Urban Simulation Using Neural Networks and Cellular Automata for Land Use Planning

Anthony Gar-On Yeh[1] and Xia Li[2]

Centre of Urban Planning and Environmental Management, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, P.R. China (hdxugoy@hkucc.hku.hk, lixia@graduate.hku.hk)[1]
Guangzhou Institute of Geography, Guangzhou 510070, P.R. China, xlib@gis.sti.gd.cn[2]

Abstract

The paper presents a method for integrating neural networks, GIS and Cellular Automata (CA) that can be used in land use planning for simulating alternative development patterns according to different planning objectives. Neural networks are used to simplify model structures and facilitate the determination of parameter values. Unlike traditional CA models, the proposed model does not require users to provide transition rules, which may vary for different applications. Historical remote sensing data are used as the training data to calibrate the neural network. The training is robust because it is based on the well-defined back-propagation algorithm. Moreover, original training data are assessed and modified according to planning objectives to generate alternative development patterns.
Keywords: neural networks, cellular automata, GIS, urban simulation, urban planning

1 Introduction

Urban Cellular Automata (CA) have been developing rapidly for the simulation of complex urban systems since the late 80s. A number of interesting investigations have been documented (Batty and Xie, 1994; White and Engelen, 1993; Clarke and Gaydos, 1998). Urban systems involve spatial and sectoral interactions, which cannot be easily adapted to the functionality of current GIS software (Batty et al., 1999). CA-based approaches have applications in the study of urban and regional spatial structure and evolution. Modelling cities with Cellular Automata is a relatively new approach although it has its roots in geography and relates to the work of Hägerstrand (1965) and Tobler (1979) (Clarke and Gaydos, 1998).

Three main types of urban CA models can be considered. The first type uses CA models to generate results that can be explained by urban theories. These models are used to test ideas and assumptions for hypothetical cities. For example, Webster and Wu (1999) present an interesting CA model to implement urban theories concerning developers' profit seeking and communities' welfare-seeking behaviours. In so doing, they explore the mediating effects of alternative systems of land-use rights. The second type is to apply CA models for the simulation of real cities. Clarke and Gaydos (1998) apply CA models 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) provides a realistic simulation of the land-use pattern of Cincinnati, Ohio. A constrained CA model is developed to simulate urban expansion and agricultural land use loss of Dongguan in the Pearl River Delta, a rapidly growing area in southern China (Li and Yeh, 2000). The third type is to use CA to develop normative planning models to simulate different urban forms based on planning objectives. Yeh and Li (2001) use CA to generate different urban forms, ranging from monocentric to polycentric urban development. These urban forms can be assessed to meet selected criteria for sustainable development through minimising agricultural land use in an effort to achieve compact development. Ward et al. (2000) also develop a constrained CA model which has been applied to an area in the Gold Coast, a rapidly urbanising region of eastern Australia. They demonstrate that CA models can simulate planned development as well as realistic development by incorporating sustainable criteria in the simulation.

In the simulation of real cities, model calibration is needed to find suitable values of simulation parameters that can best fit actual development. Unfortunately, there is no universally applicable method of calibration due to the complexity of nature. Another reason is that appropriate methods have not been well developed. There are only very limited studies in addressing the calibration issues in CA simulation. Wu and Webster (1998) use multi-criteria evaluation (MCE) to heuristically define the values of parameters for CA simulation. 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 while holding other parameters constant. Clarke and Gaydos (1998) suggest that calibration can be done by statistically testing the observed criteria against the expected. These methods are very time-consuming because they need to compare all possible combinations of parameters. Another problem is that the combinations are infinite and a sound search procedure is difficult to design.

This paper presents a CA model using neural networks to simulate potential or alternative urban development patterns based on different planning objectives. The simulation of the conversion of non-urban land use to urban land use should be very useful to urban planning. The structure of neural networks is simple and the calibration is easy. Neural network can be used to replace the transition rules used by conventional CA models in a simple and effective way. The following sections will discuss how to develop neural networks to simulate different urban forms based on different planning objectives.

