Article

A random network based, node attraction facilitated network evolution method

WenJun Zhang
School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong
E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Received 6 August 2015; *Accepted 28 September 2015*; *Published online 1 March 2016*

Abstract
In present study, I present a method of network evolution that based on random network, and facilitated by node attraction. In this method, I assume that the initial network is a random network, or a given initial network. When a node is ready to connect, it tends to link to the node already owning the most connections, which coincides with the general rule (Barabasi and Albert, 1999) of node connecting. In addition, a node may randomly disconnect a connection i.e., the addition of connections in the network is accompanied by the pruning of some connections. The dynamics of network evolution is determined of the attraction factor λ of nodes, the probability of node connection, the probability of node disconnection, and the expected initial connectance. The attraction factor of nodes, the probability of node connection, and the probability of node disconnection are time and node varying. Various dynamics can be achieved by adjusting these parameters. Effects of simplified parameters on network evolution are analyzed. The changes of attraction factor λ can reflect various effects of the node degree on connection mechanism. Even the changes of λ only will generate various networks from the random to the complex. Therefore, the present algorithm can be treated as a general model for network evolution. Modeling results show that to generate a power-law type of network, the likelihood of a node attracting connections is dependent upon the power function of the node's degree with a higher-order power. Matlab codes for simplified version of the method are provided.

Keywords network evolution; node attraction; connection probability; disconnection.

Selforganizology
ISSN 2410-0080
URL: http://www.iaees.org/publications/journals/selforganizology/online-version.asp
RSS: http://www.iaees.org/publications/journals/selforganizology/rss.xml
E-mail: selforganizology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

In 1998, Watts and Strogatz presented a method for generating random graphs. Thereafter, Barabasi and Albert (1999) proposed a general and known mechanism for network evolution. The algorithm developed by Cancho and Sole (2001) can generate a variety of complex networks with diverse degree distributions. Zhang (2012a, 2012b, 2015, 2016) proposed a series of methods for network generation and evolution. In present study, I will propose method of network evolution that based on random network, and facilitated by node attraction. Deep analyses will be implemented to understand the properties of the method.

2 Algorithm

I assume that the initial network is a random network (this is popular in nature. For example, the distribution of particles with different sizes with which the water vapor attaches to generate droplets; the random distribution of matter/nucleus of crystallization for the generation of stars/crystals, etc.), or a given initial network. When a node is ready to connect, it tends to link to the node already owning the most connections (the largest node), which coincides with the general rule of node connecting (Barabasi and Albert, 1999). In addition, a node may randomly disconnect a connection, i.e., the addition of connections in the network is accompanied by the pruning of some connections.

Assume there are totally v nodes in the network. Expected initial connectance (connectance=practical connections/potential maximum connections) is c (initial condition) if the initial network is a random network generated by the algorithm, expected final connectance is c_e (termination condition), the attraction factor of nodes is $\lambda(t,a)$ (driving variable; $\lambda(t,a)>0$, where a is the node degree), the probability of node connection is $p(t,a)$ (driving variable), the probability of node disconnection is $q(t,a,b)$ (driving variable, where b is the connected node's degree), maximum number of iterations is *iter* (termination condition), and the confidence degree for detecting the statistic significance of network type is α (auxiliary constant). $\lambda(t,a)$, $p(t,a)$, and $q(t,a,b)$ are time and node (in particular node degree) varying. The procedures are as follows

(1) Generate the initial network. In the situation of random initial network, assume the adjacency matrix of the random network is $d=(d_{ij})$, i, $j=1,2,\ldots,v$, where $d_{ij}=d_{ji}$, $d_{ii}=0$, and if $d_{ij}=1$ or $d_{ji}=1$, there is a connection between nodes i and j. For each pair of i, j ($i=1,2,\ldots,v-1$; $j>i$), generate a random value r, if $r<c$, $d_{ij}=1$ and $d_{ji}=1$. Otherwise, the initial network is a given network.

