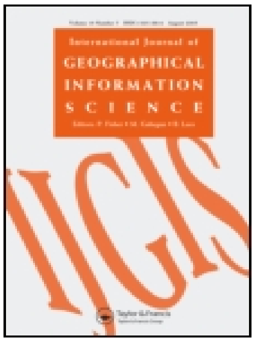

International Journal of Geographical Information Science

ISSN: 1365-8816 (Print) 1362-3087 (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

An intelligent method to discover transition rules for cellular automata using bee colony optimisation

Jianyi Yang, Guo'an Tang, Min Cao & Rui Zhu

International Journal of Geographical Information Science, 2013
Vol. 27, No. 10, 1849–1864, http://dx.doi.org/10.1080/13658816.2013.823498

An intelligent method to discover transition rules for cellular automata using bee colony optimisation

Jianyi Yang[a], Guo'an Tang[a]*, Min Cao[a] and Rui Zhu[b]

[a]*Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, Nanjing, Jiangsu, China;* [b]*Department of Urban Planning and Environment, Royal Institute of Technology – KTH, Sweden*

(*Received 8 January 2013; accepted 5 July 2013*)

This paper presents a new, intelligent approach to discover transition rules for geographical cellular automata (CA) based on bee colony optimisation (BCO–CA) that can perform complex tasks through the cooperation and interaction of bees. The artificial bee colony miner algorithm is used to discover transition rules. In BCO–CA, a food source position is defined by its upper and lower thresholds for each attribute, and each bee searches the best upper and lower thresholds in each attribute as a zone. A transition rule is organised when the zone in each attribute is connected to another node by the operator 'And' and is linked to a cell status value. The transition rules are expressed by the logical structure statement 'IF-Then', which is explicit and easy to understand. Bee colony optimisation could better avoid the tendency to be vulnerable to local optimisation through local and global searching in the iterative process, and it does not require the discretisation of attribute values. Finally, The BCO–CA model is employed to simulate urban development in the Xi'an-Xian Yang urban area in China. Preliminary results suggest that this BCO approach is effective in capturing complex relationships between spatial variables and urban dynamics. Experimental results indicate that the BCO–CA model achieves a higher accuracy than the NULL and ACO–CA models, which demonstrates the feasibility and availability of the model in the simulation of complex urban dynamic change.

Keywords: bee colony optimisation; transition rules; CA; urban simulation

1. Introduction

Cellular automata (CA), which were presented by Ulam and Neumann in the 1940s, are capable of simulating the evolution of complex geographical phenomena (Liu *et al*. 2008). Currently, CA have successfully been applied to urban expansion and population dynamics (Couclelis 1985, 1988, 1989), wildfire diffusion (Clarke *et al*. 1994, Hargrove 2000), landscape changes (Wang and Zhang 2001, Soiares-Filho *et al*. 2002), urban evolution (Batty and Xie 1994, Clarke *et al*. 1997, Couclelis 1997, Li and Yeh 2000, 2002, Wu 2002, Liu *et al*. 2008) and land-use changes (White and Engelen 1993, Batty and Xie 1994, Clarke and Gaydos 1998, Wu and Webster 1998, Li and Yeh 2002, 2004). These studies have demonstrated the capability of CA for simulating and predicting complex geographical processes.

*Corresponding author. Email: tangguoan@njnu.edu.cn

The core of CA is transition rules, which express a logistic relationship for an evolutionary process and determine the state conversion of geographical processes (Liu *et al*. 2008). Many methods have been put forward for defining transition rules, such as the hierarchical analysis process (AHP) (Wu 1998), genetic algorithm (GA) (Jenerette and Wu 2001), logistic regression models (Wu 2002), artificial neural network (ANN) (Li and Yeh 2002), decision tree models (Li and Yeh 2004) and the kernel-based learning machine (Liu and Li 2006). However, these methods are each limited for different reasons. For example, the decision tree model tends to be vulnerable to local optimisation, and the kernel-based learning machine is constrained by the use of implicit transition rules and requires intensive computation (Liu *et al*. 2008). To avoid these limitations, intelligent methods for discovering transition rules were conceived. Ant colony optimisation (ACO) was put forward for discovering transition rules of CA (Liu *et al*.2008), and a CA model that was based on the ACO and Markov chain was proposed to simulate land-use change (Yang *et al*. 2012). These studies have demonstrated that intelligent methods are better for discovering transition rules for CA. However, it is necessary to discretise attributes in the CA model based on ACO (ACO–CA), which are influenced by discretisation algorithms. Thus, exploration for a new, intelligent method for discovering transition rules in CA is needed.

In this paper, a new intelligent method, bee colony optimisation (BCO), is used to discover transition rules for CA. BCO is a computational method that is derived from natural biological systems. The self-organisation model of a bee colony was first proposed by Seeley (1995), but Karaboga (2005) proposed a more favourable artificial bee colony (ABC) algorithm and proved that it outperformed others in numerical optimisation. BCO is a multi-agent system that simulates the natural behaviour of a bee colony searching for food sources according to the mechanisms of cooperation and adaptation. BCO is a new swarm intelligence-based heuristic system, which has been applied to many fields, such as numerical function optimisation (Karaboga and Basturk 2007a, 2007b, 2008, Chen *et al*. 2012), constrained numerical optimisation (Karaboga and Basturk 2007a, 2007b), training ANNs (Karaboga and Akay 2007, Karaboga *et al*. 2007), digital IIR (Infinite Impulse Response) filters (Karaboga 2009) and digital image processing (Duan *et al*. 2010). However, the application of BCO in discovering the transition rule of the CA model has not been reported. BCO and CA both solve complicated problems based on a ‘bottom-up’ approach; therefore, it is appropriate to apply BCO to discover transition rules for CA.