2 Neural Networks for Urban CA Simulation

Neural networks have the capacity to recognise and classify patterns through training or learning processes. They have been used in urban studies, such as journal-to-work flows and airline and telecommunication traffic (Fischer and Gopal, 1994; Openshaw, 1993). These studies indicate that neural networks provide superior levels of performance to those of conventional statistical models because they can well handle the uncertainties of spatial data.

The proposed model consists of two separate parts – training and simulation (Fig. 1). Training is based on the back-propagation procedure, which can generate optimal weights from a set of training data. In this study, remote sensing and GIS are used to provide the historical empirical data to reveal the relationships between site attributes and urban development. There are two ways in using the empirical data for training. The original data can be directly used to generate realistic simulation, which assumes that urban growth proceeds according to historical trends. However, the original data can also be modified according to some criteria, which are related to planning objectives.

In creating new sets of training data, some criteria should be provided to evaluate the past development 'points' in the original data set. It is assumed that some better and worse development 'points' can be identified according to their costs and benefits, which are measured by the criteria. A simple way is to remove some 'bad points' based on the evaluation. The modified data set can allow the training to ignore 'false' information. This is important for the model to generate ideal or optimal simulation results. A number of modified data sets can be obtained by using different sets of criteria, which correspond to planning objectives.

The new sets 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 generative alternative or ideal urban patterns. This can help to correct the historical land use problems and search for more appropriate urban growth in the future. The whole procedure is simple and robust because the parameter values are not arbitrarily defined.

Site attributes determine development probability. It is very convenient to obtain site attributes when GIS analysis functions are employed. The site attributes for CA simulation usually includes the development level in the neighbourhood, as well as various types of proximity attractiveness (Batty and Xie, 1994; Wu and Webster, 1998). Common GIS analysis functions, such as buffer and overlay, can be carried out to obtain these spatial variables.

The CA simulation is based on the algorithm of neural networks. At each iteration, the neural networks will determine the development probability, which is subject to the input of site attributes and weights. A cell may have n site attributes (variables):