(2) Let $t=1$. Calculate the degree of node, $a_i(t)$, $i=1,2,\ldots,v$. The cumulative attraction strength of node 1 to node i is

$$p_i(t) = \sum_{j=1}^{i} a_j(t)^{\lambda(t,a_j)} / \sum_{j=1}^{v} a_j(t)^{\lambda(t,a_j)}$$

(3) Generate/disconnect connections. For the node i, $i=1,2,\ldots,v$, generate a random value s, if $s<p(t,a_i)$, the node i is ready to connect to one of the remaining nodes. Let $p_0(t)=0$. For the node j, $j=1, 2,\ldots,v$, $j\neq i$, generate a random value w. $d_{ij}=1$ and $d_{ji}=1$, if $p_{i-1}(t)\leq w<p_i(t)$. In the practical uses, the interval $[p_{i-1}(t), p_i(t))$ represents the mass or volume of the particle i, the gravity of the celestial body i, the personality charm of the person i, the academic impact of the scientist i, etc.

For the node i, $i=1,2,\ldots,v$, generate a random value g; if $g<q(t,a_i,b)$, one of the connections of the node i, e.g., d_{ij}, is randomly disconnected, and let $d_{ij}=0$ and $d_{ji}=0$.

By doing so, a network at time t is generated. Various indices and methods, e.g., coefficient of variation (CV), aggregation index (AI), and entropy (Zhang and Zhan, 2011; Zhang, 2012a) can be used to detect the types and properties of the network.

(4) Calculate the connectance C of the network. Let $t=t+1$ and return (2), if C is less than the expected final connectance c_e; otherwise, the algorithm terminates, if C is not less than c_e, or the maximum iterations *iter* are achieved.

For convenience and simplicity, assume $\lambda(t,a)=\lambda$, $p(t,a)=p$, $q(t,a,b)=q$, i.e., the attraction factor of nodes, the probability of node connection, and the probability of node disconnection are constants for any degree of nodes at any time. Thus we obtain a simplified version of the algorithm.

The following are Matlab codes for simplified version of the algorithm (netEvolution.m)

```
%Reference: Zhang WJ. 2016. A random network based, node attraction facilitated network evolution method. Selforganizology, 3(1): 1-9
v=input('Total number of nodes in the network= ');
choice=input('Input the type (1: a random network generated by the algorithm; 2: a given network) of generating initial network: ');
if (choice==1) ci=input('Expected initial connectance (=practical connections/potential maximum connections; e.g., 0.05, etc)= ');
end
if (choice==2) adjstr=input('Input the file name of adjacency matrix of the given initial network (e.g., raw.txt, raw.xls, etc. Adjacency matrix is d=(dij)v*v, where v is the number of nodes in the network. dij=1, if vi and vj are adjacent, and dij=0, if vi and vj are not adjacent; i, j=1,2,…, v: ','s'); end
ce=input('Expected final connectance (=practical connections/potential maximum connections; e.g., 0.1, 0.15, etc)= ');
lamda=input('Attraction factor of nodes (lamda; e.g., 2, 4, etc. lamda>0)= ');
p=input('Probability of node connection (e.g., 0.1, 0.2)= ');
q=input('Probability of node disconnection (e.g., 0, 0.01)= ');
alpha=input('Confidential degree for detecting network type (e.g., 0.05, 0.01)= ');
iter=input('Permitted maximum iterations (e.g., 5000)= ');
adj=zeros(v);
degr=zeros(1,v);
prop=zeros(1,v);
z=zeros(v);
if (choice==1)
for i=1:v-1
for j=i+1:v
if (rand()<ci) adj(i,j)=1; adj(j,i)=1; end
end; end; end
if (choice==2) adj=load(adjstr); end;
degr=sum(adj);
fprintf('Initial adjacency matrix\n')
disp([adj])
fprintf('Initial degree distribution\n')
disp([degr])
 t=1;
while (v>0)
propdegr=degr.^lamda;
prop(1)=propdegr(1)/sum(propdegr);
for i=2:v;
prop(i)=prop(i-1)+propdegr(i)/sum(propdegr);
end
node=zeros(1,v);
nu=0;
for i=1:v
if (rand()<p) nu=nu+1; node(nu)=i; end
end
for k=1:nu;
for i=1:v
```

```
if (node(k)~=i) continue; end
lab=0;
ran=rand();
for j=1:v
if (j==i) continue; end
if (j==1) st=0; end
if (j>=2) st=prop(j-1); end
if ((ran>=st) & (ran<prop(j))) lab=1; adj(i,j)=1; adj(j,i)=1; break; end
end
if (lab==1) break; end
end; end
nodes=zeros(1,v);
for i=1:v
if (rand()<q) nodes(i)=1; end
end
for i=1:v-1
if (nodes(i)~=1) break; end
nuu=0;
for j=i+1:v
if (adj(i,j)==1) nuu=nuu+1; z(i,j)=nuu; end
end
np=round(rand()*nuu+0.5);
for j=i+1:v
if (z(i,j)==np) adj(i,j)=0; adj(j,i)=0; end
end; end
fprintf(['\n\nTime ' num2str(t)])
fprintf(['\n\nAdjacency matrix\n'])
disp([adj])
degr=sum(adj);
fprintf('\nDegree distribution\n')
disp([degr])
cnow=(sum(degr)/2)/((v^2-v)/2);
fprintf(['\nConnectance=' num2str(cnow) '\n'])
meann=mean(degr);
varr=(std(degr))^2;
fprintf(['\nEntropy=' num2str(varr-meann) '\n'])
num=0;
cv=varr/meann;
fprintf(['\nCoefficient of variation (CV)=' num2str(cv) '. '])
x2=cv*(v-1);
sig=chi2cdf(x2,v-1);
if (sig<=alpha) fprintf('The network is a random network according to CV.\n'); end
if ((sig>alpha) & (cv>1)) fprintf('The network is a complex network according to CV.\n'); num=num+1; end;
summ=sum(degr);
summa=sum(degr.*(degr-1));
```

```
h=v*summa/(summ*(summ-1));
fprintf(['\nAggregation index (AI)=' num2str(h) '. '])
x2=h*(summ-1)+v-summ;
sig=chi2cdf(x2,v-1);
if (sig<=alpha) fprintf('The network is a random network according to AI.\n'); end;
if ((sig>alpha) & (h>1)) fprintf('The network is a complex network according to AI.\n'); num=num+1; end;
if (num>=2) fprintf('\nThe network is a complex network according to all indices.\n'); end;
if (cnow>=ce) break; end
if (t>=iter) break; end;
t=t+1;
end
```

