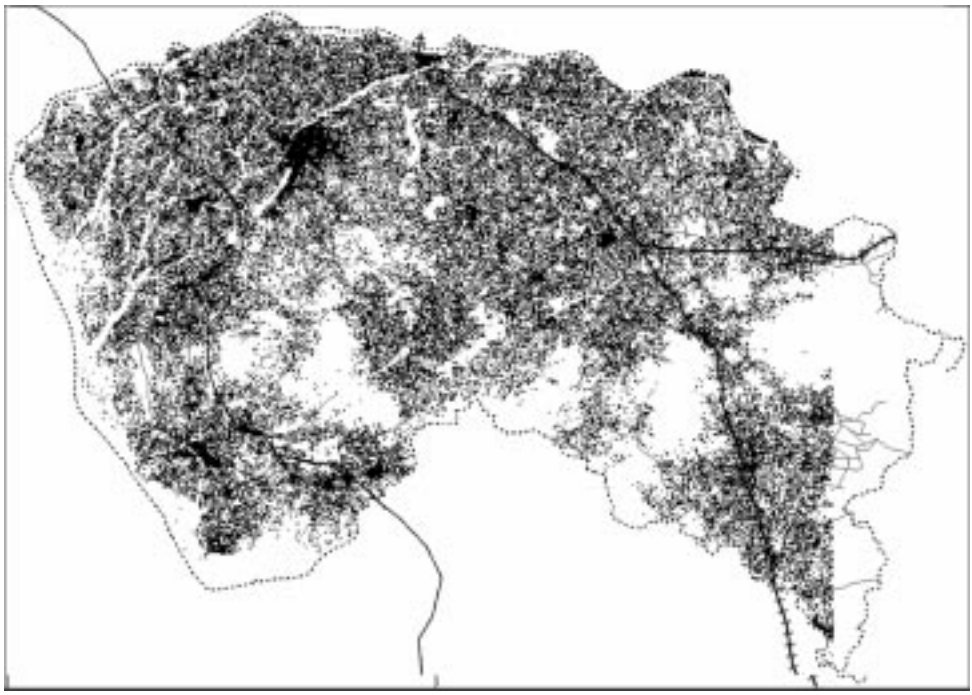


(a)

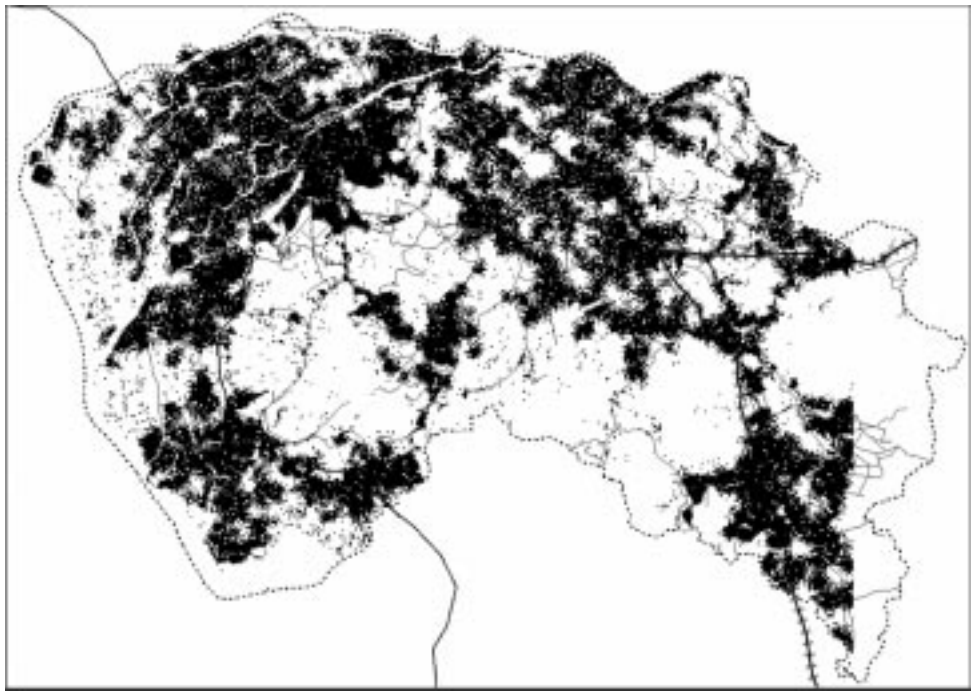


(b)

Figure 5. Simulation of urban areas of 1993 using the ANN-CA model based on the initial stage of 1988: (a) dispersion factor $\alpha = 3$; (b) dispersion factor $\alpha = 1$.