$$(S_1, S_2, S_3, S_4, S_5, S_6, \ldots, S_n) \quad (1)$$

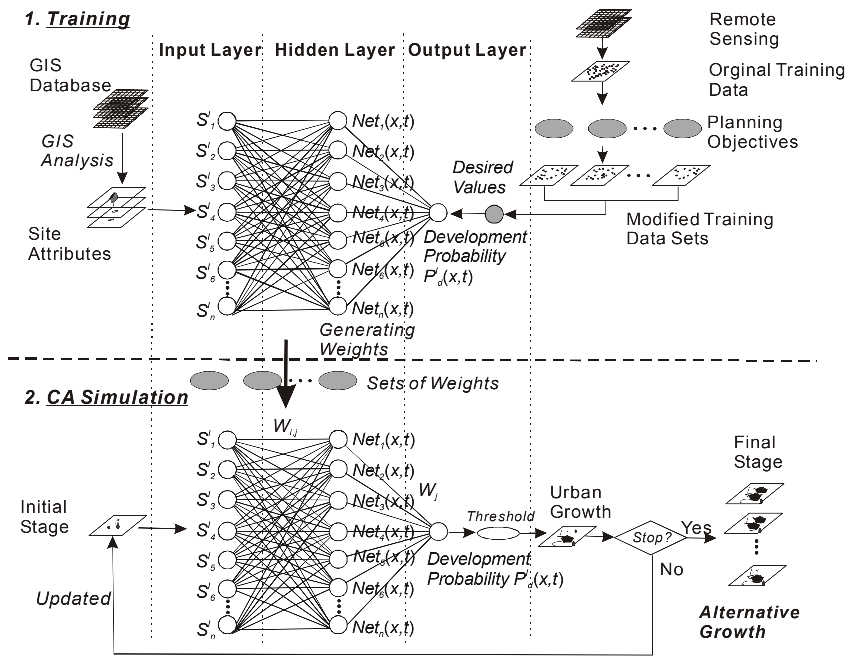

Fig. 1. Simulation of urban forms using the neural-network-based CA model

A neural network can be designed to estimate development probability at each iteration of the CA simulation. The neural network has three layers, one input layer, one hidden layer and one output layer. The input layer has n neurons with regard to these site attributes. The hidden layer may also have n neurons. The output layer has only one neuron, which calculates the development probability. 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 at each iteration.

The original data are usually scaled into the range of [0, 1] before they are input to neural networks (Gong, 1996). Scaling each variable treats 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:

$$S_i' = (S_i - \text{minimum}) / (\text{maximum} - \text{minimum}) \tag{2}$$

The algorithm for the CA model is based on a simple three-layer network. In the neural network, the signal received by neuron j of the hidden layer from the first input layer for cell x is calculated by:

$$net_j(x,t) = \sum_i W_{i,j} S_i'(x,t) \tag{3}$$

where x is a cell, $net_j(x,t)$ is the signal received for neuron j of cell x at time t, $W_{i,j}$ is the weight of the input from neuron i to neuron j and $S'_i(x,t)$ is the site attributes for variable (neuron) i.

The activation of the hidden layer for the signal is:

$$\frac{1}{1+e^{-net_j(x,t)}} \tag{4}$$

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

$$P_d(x,t) = \sum_j W_j \frac{1}{1+e^{-net_j(x,t)}} \tag{5}$$

This simulation is loop-based. Development probability is calculated according to site attributes using the neural network at each iteration. The development probability decides whether a cell is converted or not for development. A stochastic disturbance term can be added to represent unknown errors during the simulation. This can allow the generated patterns to be more close to reality. The error term (RA) is given by (White and Engelen, 1993):

$$RA = 1+(-\ln \gamma)^{\alpha} \tag{6}$$

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

$$\begin{aligned} P_d'(x,t) &= RA \times \sum_j W_j \frac{1}{1+e^{-net_j(x,t)}} \\ &= (\mathit{1} + (\mathit{-ln}\ \gamma)^{\alpha}) \times \sum_j W_j \frac{1}{1+e^{-net_j(x,t)}} \end{aligned} \tag{7}$$

A cell with a high value of development probability will be likely urbanised during the simulation. A predefined threshold value should be used to decide whether a cell is developed or not at each iteration. If a cell has a probability greater than the threshold value, it will be converted for development. The number of cells in the neighbourhood is recalculated and the site attributes are updated at the end of each iteration. The simulation will continue until the total number of converted cells is equal to the required land consumption.

3 Applications

3.1 Study Area and Site Attributes

A real city, Dongguan, was selected for the experiment to demonstrate the advantages of the neural-network-based CA model. The model was used to simulate the conversion from non-urban land use to urban land use. Located in the Pearl River Delta region, China, it has a total land area of 2,465 km^2. It is a fast growing region with a tremendous amount of land use changes and urban sprawl in recent years (Yeh and Li, 1997; Yeh and Li, 1999). In this study, seven spatial variables are defined to represent the site attributes of each cell for the simulation of urban development. These variables include:

1. Distance to the major (city proper) urban areas S_1;
2. Distances to sub-urban (town) areas S_2;
3. Distance to the nearest road S_3;
4. Distance to the nearest expressway S_4;
5. Distance to the nearest railways S_5;
6. Neighbourhood development level (the window of 7 × 7 cells) S_6;
7. Agricultural suitability S_7.

The reason to choose these variables is that they are important factors to decide development probability (Wu and Webster, 1998). These site attributes were obtained 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 neighbourhood development level was measured based on the number of developed cells in the 7 × 7 cells adjacent to the central cell. It was calculated using the *Focal* function of *ARC/INFO GRID*. This variable was also dynamically updated during the simulation. The initial neighbourhood development level was calculated from the 1988 binary image.