In the context of data mining, the use of BCO for discovering classification rules is a new research area with limited studies. The method of rule discovery that is based on the artificial bee colony rule miner algorithm (ABC-Miner) programme was first proposed by Shukran et al. (2011). In that study, the rules that were discovered by the ABC-Miner were more accurate and simple than those by particle swarm optimisation.

This paper will explore the feasibility of using ABC-Miner to discover transition rules for CA and of finding complex relationships that are hidden in large datasets, using GIS and remote-sensing data. It is expected that the BCO will produce better results in discovering optimal rules by simulating the behaviour of a bee colony seeking food. The CA based on BCO (BCO–CA) model is applied to a simulation of urban development in the Xi’an-Xian Yang urban area in China.

2. Geographical CA model based on BCO

2.1. Bee colony optimisation

BCO determines the optimised solution by simulating bees’ behaviours for seeking the best food source. In the process of seeking food, it is most important to form collective knowledge through the exchange of information between bees. Communication information about the quality and position of food sources among bees is through a dance termed the 'waggle dance', and the information from this dance is available to an onlooker on the dance floor. A greater probability exists for onlookers to choose more profitable sources because more information about the more profitable sources is circulated (Karaboga *et al*. 2012). Hence, recruitment is proportional to the profitability of the food source.

The BCO model includes three essential components: food sources, employed foragers and unemployed foragers. If a food source exists, the behaviours of bees seeking the best food source are shown in Figure 1. Initially, all foragers around the nest are unemployed and have no information regarding food sources. These individuals have two options: one is to scout and start searching for food around their hive (route 'S'), and the other is to be recruited after finding the waggle dances and then begin searching for food (route 'R'). When onlookers find food sources, they become employed foragers, record the information about the position and nectar amount of the food source and then return to the hive. After unloading the food, the foragers will calculate the nectar amount, and if the nectar amount of the new food source is not higher than that of the previous source, they will continue to search for a new food source (UF). Otherwise, the bees will recruit other bees through the 'waggle dance' (EF1) or continue to gather honey (EF2). When compared to other colony optimisations, the essential advantage of BCO is that foragers will abandon food sources when the number of searching times reaches a threshold. With this mechanism, the BCO could better avoid the tendency to be vulnerable to local optimisation.

2.2. *Discovering rules based on BCO*

2.2.1. *Rule construction*

BCO is a complex, self-adaptive system that has four characteristics to which a self-organisation system relies: positive feedback, negative feedback, fluctuations and multiple interactions (Karaboga *et al*. 2012). In the context of data mining, the use of BCO for discovering classification rules is a new research area with limited studies. The rule discovery method that is based on BCO, i.e. the ABC-Miner, was first proposed by Shukran et al. (2011). In this paper, the ABC-Miner program is used to discover transition rules for CA from geographic data.

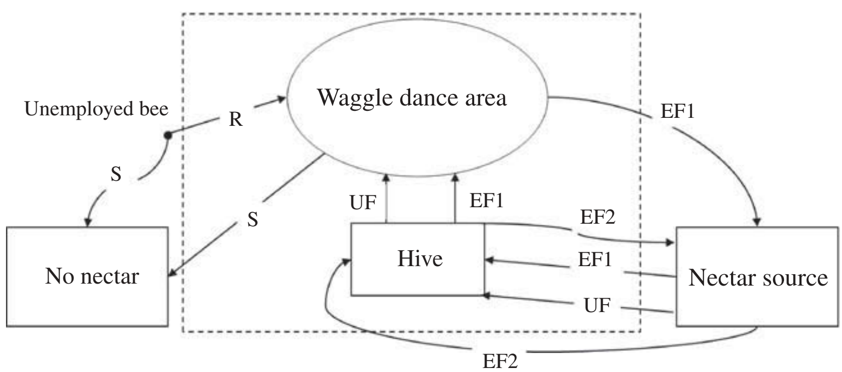

Figure 1. Behaviours of bees seeking the best food source.

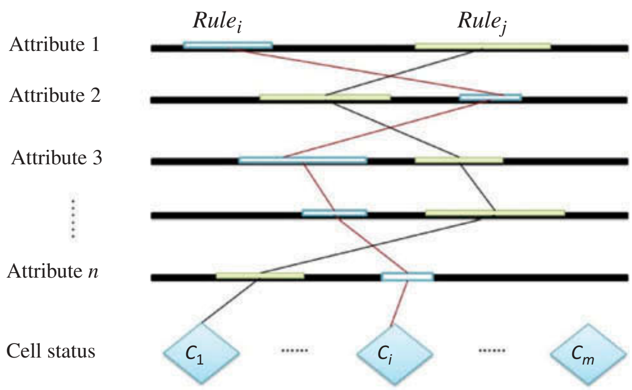

Figure 2. The principle of structuring CA rule based on ABC-Miner.

This ABC-Miner program can discover optimised transition rules by simulating the behaviour of bees. A transition rule is constructed by a cell status value and zones, each of which is defined by a lower and an upper threshold value for each attribute. As is shown in Figure 2, the cell status and attribute nodes with the same colour are linked to the structure of a rule. Artificial bees can find the best lower and upper threshold values of the zones in each attribute, and data mining for a transition rule can be regarded as a process of searching for optimal food sources. The transition rule of CA could be represented by the following reasoning:

If $lower\_threshold_1$ < Attribute 1 < $upper\_threshold_1$
And $lower\_threshold_2$ < Attribute 2 < $upper\_threshold_2$
. . . . . .
And $lower\_threshold_n$ < Attribute n < $upper\_threshold_n$
then
The cell status is C_i

where n is the number of attributes; the $lower\_threshold_k$ is the best lower threshold value; the $upper\_threshold_k$ is the best upper threshold value of the zone in the kth attribute, $k \in \{1, 2, \ldots\ldots, n\}$; C_i refers to a cell status value ($i \in \{1, 2, \ldots\ldots, m\}$); and m is the amount of cell status.