3 Results and Analysis

In present study, for the convenience, the initial network is a random network generated by the algorithm.

3.1 Dynamics of network evolution

Running the simplified version of the network evolution algorithm, Fig. 1 illustrates a case of the dynamics of properties of the network in the evolution. The parameters were set as v=30, c=0.05, c_e=0.2, λ=3, p=0.2, q=0, α=0.05. In this case, the connected nodes increase quickly until all nodes are connected. Connectance of the network increases with the time. Coefficient of variation (CV) increases quickly to an approximate upper asymptote. However, aggregation index increases quickly to a climax and drops down thereafter. Generally, the three indices increase most quickly until all nodes are connected.

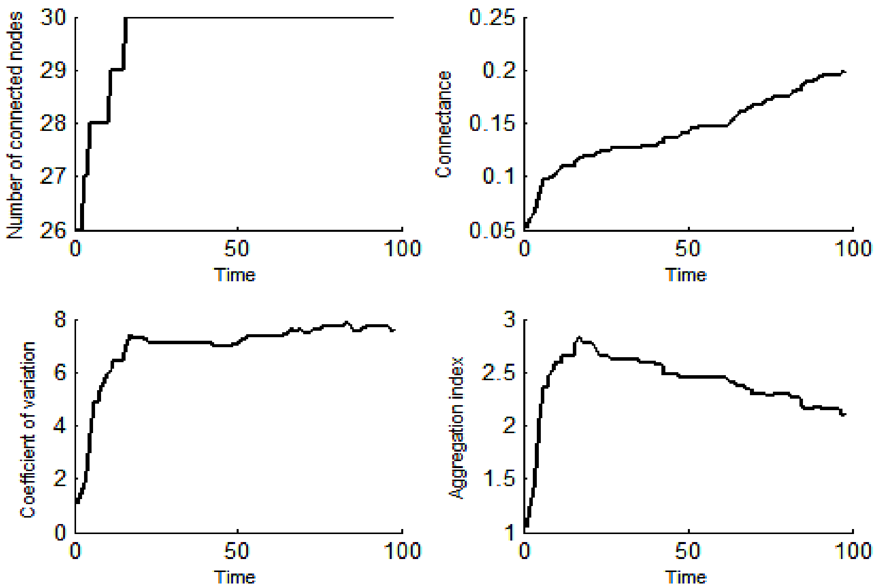

Fig. 1 Property dynamics of network evolution (v=30, c=0.05, c_e=0.2, λ=3, p=0.2, q=0, α=0.05).