3.2 Training

Training data for urban growth were obtained by the classification of satellite TM images of 1988 and 1993 (Fig. 2). 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 reduced to 50 × 50 m by re-sampling for faster simulation. Binary values were used to represent developed and non-developed areas in 1988 and 1993. The value of 1 represents developed (converted) cells and 0 represents non-developed ones. The urban areas of 1988 were used as the starting point of the simulation. The urban areas of 1993 were also obtained for testing the simulation by comparing the actual development with the result of the simulated development. The CA simulations were carried out from 1988 to 1993 so that different urban forms generated by the model can be compared and evaluated against the actual development in 1993.

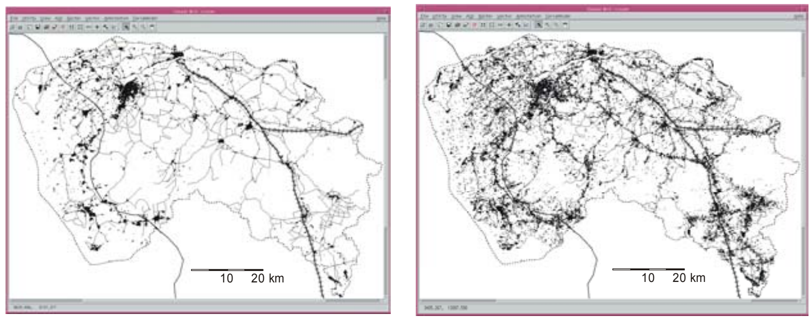

Fig. 2. Urban areas of Dongguan in 1988 and 1993 classified from satellite TM images

Training was carried out by using a neural network package referred to as THINKS PRO[1] . An essential task is to design the network structure for the CA simulation. The design of the network structure is relaxed since the numbers of layers and neurons in each layer can be subjectively determined. There are, however, some principles that can be used to guide the determination of the network structure. The increase of the numbers of layers and neurons will drastically increase the computation time for the loop-based CA model. It is practical to use the numbers of layers and neurons as few as possible without severely compromising the model accuracy. Kolmogorov's theorem suggests that any continuous function $\phi: X^n \rightarrow R^c$ 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). 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 CA models, which are based on 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).

Therefore, it is appropriate to use three layers of the neural network for urban simulation. The input layer has seven neurons corresponding to 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 7 weights between the hidden layer and the output layer. A total of 56 parameters were used for the neural-network-based CA model.

Original training data were obtained by the overlay of historical urban growth (1988-93) from remote sensing and site attributes from GIS. The training data can be used to calibrate the network to produce the realistic simulation of the study area. 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 the data volume and data redundancy.

The random stratified sampling points were generated by the *ERDAS IMAGINE*[1] package. Their co-ordinates 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. 1,000 random sampling points were obtained and used to train the neural network.

3.3 Simulation Results

The simulation was totally based on the neural network. 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 in a GIS platform by the integration of neural network and GIS. GIS facilitates access to spatial data, which are used as site attributes for the simulation. The model was developed in *ARC/INFO*[2] *GRID* using the Arc Macro Language (AML). The GIS package also provides powerful spatial handling functions that are useful for CA simulation.

The output neuron of the network generated the development probability for each cell at each iteration. The cells with development probability greater than the threshold of 0.85 were converted into developed cells. The neighbourhood development level was then recalculated again to update the site attributes. The parameter α of the random disturbance was set to 1 so that only a small amount of uncertainty was presented in the simulation. The simulation time was automatically determined to ensure that the amount of land conversion was finally equal to that of actual development in 1993.