Each bee randomly begins with a food source position, and the update for a food source position relies on the nectar amount, which will be discussed in detail in Section 2.2.3.When a rule is constructed, it is important to prune the rule and update the training dataset, which will be discussed in detail in Section 2.2.4.

2.2.2. *Rule evaluation*

In the BCO, the validity of a transition rule is defined as the nectar amount and determines the optimisation direction. The Gini index is applied to the calculation for validity (Shukran *et al*. 2011) and is defined by the following expression:

$$f = \left(\frac{TP}{TP + FN}\right) \times \left(\frac{FP}{FP + TN}\right) \tag{1}$$

where *TP* (true positives) is the total number of cases that are covered by the rule that have predicted the class by the rule; *FP* (false positives) is the total number of cases that are covered by the rule that have a class that is different from that which is predicted by the rule; *FN* (false negatives) is the total number of cases that are not covered by the rule but that have the class that is predicted by the rule; *TN* (true negatives) is the total number of cases that are not covered by the rule and do not have the class that is predicted by the rule. A higher value of f indicates higher validity.

2.2.3. Process of the discovering rule

The BCO constructs transition rules by stimulating the foraging behaviours of a bee colony. Each bee searches the upper (X_i^{j-}) and lower thresholds (X_i^{j+}) for the best value in each attribute. The best value zones [X_i^{j-},X_i^{j+}] in each attribute can be connected with each other by the operator 'And' and then linked with the cell status value, which forms the transition rules.

Suppose *m*-bees and *n*-attributes are present, the food source position of the *i*th bee could be represented as (X_1^{j-}, X_1^{j+}, ... , X_{i}^{j-}, X_{i}^{j+}, ... , X_n^{j-}, X_n^{j+}). Each bee searches the best value in a 2*n*-dimensional space; so, the following steps express the main process for the construction of transition rules:

(1) Initialise food source positions: Each bee randomly generates an initial food source position, and the initial position values can be represented by the following equation:

$$\begin{aligned} X_i^{j-} &= X_{\min}^j + rand(0,1)\left(X_{\max}^j - X_{\min}^j\right) \\ X_i^{j+} &= X_{\min}^j + rand(0,1)\left(X_{\max}^j - X_{\min}^j\right) \end{aligned} \tag{2}$$

where $X_1^{j-}$and X_i^{j+}represent the lower and upper thresholds of the *i*th bee at the *j*th attribute, respectively, and, if $X_i^{j-}>X_i^{j+}$, these values will be exchanged. $X_{\min}^j$ and $X_{\max}^j$ are the minimum and maximum values, respectively, of the *j*th attribute, and *rand*(0,1) is a random value that varies between [0,1].

Subsequently, the nectar amount of the food sources will be calculated. The top $m/2$ food sources will be set as the initial foodsource positions of the employed bees, and the others will be abandoned.

(2) Update food source positions: Based on the nectar amount of the food sources, bees begin to iteratively search for new food sources. The employed and unemployed bees use different methods to search for food sources.

Each employed bee searches for a new food source in the neighbourhood of previous food sources, then the bee calculates the new nectar amount and chooses the better one to be the new food source. The searching method is represented by Formula (2).

$$\begin{aligned} V_i^{j-} &= X_i^{j-} + \theta_i^j\left(X_i^{j-} - X_k^{j-}\right) \\ V_i^{j+} &= X_i^{j+} + \phi_i^j\left(X_i^{j+} - X_k^{j+}\right) \end{aligned} \tag{3}$$

where V_i^{j-} and V_i^{j+} denote the new lower and upper thresholds, respectively, of the ith bee at the jth attribute; and, if $V_i^{j-} > V_i^{j+}$, these values will be exchanged. X_k^{j-} and X_k^{j+} refer to the lower and upper thresholds, respectively, of the kth bee at the jth attribute, where k is a random value that belongs to $\{1, 2, \ldots, m/2\}$ and is not equal to i. θ_i^j and ϕ_i^j are random values that vary in the range of [–1,1], and j is a value that is randomly selected from $\{1, 2, \ldots, n\}$.

After the employed bees complete their search, the unemployed bees probabilistically choose their food sources depending on the information provided by the employed bees and update the food source position based on Formula (2). If the new food source is better than the previous one, the bees save the new food source, and this method of selecting food is represented by the following equation:

$$P(X = X_i) = \frac{f(X_i)}{\sum_{i=1}^{N} f(X_i)} \tag{4}$$

where X_i refers to the food source of the ith employed bee; and $f(X_i)$ refers to the nectar amount of the employed bee X_i.

(3) Local and global search mechanisms: If the searching time exceeds the limitation value for a bee and the bee still cannot find a better food source, then the bee will reinitialise a food source. This mechanism could avoid the tendency of the BCO to be vulnerable to local optimisation as much as possible. This mechanism can be expressed by the following equation:

$$\begin{cases} X_{t+1} = reInit() & sNum \geq Limit \\ X_{t+1} = Gent(X_t) & sNum < Limit \end{cases} \tag{5}$$

where X_{t+1} refers to the position at time $t + 1$; *sNum* refers to the number of searches; the function of *reInit*( ) is to reinitialise a position based on Formula (1); and the function of $Gent(X_t)$ is to generate a new position from X_t based on Formula (3).

When all bees accomplish their current search, the best food source will be selected and compared to the global optimum food source from the previous iteration, and the better one will be recorded as the new global optimum.

