Short Communication

Computer generation of initial spatial distribution for cell automata

GuangHua Liu[1], WenJun Zhang[2]

[1]Guangzhou Experimental Station, Chinese Academy of Tropical Agricultural Sciences, Guangzhou 510140, China; Guangdong AIB Polytech College, Guangzhou 510507, China

[2]School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong

E-mail: ghliu@gdaib.edu.cn, zhwj@mail.sysu.edu.cn

Received 8 May 2011; *Accepted 13 June 2011*; *Published online 1 December 2011*

IAEES

Abstract

The algorithm to generate spatial distribution patterns was developed and presented in this study. Three typical spatial distribution patterns, i.e., Poisson distribution, binomial distribution, and negative binomial distribution were included in the algorithm. The Java program was also provided. The algorithm can be used to generate initial distribution in cell automata modeling.

Keywords spatial distribution patterns; computer generation; cell automata.

1 Introduction

Spatial distribution pattern refers mainly to the two-dimensional distribution of individuals in a region. There are many types of spatial distribution patterns (Krebs, 1999; Zhang, 2007). They can be classified into three categories, namely random distribution, uniform distribution, and aggregated distribution, which can be represented by Poisson distribution, binomial distribution and negative binomial distribution, respectively. In this study I developed and presented an algorithm and Java program to generate spatial distribution patterns for cell automata uses (Qi et al., 2011).

2 Algorithms

2.1 Spatial distribution patterns

Supposedly there are three types of spatial distribution patterns, binomial distribution, Poisson distribution and negative binomial distribution (Krebs, 1999; Zhang, 2007). They represent the uniform distribution, random distribution and aggregated distribution, respectively. Recursive formulae for calculating probability of binomial distribution are:

$$p_r=q^n, \qquad r=0;$$
$$p_r=p(n-r+1)p_{r-1}/(r\,q), \qquad r=1, 2, \ldots, n,$$

where, p: probability of a individual occurred in a cell, $p=\sum if_i/(n\sum f_i)$, $q=1-p$, $i=0, 1, 2, \ldots, n$; n is the maximum possible number of individuals in a cell, f_i is the number of cells with i individuals, $i=0, 1, 2, \ldots, n$.

Recursive formulae for calculating probability of Poisson distribution are:

$$p_r = e^{-m}, \qquad r=0;$$
$$p_r = m\, p_{r-1}/r, \qquad r>0,$$

where, m: mean of the number of individuals in a cell.

In the negative binomial distribution, the number of clusters per cell is Poisson distribution, the mean number of clusters is $k\times\ln Q$, and the number of individuals per cluster is the logarithmic distribution. The recursive formulae are:

$$p_r=(1-1/Q)^r/(r\,|\ln(1/Q)|), \quad r>0,$$

where,

$$Q=1+m/k,\ k=m^2/(s^2-m).$$

and, m: mean number of individuals, s^2: variance of the number of individuals.

2.2 Algorithm for generation of spatial distribution patterns

Given p_r, r=0, 1, 2, …, are obtained from the formulae above, and there are N cells. Then the number of cells with r individuals inside the cell is int (p_r+a) , r=0, 1, 2, …, M-1, and:

$$\sum \mathrm{int}(N\, p_r+a)=N,$$

where the value of a, $0<a<1$, makes M the limited number, and int means to get integer number. Coding all cells with the ID number, 1, 2, …, N. Similar to the Monte Carlo simulation used by Ferrarini (2011), generate a number between 1 to N using random number generator, e.g., $V_{b'}$, and let $Q_1=V_{b'}$. Remove $V_{b'}$, and re-coding the remaining N-1 numbers, and generate a number between 1 to N, e..g., $V_{a'}$. Repeat this process until $Q_N= V_{q'}$. As a result, the number of individuals in cells Q_1, Q_2, …, and $Q_{\mathrm{int}(p0N+a)}$ are zeros, and the number of individuals in cells $Q_{\sum\mathrm{int}(prN+a)+1}$, …, and $Q_{\sum\mathrm{int}(prN+a)}$ are i, i=1, 2, …, M-1, where for the first summation term, r=0, 1, 2, …, i-1, and for the last summation term, r=0, 1, 2, …, i. For negative binomial distribution, use the above algorithm in each cell. An initial spatial distribution is thus generated.