The model can be conveniently used to predict urban development based on past trends. However, an import task of our CA simulation is to generate alternative development patterns based on different pre-defined planning objectives. A way to generate alternative development patterns is to integrate MCE with CA models (Wu and Webster, 1998). Parameter values can be adjusted corresponding to various planning objectives. However, there are uncertainties about the method because the determination of parameter values is quite relaxed.

---

[1] *ERDAS IMAGINE* is a trademark of ERDAS, Inc.

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

A way to remove the uncertainties is to obtain the parameter values based on training. A number of different training data sets can be defined based on quantitative evaluation. The sample points from the original data sets were evaluated according to some criteria related to a planning objective. Unsuitable development cells were identified according to these criteria. For example, the sites with the distance greater than 30 km from town centres are considered as unsuitable for development of the objective of town-centre-based development. If a development point in the original sampling data set has a distance greater than the threshold, its desired value (development or not) will be adjusted from 1 to 0. For the objective of agricultural-conservation, all development points with the suitability score greater than 0.8 will be considered as unsuitable for development. Their desired values will be changed to 0 accordingly so that the neural network can be trained to avoid the encroachment on good agricultural land in the simulation of urban growth.

The modification of the training data set can help the neural network to remember the 'failure' and prevent the problem in the simulation. It is easy to create a couple of alternative training data sets from the original data set based on the evaluation of past development. The modification is related to planning objectives. We only use four typical planning objectives for urban growth because other options can be easily defined by the same method (Table 1).

Table 1. Rules for creating alternative training data sets

Planning Objectives	Modification Rules
1. Continuation of past development trend	Use original data without any modification
2. Promotion of mono-centric development	Change the desired values to 0 for all the development sites with $S_1 > 200$
3. Promotion of poly-centric development	Change the desired value to 0 for all the development sites with $S_2 > 30$
4. Promotion of agricultural-conservation development	Change the desired value to 0 for all the development sites with $S_7 > 0.8$

After these training data sets had been created based on the above modification rules, the *THINK PRO* package was used to obtain the corresponding sets of parameters or weights by the training procedure. Four different sets of weights were obtained for the simulation of different urban forms. Normal weights were obtained using the original data (planning objective 1), but adjusted weights were obtained using the modified data (planning objectives 2, 3 and 4). Different sets of weights were then input to the CA model to generate different development patterns. The model provides a useful tool making it possible to directly link the training network to the formation of distinct urban development patterns.

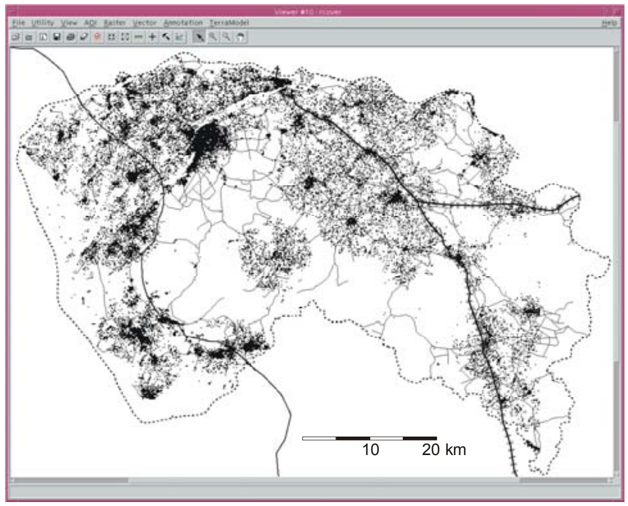

Fig. 3. Simulation of development in 1988-1993 using normal weights (planning objective 1 - continuation of past development trend)