When the number of iteration times reaches the predefined threshold, the algorithm will stop searching, and the final, best food source and its fitness will be recorded. At this time, one transition rule has been constructed.

2.2.4. Rule pruning and updating of the training dataset

The constructed rules must be pruned to improve their classification performance. The goal of rule pruning is to remove unnecessary terms and improve the quality of the rules because a shorter rule is generally more comprehensible by users (Liu *et al*. 2008). Another motivation for rule pruning is to improve the predicative accuracy of the rules and prevent the rules from overfitting the training data. In the following discussion, 'term' refers to a zone at an attribute, and the specific pruning process for one rule is as follows:

(1) remove one term from the rule;
(2) recalculate the validity of the rule;
(3) add back the term that was removed from the rule if the validity of the pruned rule cannot be improved. Otherwise, remove the next term and go to step 2; and
(4) cycle step 2 to step 3 until removing any term reduces the validity.

After pruning a rule, instances that meet the rule are removed from the training dataset to obtain the highest-quality rule. When the number of instances that are related to a cell status is less than the predefined threshold, the algorithm stops discovering a rule for the cell status.

2.3. BCO-based, geographical CA model building

The basic principle of BCO is to search for the optimised food sources by communicating information about the food source position and nectar amount. BCO BCO–CA are put forward to simulate the complex geographic processes by some explicit transition rules. This paper introduces the ABC-Miner program to automatically derive transition rules of CA from training data, and the structure of the BCO-based, geographical CA model is shown in Figure 3.

BCO–CA consists of two parts: building transition rules based on the ABC-Miner and simulating land-use change based on the rules. This paper first uses the ABC-Miner algorithm to discover transition rules from land-use data within two periods and constructs a BCO–CA model. The pseudo-code of the ABC-Miner algorithm is shown in Table 1, and the ABC-Miner algorithm relates to the following parameters:

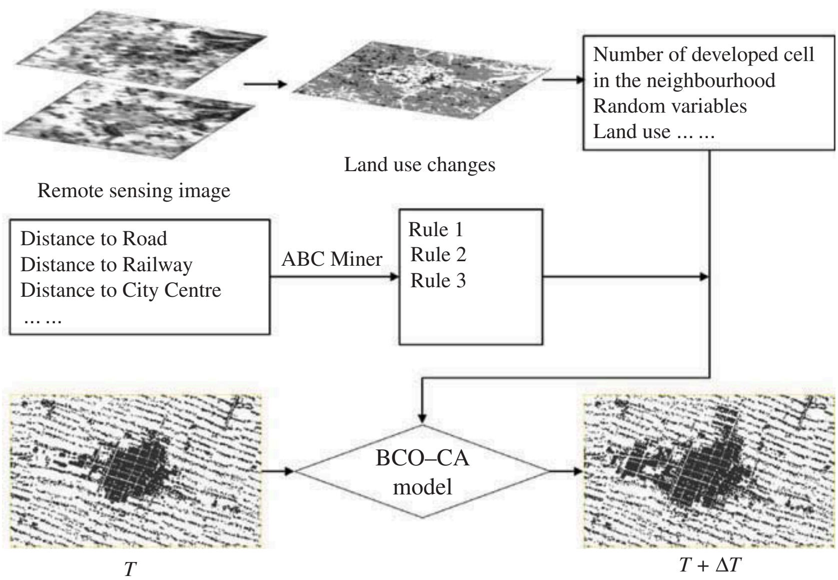

Figure 3. Structure BCO-based geographic CA model.

Table 1. The pseudo-code of the ABC-Miner algorithm.

```
Input training dataset;
Initialise control parameters of ABC-Miner algorithm;
while remain instance number> Min _Num_Ins {
    rules = Initialise(training_dataset, Bee_Num/2);
    while iter < Max_IterTimes {
        for i = 1 to Bee_Num/2 {
            Optimise upper_value of rules(i);
            Optimise lower_value of rules(i);
            if the validity of rules(i) is not improved then
                unimprovetimes(i) ++; }
        NormFitnesses = Fitnesses/sum(Fitnesses);
        counter = 1, t = 0;
        while t < Bee_Num/2 {
            if rand(0,1) < NormFitness(counter) then
                Optimise upper_value of rules(i);
                Optimise lower_value of rules(i);
                if the validity of rules(i) is not improved then
                    unimprovetimes(i) ++ ;
                t = t + 1;
            else
                counter ++;
                if counter > Bee_Num/2 then counter = 1; }
        if unimprovetimes(i) > Max_UnImNum then
            reinitialise rules(i); }
        iter++; }
    best_rule = SelectBestRule(rules) ;
    best_rule = RemoveUnnecessaryConditions(best_rule);
    Removeclassifiedsamples(best_rule, traing_dataset);
    Add the best rule to ruleset;}
ruleset = RemoveUnnecessaryRules(ruleset)
```

(1) *Bee_Num*: the number of bees, which means the number of candidate rules during iteration;
(2) *Min_Num_Ins*: the minimum number of instances. If the number of instances related to a cell status is less than the threshold, the algorithm stops discovering rules for the current cell status;
(3) *Max_IterTimes*: the limitation of iteration-times. The program stops when the number of iterations exceeds this limitation; and
(4) *Max_UnImNum*: the limitation of unimproved-times. A rule will be reinitialised if the number of unimproved-times that are related to the rule exceeds this limitation.

During the simulation, the state conversion of a central cell is determined by these transition rules and by some uncertain factors. Therefore, a random variable is often incorporated to generate realistic patterns for urban simulations (White and Engelen 1993, Wu and Webster 1998, Liu *et al*. 2008). However, the probabilistic elements could not affect the accuracy of the CA model because the uncertain parts cover small proportions and are usually located at the fringes of urbanised clusters (Yeh and Li 2006, Liu *et al*. 2008). In this paper, a random variable (γ) is used to determine the effect of uncertain factors. Subsequently, the status of a cell should be altered according to the following conditions:

(1) Meeting one of the original rules;
(2) $\gamma < \alpha$, where$\alpha$is calculated by Formula (6):

$$\alpha = \frac{1}{K} \tag{6}$$

where K is the number of iterations in the CA model.