3.2 Parametrical analysis

3.2.1 Effects of the attraction factor of nodes λ

Fixing v=30, c=0.05, c_e=0.2, p=0.2, q=0, α=0.05, and running the simplified version of the network evolution algorithm, analyze the effects of the attraction factor of nodes λ on network evolution. Fig. 2 indicates that for λ=0.05, 1, 2, 3, and 4, connectance of the network increases more quickly if λ is smaller. Coefficient of variation and aggregation index increases more quickly if λ is greater.

Table 1 indicates the results of properties of network in the last iteration (expected final connectance is achieved). It can be found that required iterations increase with λ. With the same expected final connectance, the coefficient of variation, aggregation index, and entropy increase with λ, which means the complexity of the final network increases with λ.

3.2.2 Effects of the probability of node connection *p*

Fixing v=30, c=0.05, c_e=0.2, λ=3, q=0, analyze the effects of the probability of node connection p on network evolution. Fig. 3 indicates that for p=0.01, 0.05, 0.1, and 0.2, the three indices increase more quickly if p is greater.

3.2.3 Effects of the probability of node disconnection *q*

Fixing v=30, c=0.05, p=0.2, c_e=0.2, λ=3, analyze the effects of the probability of node disconnection q on network evolution. Fig. 4 indicates that for q=0, 0.01, 0.02, and 0.03, connectance of the network increases more quickly if q is smaller. Coefficient of variation and aggregation index increases more quickly if q is greater.

3.2.4 Effects of expected initial connectance *c*

Fixing v=30, c_e=0.2, λ=3, p=0.2, q=0, analyze the effects of the expected initial connectance c on network evolution. Fig. 5 indicates that for c=0.01, 0.03, 0.05, and 0.08, connectance of the network increases more quickly if c is greater. Coefficient of variation and aggregation index increases more quickly if c is smaller.