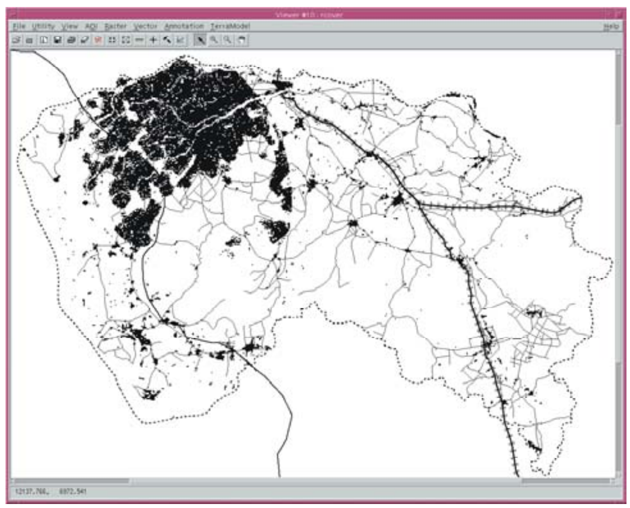

Fig. 4. Simulation of mono-centric development (planning objective 2) using adjusted weights

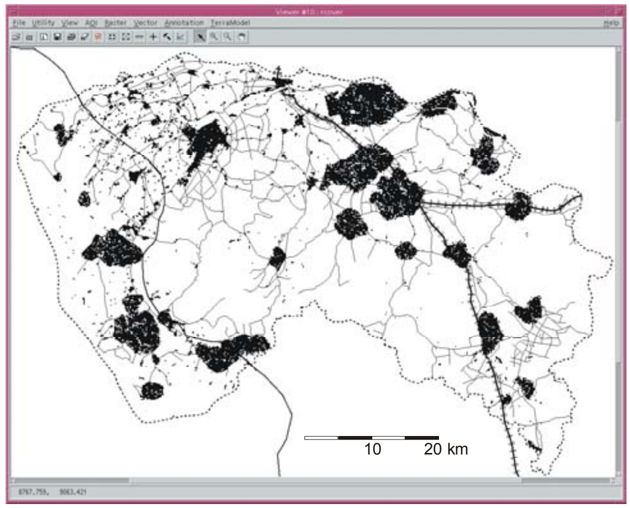

Fig. 5. Simulation of poly-centric development (planning objective 3) using adjusted weights

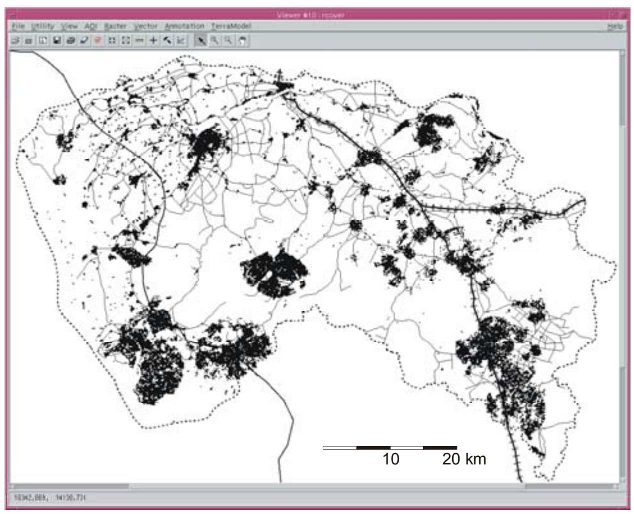

Fig. 6. Simulation of urban development with strict protection of agricultural land (planning objective 4) using adjusted weights

Fig. 3 shows the simulation output by using the normal weights obtained from the original training data set which is based on the past growth trend without any modification. As a result, the simulation is quite similar to the actual development (Fig. 2). Alternative urban forms can be derived by proper training of the network and through using different planning objectives as shown in Table 1. Fig. 4 shows the effects of promoting mono-centric development around the existing city proper by using the adjusted weights obtained from planning objective 2. Fig. 5 shows the simulation of promoting poly-centric development around the existing 29 town centres. In this case the adjusted weights obtained from planning objective 3 are used. Finally, the strict control on the conversion of the best agricultural land can be implemented by applying the fourth set of adjusted weights according to planning objective 4 (Fig. 6). It can be found that land development is well controlled in the alluvial plain in the northwestern part of the region.