Because the survey interval (ΔT) between the remote-sensing data is generally far greater than the iteration interval (Δt) of the CA simulation (Liu *et al*. 2008), it is necessary to determine the number of converting cells in the iteration interval (Δt) to make the observation interval (ΔT) equal or close to the iteration interval (Δt) (Li and Yeh 2004).

The number of iterations (K) of the CA model can be calculated by the following equation:

$$K = \frac{\Delta T}{\Delta t} \tag{7}$$

Then, the number of conversions (ΔQ) is obtained from the land-use data for the survey interval. Because $\Delta T > \Delta t$, only a portion of the cells are converted in an iteration interval (Δt), and the amount of converting cells (ΔQ_0) between t and $t + 1$ is calculated by the following formula:

$$\Delta Q_0 = \frac{\Delta Q}{K} \tag{8}$$

where ΔQ_0 is the amount of land-use conversion in an iteration interval Δt.

3. Application and simulation

3.1. Test area and data

The BCO–CA model was applied to the simulation of urban expansion in the Xi'an-Xian Yang urban area of China. TM satellite images, with a 30 metre resolution, are used to investigate the urban areas in 2006 and 2009, and a supervised classification method based on the minimum distance is used to obtain the land-use data from remote-sensing images. The classification accuracies are 89.06% and 89.43%, respectively, in 2006 and 2009, and the corresponding kappa coefficients are 0.8022 and 0.8298, respectively. The urban areas are obtained from the classification results and used to confirm the predictability of the CA model. The probability of urban development is influenced by a series of spatial distance variables, neighbourhood conditions and physical attributes (Liu *et al*. 2008), which are derived from remote-sensing and basic geographic information data. These spatial variables for discovering transition rules are listed in Table 2.

For discovering transition rules, the samples are extracted from the classification data through a stratified, random sampling method. Eight thousand samples are randomly selected and divided into two parts: 4000 to be used as training data, and the remainder to be used as test data.

Table 2. Spatial variables required for derivation of transition rules using ABC-Miner.

Spatial variables	Calculate method	Value range
1. Distance variables		
Distance to City Proper (DistC)	Straight Line tools of Arc/Info GRID	0∼28.9 km
Distance to Road (DistR)		0∼15.5 km
Distance to Railway (DistRW)		0∼22.6 km
2. Number of urban cells (NeighbourInfo)	Focal functions of Arc/Info GRID (window size is 3)	0∼9 cells
3. Land-use type (LandUse)		1∼4

3.2. Discovering transition rules

The predictive class of the transition rules is determined by whether the cells are translated into urban land-use. Cells that are translated into urban land-use are marked as 1; and others are marked as 0. The transition rules for urban development can be automatically derived from training data through C# programming, and 13 rules yielded, a portion of which are listed in Table 3. The parameter settings for ABC-Miner are as follows: *Bee_Num* = 400, *Min_Num_Ins* = 5, *Max_IterTimes* = 2500 and *Max_UnImNum* = 200.

It is necessary to test the capability of the BCO in discovering rules. To reach this objective, the rules that are derived by BCO and ACO are applied to the classification of test data. It should be noted that attribute values must be discretised before deriving rules by ACO. In this paper, three discretisation methods, equal interval, equal frequency and information entropy, are applied to discretise the attribute values. As is shown in Table 4, the classification accuracies of ACO vary with different discretisation methods. The classification

Table 3. Part of the transition rules derived by using ABC-Miner.

Rule 1
IF
DistC > = *0.17* & DistC < = *28.8* & DistR < = *15.4* & DistRW < = *22.5* & LandUse = *1*
THEN
class = *0* confidence = *0.70*
END

Rule 2
IF
DistC > = *23.4* & DistC <*27.1* & DistR > = *0.09* & DistR <*15.3* & DistRW > = *12.4* & DistRW < *20.8* & NeighbourInfo <*6* & (LandUse = *1 or* LandUse = *2)*
THEN
class = *1* confidence = *0.80*
END

Rule3
IF
DistC > = *0.7*& DistC < *23.7* & DistR <*14.6* & DistRW < *17.6* & NeighborInfo <*8* & LandUse = *2*
THEN
class = *1* confidence = *0.70*
END

. . . . . .

Table 4. Comparison of accuracy for test data.

ACO			
Equal interval	Equal frequency	Information Entropy	BCO
71.13%	73.35%	79.43%	80.1%

accuracy of ACO based on information entropy is higher than that which is based on other discretisation methods, but it is still lower than that of BCO.

3.3. Land-use simulation and result analysis

The transition rules that are obtained by ABC-Miner are applied to the simulation of urban dynamics in the urban area around Xi'an-Xian Yang during the period of 2006–2009. The simulation begins from the initial land use, which is obtained by classifying the TM data in 2006. The land use in 2009 is then simulated by running the BCO–CA model with 200 iterations, and the process of iteration is listed in Figure 4, where T is the number of iterations.

Validation is usually required if urban CA are applied to the simulation of actual cities (Jenerette and Wu 2001). A simple method to assess the goodness-of-fit is to compare the simulated patterns with the actual ones to visually validate CA (Clarke *et al*. 1997, White *et al*. 1997, Ward *et al*. 2000). Visual comparison indicates that the simulated patterns fit well with the actual patterns that are classified using remote-sensing images (Figure 4).