(a)



(b)

Figure 6. Simulation of urban areas of 2003 (future) using the ANN-CA model based on the initial stage of 1993: (a) dispersion factor $\alpha = 3$; (b) dispersion factor $\alpha = 1$.

model because it uses a three-layer network. The experiment shows that total simulation time is not substantially longer than that of traditional CA models. The simulation time is about 5 – 10 minutes longer than that of a previous model using the same data set (Li and Yeh, 2000). However, the neural-network-based CA model can save much time in calibration which may need a high-end workstation running hundreds of hours for the blind trial (Clarke and Gaydos, 1998). In this model, only about 2 – 3 minutes are needed to obtain the parameter values from the calibration procedure of a neural network package. Neural networks are more robust for obtaining weights than traditional MCE methods because neural networks are strictly based on a back-propagation training procedure.

The experiment also indicates that the neural-network-based CA model is better at simulating complex nonlinear systems. Wu (1998) suggests that a logistic regression model could be used to estimate development probability for the integrated MCE and CA simulation. The same sampling data set was used to compare the accuracies of the two methods. Table 2(a) is the error matrix for the prediction of actual urban development in 1993 using the neural network method from the training data. The overall accuracy is 0.79. Table 2(b) is the error matrix for the prediction based on the logistic regression model. The overall accuracy is 0.73. Moreover, the logistic regression model may have multicollinearity problems when the variables are highly correlated. The logistic model also has great difficulties in handling nonlinear functions.

Another important feature of the ANN-CA model is that the transition rules of CA simulation are represented by the neural network. This means that users do not need to design transition rules. In traditional CA models, users may face great difficulties in choosing transition rules, so a variety of transition rules have been proposed from different CA models. The model proposed here does not require explicit modeling methods and strict data sources. The main task is to provide training data so that the neural network can learn and generalize from them. Therefore, neural-network-based CA models are very suitable for the simulation of complex urban systems.

Table 2. Prediction accuracy of urban development in 1988 – 93 from (a) the ANN-CA model; (b) the logistic regression model.

Observed	Predicted		Percentage correct
	nonconverted	converted	
(a) ANN-CA model			
Nonconverted	99	30	0.77
Converted	24	102	0.81
Overall percentage			0.79
(b) Logistic regression model			
Nonconverted	134	22	0.86
Converted	48	60	0.56
Overall percentage			0.73

5 Conclusion

This study has developed a new type of CA model to simulate urban development using artificial neural networks. General CA models have problems in obtaining consistent parameter values when there are many variables in the simulation. It is very time consuming to define parameter values for promoting the 'best-fits'. This paper has demonstrated that neural networks can be conveniently used to solve the problems of calibration. In the proposed method, most of the parameter values

required for CA simulation are automatically determined by the training procedures rather than by the users.

The algorithm is quite simple and the simulation time is not very long. The proposed method can significantly reduce the time needed for calibration. The traditional methods may need to run a calibration model repetitively for hundreds of hours by using different combinations of parameters before finding a suitable set of values. The total time for calibration and simulation can be substantially reduced by the proposed model.

Moreover, the transition rules of CA simulation are represented by a neural network. This means that users can overcome the problem of providing detailed transition rules that are often difficult to define. The users are required only to provide training data for the simulation. The proposed model can significantly reduce the requirements for explicit knowledge in identifying relevant criteria, assigning scores, and determining criteria preference. Variables used in spatial decisions are very often dependent on each other. General MCE methods are not suitable for handling highly correlated variables. Neural networks can learn and generalize correctly and handle redundant, inaccurate, or noisy data which are frequently found in land-use information. Users do not need to worry about which variables should be selected. Knowledge and experiences can be easily learnt by the model and stored for further simulation.

This model can be further extended to simulate competing land-use conversion between multiple land uses. A network with multiple output neurons can be devised to represent various conversion probabilities. Land-use conversion involving multiple land uses is much more complex and a much larger set of spatial variables and parameters would be required for the simulation. It is very difficult to determine parameter values and define model structures for conventional CA methods. Further study is needed to examine the issues related to the simulation of multiple land uses by means of neural networks.