4 Conclusion

It is very tedious to define model structure, transition rules and parameter values for conventional CA models. The proposed CA model can simplify these jobs considerably. Although multi-criteria evaluation (MCE) techniques can be used to define various parameter values, the method substantially relies on expert knowledge and experiences. There are also uncertainties because parameter values are usually defined in a relaxed way. The neural network-based model can significantly reduce the requirements for explicit knowledge for identifying relevant criteria, for assigning scores, and for determining criteria preferences. The model can effectively map the non-linear features of urban systems because it uses neural networks.

In this model, the original training data set can be evaluated to identify the 'good' or 'bad' performance of developed cells according to some criteria, which are related to planning objectives. New training data sets can be formed to train the network to accommodate interventions in the simulation process. The model can be used in two situations: simulation of urban development based on the existing development trend, and generation of alternative development patterns based on different planning objectives. Distinctive development patterns can be easily simulated based on the training of neural networks using different adjusted weights to reflect different planning objectives. Remote sensing and GIS data can be used to prepare the training data sets for a simulation that is more realistic. Based on planning objectives and development evaluation, original training data sets can be modified to obtain different sets of adjusted weights through the training procedure of neural networks. These adjusted weights can be applied to the CA model in generating preferable patterns.

The neural-network-based model is simple and convenient to use, but it can generate very complex features of urban systems. However, an inherent problem with neural network models 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 the network structure is also subject to user's preferences. It seems that there is no way to determine what is the best network structure.

The simulation also assumes that transport systems (railways and roads) would not change during the simulation period. It is assumed they were just upgraded and should therefore remain stable for the short period of time during which the simulation takes place. Future studies should accommodate the possible changes in the transport system for testing in the model. There are substantial uncertainties in simulating future changes of road systems since they are subject to relatively frequent changes and are often affected by urban transportation policies. It is more reasonable to tackle this issue by considering future transport development as an exogenous factor, which can be addressed by GIS for subsequent input to CA models.

Acknowledgements

This research is supported by the funding from the Croucher Foundation, Hong Kong.

References

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

Batty M, Xie Y, and Sun ZL (1999) Modeling urban dynamics through GIS-based cellular automata. Computers, Environment and Urban Systems 23: 205-233

Clarke KC, Gaydos LJ (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

Clarke KC, Gaydos L, Hoppen S (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

Fischer MM, Gopal S (1994) Artificial neural networks: a new approach to modelling interregional telecommunication flows. Journal of Regional Science 34: 503-527

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 VI: 43-67

Li X, Yeh AGO (2000) Modelling sustainable urban development by the integration of constrained cellular automata and GIS. International Journal of Geographical Information Science 14(2): 131-152

Openshaw S (1993) Modelling spatial interaction using a neural net. In: Fischer MM, Nijkamp P (eds) GIS, Spatial modelling and policy. Springer, Berlin, pp 147-164

Tobler WR (1979) Cellular geography. In: Gale S, Olssen G (eds) Philosophy in Geography. Reidel, Dordrecht, pp 379-386

Ward DP, Murray AT, Phinn SR (2000) A stochastically constrained cellular model of urban growth. Computers, Environment and Urban Systems 24: 539-558

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

Webster CJ, Wu F (1999) Regulation, land-use mix, and urban performance. Part 2: simulation. Environment and Planning A 31: 1529-1545

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, Uijee 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, Webster CJ (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 AGO, Li X (1997) An integrated remote sensing and GIS approach in the monitoring and evaluation of rapid urban growth for sustainable development in the Pearl Rive Delta, China. International Planning Studies 2: 195-222.

Yeh AGO, Li X (1999) Economic development and agricultural land loss in the Pearl River Delta, China. Habitat International 23: 373-390

Yeh AGO, Li X (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