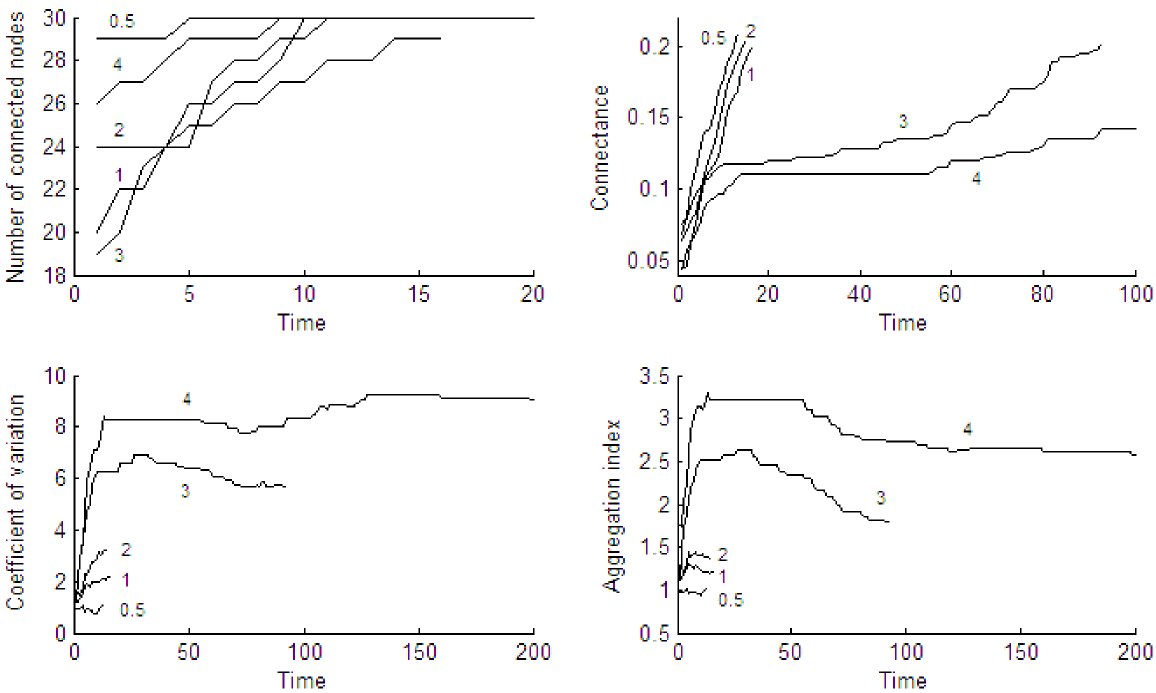

Fig. 2 Effects of the attraction factor of nodes λ (λ=0.5,1,2,3,4), fixing v=30, c=0.05, c_e=0.2, p=0.2, and q=0.

Table 1 The changes of network properties for the last iteration over λ, fixing v=30, c=0.05, c_e=0.2, p=0.2, q=0, α=0.05.

λ	0.01	0.5	1	2	3	4
t	11	13	16	15	93	323
Connectance	0.20	0.21	0.20	0.20	0.20	0.20
Entropy	-2.14	0.27	6.92	12.82	27.98	44.50
Coefficient of Variation (CV)	0.64	1.05	2.19	3.16	5.77	8.67
Network type (CV)	Random network	Complex network	Complex network	Complex network	Complex network	Complex network
Aggregation Index (AI)	0.94	1.01	1.19	1.35	1.79	2.29
Network type (AI)	Random network	Complex network	Complex network	Complex network	Complex network	Complex network
Network type (Overall judge)	Random network	Complex network	Complex network	Complex network	Complex network	Complex network

λ	0.01			0.5			1			2			3			4		
Degree distribution	5	7	5	7	8	5	10	8	9	4	8	1	4	2	3	6	3	3
	2	10	5	8	5	11	1	11	10	4	2	10	8	10	3	3	5	3
	4	7	4	3	5	5	2	5	3	5	2	6	4	6	5	2	4	3
	8	5	6	6	5	8	9	10	2	6	10	6	4	29	2	2	2	7
	7	8	2	4	8	6	1	0	9	1	5	8	10	2	4	4	3	3
	4	6	6	1	6	11	3	10	10	6	3	2	3	4	6	3	5	2
	4	8	7	5	2	4	9	2	4	11	3	4	3	5	2	3	20	4
	5	6	8	6	8	6	8	1	2	8	7	7	4	2	4	2	4	29
	7	7	7	8	4	9	7	4	7	11	4	6	5	6	4	29	3	3
	9	3	6	2	6	10	4	4	9	23	3	2	22	3	7	6	4	4