The simulation of future development based on past growth may have some problems because of uncertainty. Like many other computer models, it is difficult to determine exogenous factors or interventions and quantify them for the simulation. However, the proposed model is not limited to the simulation which best fits the historical trend. It is able to simulate alternative development by training the network properly. For example, the training data set can be evaluated to identify the 'good' or 'bad' performance of developed cells according to some criteria. New training data sets can be formed to train the network to accommodate interventions in the change process. Further studies can be carried out to simulate planned development instead of 'actual' development.

Neural networks have some other inherent shortcomings. The optimal structure for the numbers of layers and neurons is still unclear for a specific application. The principle is to use a structure that is as simple as possible. The effects of a structure need to be validated by experiments. More studies may be needed to study the influences of the number of layers and neurons on the simulation results. Furthermore, one should be cautious about the 'overfitting' problem—the network fits well for the training samples but performs poorly for unseen data. The way to avoid the problem is to use the fewest hidden neurons (Zhou and Civco, 1996) and limited iterations for learning (Openshaw and Openshaw, 1997). This can guarantee that most of the generalization features can be conserved.

Acknowledgements. This study is supported by the funding from the Croucher Foundation, Hong Kong. The authors would like to thank the three anonymous referees for their useful comments on revising this paper.

References

Batty M, 1997, "Cellular automata and urban form: a primer" *Journal of the American Planning Association* **63** 266 – 274

Batty M, Xie Y, 1994, "From cells to cities" *Environment and Planning B: Planning and Design* **21** S31 – S48

Batty M, Xie Y, 1997, "Possible urban automata" *Environment and Planning B: Planning and Design* **24** 175 – 192

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

Clarke K C, Gaydos L J, 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** 699 – 714

Clarke K C, Hoppen S, Gaydos L, 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

Congalton R G, 1991, "A review of assessing the accuracy of classification of remotely sensed data" *Remote Sensing of Environment* **37** 35 – 46

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

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

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

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

Hecht-Nielsen R, 1987, "Kolmogorov's mapping neural network existence theorem", in *Proceedings of IEEE First International Conference on Neural Networks, June 1987, San Diego, CA* (IEEE, New York) pp III-11 – III-14

Hepner G, 1990, "Artificial neural network classification using a minimal training set: comparison to conventional supervised classification" *Photogrammatic Engineering and Remote Sensing* **56** 469 – 473

Li X, Yeh A G O, 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** 1501 – 1518

Li X, Yeh A G O, 2000, "Modelling sustainable urban development by the integration of constrained cellular automata and GIS" *International Journal of Geographical Information Science* **14** 131 – 152

Lombardo S T, Rabino G A, 1986, "Calibration procedures and problems of stability in nonlinear dynamic spatial interaction modelling" *Environment and Planning A* **18** 341 – 350

Openshaw S, 1998, "Neural network, genetic, and fuzzy logic models of spatial interaction" *Environment and Planning A* **30** 1857 – 1872

Openshaw S, Openshaw C, 1997 *Artificial Intelligence in Geography* (John Wiley, Chichester, Sussex)

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

Tobler W R, 1979, "Cellular geography", in *Philosophy in Geography* Eds S Gale, G Olssen (D Reidel, Dordrecht) pp 379 – 386

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

White R, Engelen G, 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, Engelen G, Uljee I, 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, 1998, "An experiment on the generic polycentricity of urban growth in a cellular automatic city" *Environment and Planning B: Planning and Design* **25** 731 – 752

Wu F, 2000, "A parameterised urban cellular model combining spontaneous and self-organising growth", in *GIS and Geocomputation* Eds P Atkinson, D Martin (Taylor and Francis, New York) pp 73 – 85

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

Yeh A G O, Li X, 2001a, "A constrained CA model for the simulation and planning of sustainable urban forms by using GIS" *Environment and Planning B: Planning and Design* **28**(5) forthcoming

Yeh A G O, Li X, 2001b, "The measurement and monitoring of urban sprawl in a rapid growing region using the entropy method" *Photogrammetric Engineering and Remote Sensing* **67** 83 – 90

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