3 Java Implementation

The algorithm above is implemented by a Java program (based on JDK 1.1.8) (Zhang, 2011), in which seven classes and an HTML file is included (http://www.iaees.org/publications/software/index.asp). The class, BioDistriProducer, performs all calculation, while the other class is called together to complete the entire task. The user will be asked to choose the type of spatial distribution patterns, to enter the required number of spatial distribution patterns, the number of cells along the x-axis and y-axis (Fig. 1). The codes for generating the map of spatial distribution, ran(), and for calculating spatial distribution, subpr(), are listed as follows:

```
public int[] ran() {
ep=0.5;
int s2;
s2=(int)Math.round(p[0]*na+ep);
r=1;
s2+=subpr();
while (s2!=na) {
if (s2<na) {r++;
if (r>n) {ep+=0.5;
```

```
s2=(int)Math.round(p[0]*na+ep);
r=1;
s2+=subpr(); }
else if (r<=n) s2+=subpr(); }
else if (s2>na) {ep-=0.05;
s2=(int)Math.round(p[0]*na+ep);
r=1;
s2+=subpr(); } }
for(i=0;i<=r;i++) p[i]=Math.round(p[i]*na+ep);
s2=0;
for(i=1;i<=na;i++) x[i]=i;
for(i=1;i<=na;i++) {
if (i==na) break;
q[i]=(int)Math.round(Math.random()*(na-s2)+0.5);
h[i]=x[q[i]];
for(j=q[i]+1;j<=na-s2;j++) x[j-1]=x[j];
s2++; }
h[na]=x[1];
s2=1;
for(j=0;j<=r;j++) {
if (p[j]==0) continue;
for(int k1=0;k1<=p[j]-1;k1++) {x[h[s2]]=j; s2++;} }
test: for(i=1;i<=na;i++) {
for(j=1;j<=na;j++) if (h[j]==i) continue test;
x[i]=x[0]; break; }
return x; }

public int subpr() {
if (lm==1) p[r]=m/r*p[r-1];
else if (lm==2) p[r]=(nm-r+1)*p1*p[r-1]/(r*q1);
else if (lm==3) p[r]=(k+r-1)*m/k/(r*q1)*p[r-1];
return (int)Math.round(p[r]*na+ep); }
```

Further, a window will pop up to wait for entering parameters. Enter the mean number of individuals in a cell if Poisson distribution pattern was chosen. If binomial distribution pattern was chosen, enter the probability of an individual occurred in a cell, and the maximum possible number of individuals in a cell. For negative binomial distribution, enter the mean number of individuals in a cell and its variance.

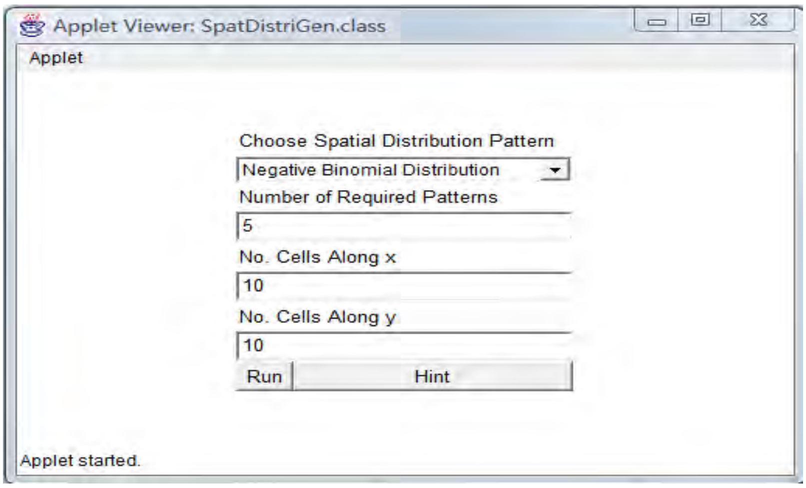

Fig. 1 An interface for the program

4 Application

Suppose the cell automata have 100 (10×10) cells. The mean number of individuals is 100 for Poisson distribution and mean number of individuals is 200 and variance is 20000 for negative binomial distribution (Fig. 2). A Poisson distribution pattern and a negative binomial distribution pattern, generated by SpatDistriGen, are as follows (and Fig. 2):

Poisson distri.