Visual comparison is a rather preliminary method; therefore, further quantitative analysis is utilised to produce a confusion matrix for the concordance between the simulated and actual land use. The simulation of the 2009 patterns is compared with the actual 2009 patterns, and Table 5 lists the comparison of the patterns in 2009 for this BCO–CA. The total accuracy is computed from cross-tabulation, which compares the goodness-of-fit on a cell-by-cell basis. The total accuracy is 74.7%, and the kappa coefficient is 0.71.

It is noted that a predictive model should be compared with a NULL model of pure persistence (no change) for model validation (Pontius and Malanson 2005). The NULL model is a model that predicts nothing because nothing would change. The baseline for this comparison is that the predictive model should show better performance than a NULL model. Meanwhile, the overall accuracy is biased because of the difference between the actual and chance agreements (Congalton 1991, Liu and Li 2006), which can be effectively explained by the Kappa coefficient. Table 6 lists the comparisons of the patterns in 2009 for this NULL model. As shown in Tables 5 and 6, during the period of 2006–2009, the total accuracy of the BCO–CA model is only 1.2% higher than that of the NULL model, but the kappa coefficient is 5% higher than that of the NULL model. These results indicate that the BCO–CA model is fairly more accurate in simulating urban development.

To further validate the BCO–CA model, it is necessary to compare it with the ACO–CA model, whose rules are derived by the ACO-Miner. For the ACO-Miner, the information entropy is used to discretise the attribute value based on the experimental results in section 3.2.Table 7 compares the patterns in 2009 for the ACO–CA model. As is shown in Tables 5 and 7, the total accuracy of the BCO–CA model is 0.6% higher than that of the ACO–CA model, and the kappa coefficient is 2% higher than that of the ACO–CA model. It is also important to assess the structural features (Wu 2002). The Moran's I index can reveal the

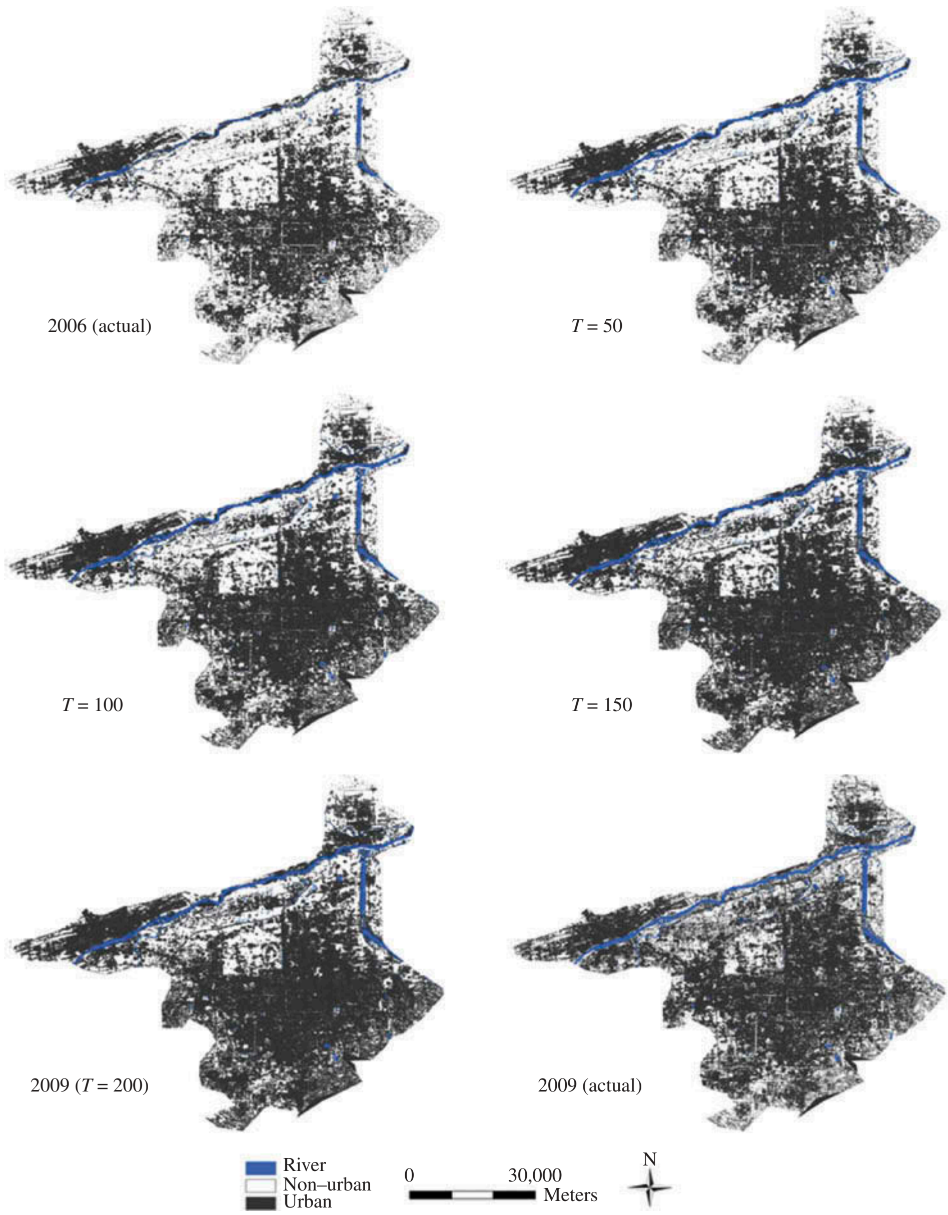

Figure 4. Simulating results.

degree of spatial autocorrelation (Goodchild 1986), and the Moran's I values for the simulation and actual land are shown in Table 8. The Moran's I values are 0.4852 and 0.5249 for the simulation of urban development using ACO–CA and BCO–CA, respectively, in 2009, and the Moran's I value is 0.5348 for the actual land development in 2009. The above results illustrate the capability of BCO–CA for simulating urban development.