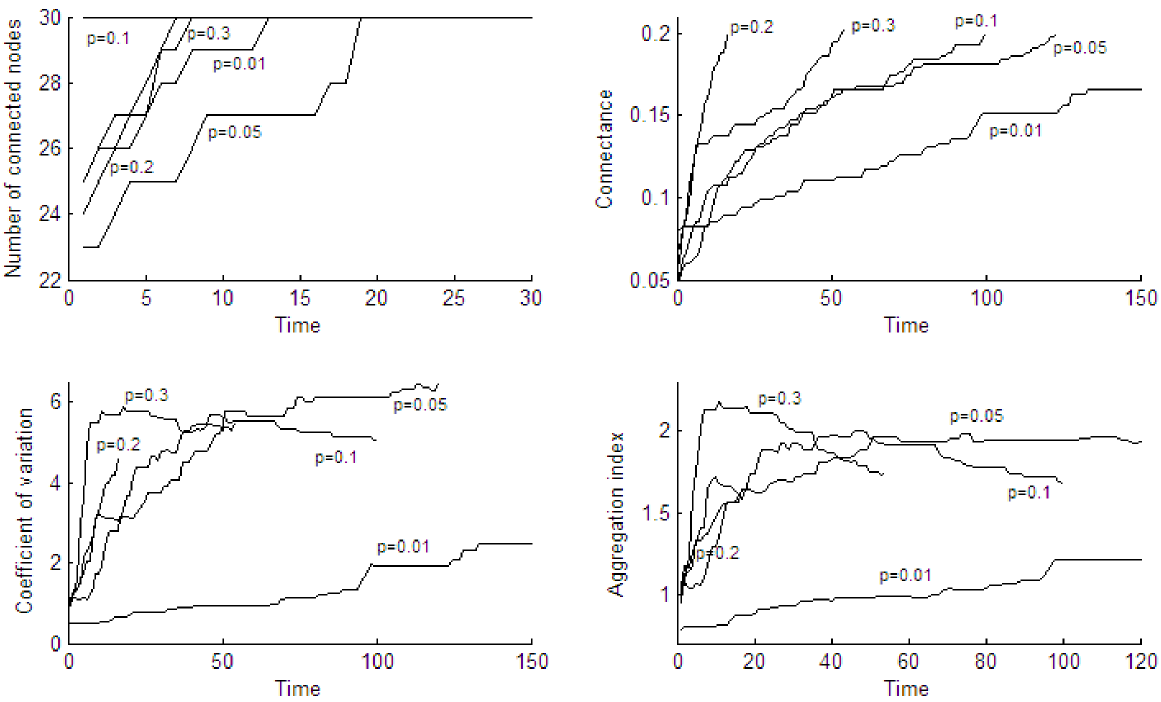

Fig. 3 Effects of the probability of node connection p (p=0.01,0.05,0.1,0.2), fixing v=30, c=0.05, c_e=0.2, λ=3, and q=0.

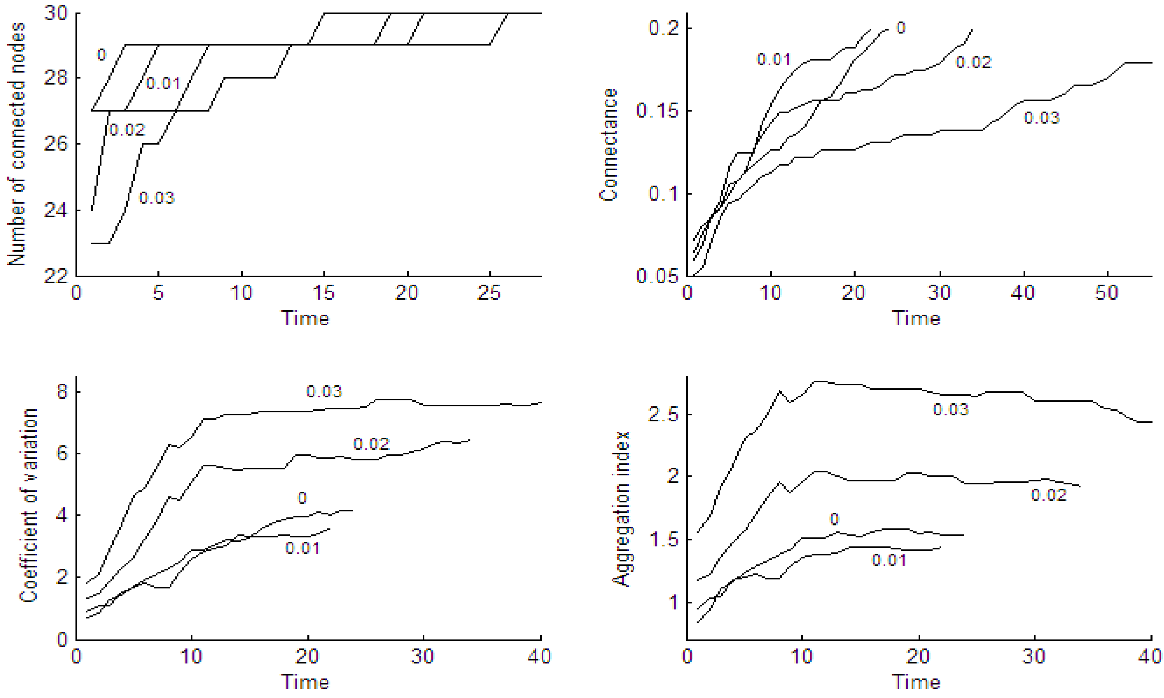

Fig. 4 Effects of the probability of node disconnection *q* (*q*=0,0.01,0.02,0.03), fixing *v*=30, *c*=0.05, c_e=0.2, *p*=0.2, and λ=3.