97	96	103	101	90	102	103	84	109	92
93	94	97	106	101	78	107	98	104	90
106	95	94	79	108	89	88	85	92	97
99	101	100	98	94	100	95	89	91	87
110	102	103	110	74	101	96	93	103	104
109	106	85	105	77	97	102	92	81	76
83	100	93	96	99	105	86	94	100	98
82	91	102	98	95	80	106	104	109	93
99	107	107	86	89	84	75	91	104	99
110	90	96	108	87	105	88	95	108	105

Nega. bino. distri.

13	44	70	76	11	79	33	38	68	52
4	40	27	31	16	84	14	86	60	17
74	36	97	25	48	21	57	23	34	41
45	39	6	35	90	63	59	12	55	83
65	99	7	19	58	89	43	85	28	30
80	72	20	32	82	5	8	51	29	77
56	2	61	50	96	49	10	22	46	26
66	37	3	69	18	9	88	62	64	93
47	87	95	92	71	24	98	54	53	75
94	67	73	0	91	78	1	42	15	81

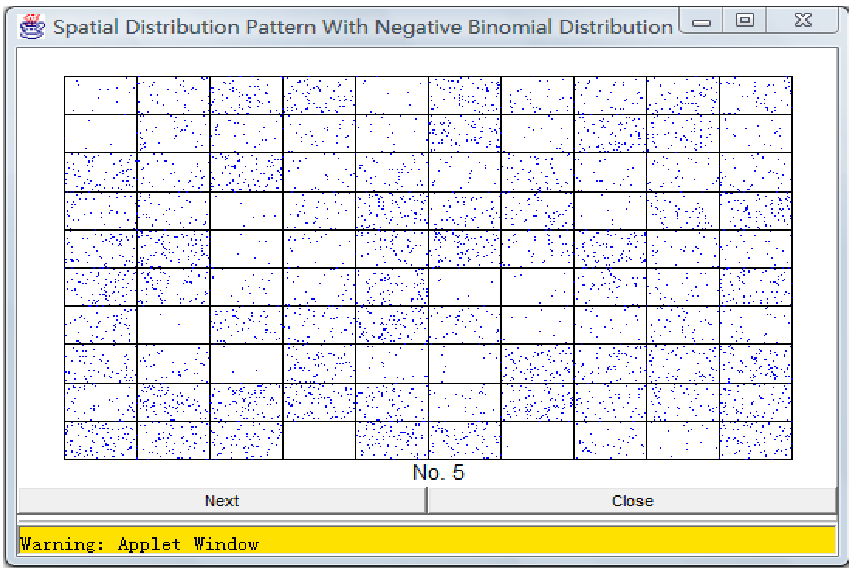

Fig. 2 Graphic output of initial spatial distribution.

The aggregation index, m^*/m, of Poisson distribution above is 1.009 and of negative binomial distribution is 1.3437. Based on the criteria for aggregation determination, the Poisson and negative binomial distributions above are random and aggregated distributions respectively, which is exactly correct.

References

Ferrarini A. 2011. A fitter use of Monte Carlo simulations in regression models. Computational Ecology and Software, 2011, 1(4): 266-269

Krebs CJ. 1999. Ecological Methodology (2nd edition). Addison-Wesley Educational Publishers, Inc., USA

Qi YH, Zhang WJ, Zhang ZG. 2012. Cell automata of pest infestation in heterogeneous habitat. Network Biology, 2(1) (to be published)

Zhang WJ. 2007. Methodology on Ecology Research. Sun Yat-sen University, Guangzhou, China

Zhang WJ. 2011. A Java program for non-parametric statistic comparison of community structure. Computational Ecology and Software, 2011, 1(3): 183-185