Figure 5 further displays the spatial distribution of agreement and disagreement between the simulated and actual development in 2009. Delft blue and Tzavorite green

Table 5. Simulation accuracies of the BCO–CA.

	Simulation 2009 non-urban	Simulation 2009 urban
Actual 2009 non-urban	15,7978	13,0984
Actual 2009 urban	98,710	51,8248
Total Accuracy	74.7%	
Kappa coefficient	0.71	

Table 6. Accuracies of the NULL model of pure persistence.

	2006 non-urban (no change)	2006 urban (no change)
Actual 2009 non-urban	25,6759	32,203
Actual 2009 urban	20,6862	41,0096
Total Accuracy	73.6%	
Kappa coefficient	0.66	

Table 7. Simulation accuracies of the CA model based on ACO.

	Simulation 2009 non-urban	Simulation 2009 urban
Actual 2009 non-urban	15,4478	13,1984
Actual 2009 urban	10,2710	51,6748
Total Accuracy	74.1%	
Kappa coefficient	0.69	

Table 8. Comparison of Moran's I values.

2006 (actual)	2009 (actual)	2009 (ACO)	2009 (BCO)
0.3717	0.5348	0.4852	0.5249

colours indicate these two correctly simulated categories, 'simulated non-urban to actual non-urban' and 'simulated urban to actual urban'. Dark umber and Mars red indicate the two incorrectly simulated categories, 'simulated urban to actual non-urban' and 'simulated non-urban to actual urban'. The BCO–CA model has a higher percentage of accuracy than the NULL model because more Mars red than Ginger pink is observed.

4. Conclusion

CA is capable of simulating the evolution of complex geographical phenomena using a set of local rules. This paper presents an intelligent method (BCO) for obtaining the transition rules of geographic CA and structures in a CA model (BCO–CA), which is based on transition rules. With positive feedback, negative feedback, fluctuations and a multiple interaction mechanism, the BCO is an effective method for discovering the reliable transition rules that reveal complex relationships that are hidden in large datasets, and especially, with local and global search mechanisms, the BCO performs better in avoiding the tendency to be vulnerable to local optimisation.

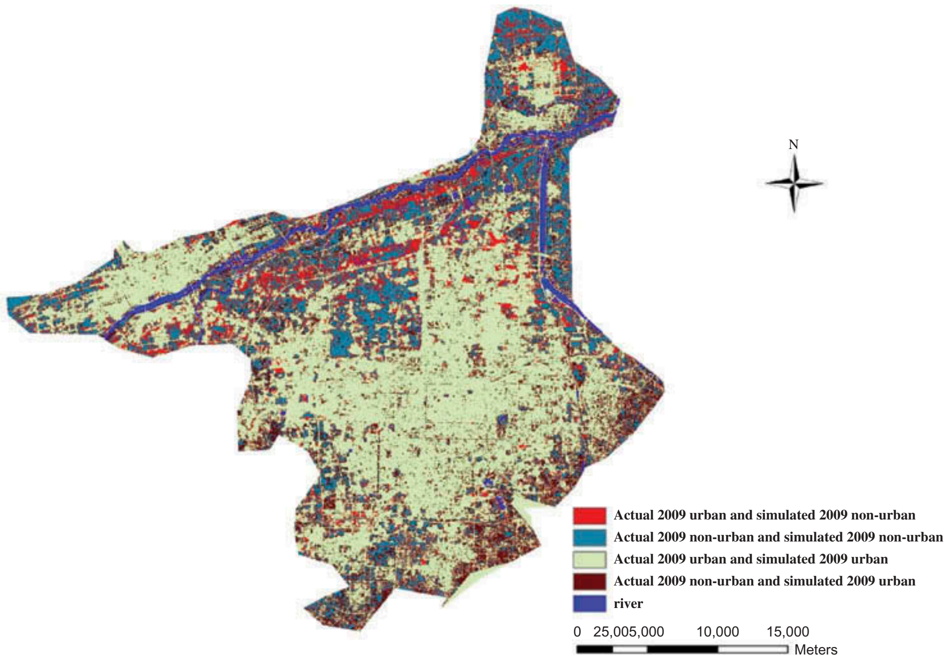

Figure 5. The distribution of agreement and disagreement between simulation and actual.

The rules derived by BCO and ACO are applied to the classification of test data to demonstrate the advantages of BCO in discovering transition rules. Three methods, i.e. equal interval, equal frequency and information entropy, are used to discretise the attribute values for deriving rules by the ACO. The classification accuracies of ACO that are based on equal interval, equal frequency and information entropy are, respectively, 71.13%, 73.35% and 79.43%. By contrast, the classification accuracy of the BCO is 80.1%. These results indicate that the BCO is a better approach for discovering the transition rules of CA.

The BCO–CA model is applied to the simulation of urban conversions in urban areas near Xi'an-Xian Yang, China. The total accuracy of the BCO–CA model is 74.7%, which is 1.1% and 0.6% higher than those of the NULL and ACO–CA model, respectively. The kappa coefficient of the BCO–CA model is 0.71, which is 0.05 and 0.02 higher than those of the NULL and ACO–CA models, respectively. The Moran's I value of BCO–CA is 0.5249, which is 0.04 higher than that of the ACO–CA model. These results prove that the BCO–CA model has great potential in simulating complex geographical systems.

Acknowledgements

This study was supported by the National Science Foundation of China (Grant No. 40930531, 41101349, and 41171320), the Research Fund of the State Key Laboratory of Resources and Environmental Information System, Chinese Academy of Science (Grant No. 2010KF0002SA), the Major University Science Research Project of Jiangsu Province (Grant No.13KJA170001), a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions, China and The Natural Science Fundamental Research Program for colleges and universities in Jiangsu Province (Grant No.13KJB420003).

References

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

Chen, S.M., Sarosh, A., and Dong, Y.F., 2012. Simulated annealing based artificial bee colony algorithm for global numerical optimization. *Applied mathematics and computation*, 219, 3575–3589.

Clarke, K.C., Brass, J.A., and Riggan, P.J., 1994. A cellular automata model of wildfire propagation and extinction. *Photogrammetric Engineering & Remote Sensing*, 60, 1355–1367.

Clarke, K.C. and 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., and Gaydos, L.J., 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.

Couclelis, H., 1985. Cellular worlds: a framework for modeling micro macro dynamics. *Environ Plann A*, 17, 585–596.

Couclelis, H., 1988. Of mice and men: what rodent population scan teach us about complex spatial dynamics. *Environ Plann A*, 20, 99–109.

Couclelis, H., 1989. Macrostructure and micro behavior in a metropolitan area. *Environ Plann B*, 16, 141–154.

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.

Duan, H.B., Xu, C.F., and Xing, Z.H., 2010. A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems. *International Journal of Neural System*, 20, 39–50.

GoodChild, M., 1986. *Spatial Autocorrelation, CATMOG, 47*. Norwich, UK: GeoBooks.

Hargrove, W.W., *et al*., 2000. Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling*, 135, 243–263.

Jenerette, G.D. and Wu, J.G., 2001. Analysis and simulation of land-use change in the central Arizona—Phoenix region, USA. *Landscape Ecology*, 16, 611–626.

Karaboga, D., 2005. *An idea based on honey bee swarm for numerical optimization*. Kayseri: Erciyes University.

Karaboga, N., 2009. A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute*, 346, 328–348.

Karaboga, D. and Akay, B. 2007. Artificial bee colony algorithm on training artificial neural networks. Proceeding of the 15th IEEE Signal Processing and Communications Applications Conference, June 11–13, 2007, Eskisehir, Turkey. 1–4.

Karaboga, D., Akay, B., and Ozturk, C. 2007. Artificial bee Colony (ABC) optimization algorithm for training feed-forward neural networks. Proceedings of Modeling Decisions for Artificial Intelligence Conference, August 16–18, 2007, Kitakyushu, Japan. 318–319.

Karaboga, D. and Basturk, B., 2007a. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39, 459–471.

Karaboga, D. and Basturk, B. 2007b. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. Proceeding of Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, June 18–21, 2007, Cancun, Mexico. 789–798.

Karaboga, D. and Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8, 687–697.

Karaboga, D., *et al*., 2012. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, doi:10.1007/s10462-012-9328-0. 1–12.

Li, X. and 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.

Li, X. and Yeh, A.G.O., 2002. Neural-network-based cellular automata for simulating multiple land use changes using GIS. *International Journal of Geographical Information Science*, 16, 323–343.

Li, X. and Yeh, A.G.O., 2004. Data mining of cellular automata's transition rules. *International Journal of Geographical Information Science*, 18, 723–744.
Liu, X.P. and Li, X., 2006. Retrieving CA nonlinear transition rule from high-dimensional feature space. *Acta Geographical Sinica*, 61, 663–672. (in Chinese).
Liu, X.P., *et al*., 2008. A bottom-up approach to discover transition rules of cellular automata using ant intelligence. *International Journal of Geographical Information Science*, 5, 37–41.
Pontius, G.R. and Malanson, J., 2005. Comparison of the structure and accuracy of two land change models. *International Journal of Geographical Information Science*, 19, 243–265.
Seeley, T.D., 1995. *The wisdom of the Hive: the social physiology of honey bee colonies*. Cambridge: Harvard University Press.
Shukran, M.A.M., *et al*., 2011. Artificial Bee Colony based Data Mining Algorithms for Classification Tasks. *Modern Applied Science*, 5, 217–231.
Soiares-filho, B.S., Cerqueira, G.C., and Pennachin, C.L., 2002. DINAMIC—a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier. *Ecological Modelling*, 154, 217–235.
Wang, Y. and Zhang, X., 2001. A dynamic modeling approach to simulating socioeconomic effects on landscape changes. *Ecological Modelling*, 140, 141–162.
Ward, D.P., Murray, A.T., and Phinn, S.R., 2000. A stochastically constrained cellular model of urban growth. *Computers, Environment and Urban Systems*, 24, 539–558.
White, R. and Engelen, G., 1993. Cellular automata and fractal urban form: a cellular modeling approach to the evolution of urban land-use patterns. *Environment and Planning A*, 25, 1175–1199.
White, R., Engelen, G., and Uijee, I., 1997. The use of constrained cellular automata for high-resolution modelling of urban land use dynamics. *Environment and Planning B*, 24, 323–343.
Wu, F., 1998. SimLand: a prototype to simulate land conversion through the integrated GIS and CA with AHP-derived transition rules. *International Journal of Geographical Information Science*, 12, 63–82.
Wu, F., 2002. Calibration of stochastic cellular automata: the application to rural–urban land conversions. *International Journal of Geographical Information Science*, 16, 795–818.
Wu, F. and 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.
Yang, X., Zheng, X.Q., and Lv, L.N., 2012. A spatiotemporal model of land use change based on ant colony optimization, Markov chain and cellular automata. *Ecological Modelling*, 233, 11–19.
Yeh, A.G.O. and Li, X., 2006. Errors and uncertainties in urban cellular automata. *Computers Environment and Urban Systems*, 30, 10–28.