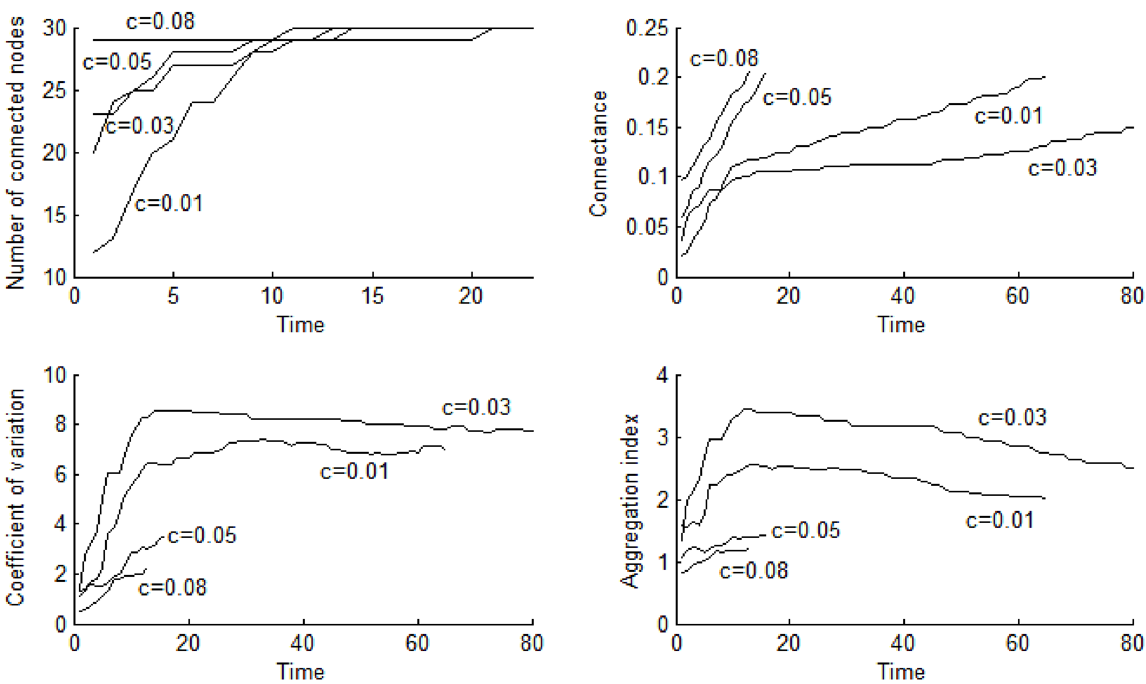

Fig. 5 Effects of expected initial connectance *c* (*c*=0.01,0.03,0.05,0.08), fixing *v*=30, c_e=0.2, λ=3, *p*=0.2, and *q*=0,

3.3 Model's universality

As mentioned above, the changes of attraction factor λ can reflect various effects of the node degree on connection mechanism (Table 1, Fig. 2). The larger λ will lead to generate complex networks, e.g., exponential law, power law networks, etc. $\lambda \to 0$ means a trend to generate the random network. Even the changes of λ only will generate various networks from the random to the complex (Table 1). Therefore, the present algorithm can be treated as a general model for network evolution.

Modeling results (Table 1) show that to generate power-law distributed node degrees (i.e., to generate a power-law type of network), the likelihood of a node attracting connections is dependent upon the power function of the node's degree with a higher-order power.

4 Discussion

In present method, the dynamics of network evolution is determined of $\lambda(t,a)$, $p(t,a)$, $q(t,a,b)$, and c. I simplify all parameters as constants. However, more complex mechanism for network evolution can be achieved by setting reasonable forms of $\lambda(t,a)$, $p(t,a)$, $q(t,a,b)$, depending on the networks being studied. In present study, I use a small range of parametrical values for parametrical analysis. More properties may be found by broadening the range of parametrical values. In present algorithm, the addition of connections coincides with the general rule of node connecting (Barabasi and Albert, 1999). However, the mechanism for pruning of connections is still unknown (unknown $q(t,a,b)$), thus in the simplified version of the algorithm, q=0 is a better choice.

Acknowledgment

I am thankful to the support of High-Quality Textbook *Network Biology* Project for Engineering of Teaching Quality and Teaching Reform of Undergraduate Universities of Guangdong Province (2015.6-2018.6), from Department of Education of Guangdong Province, Discovery and Crucial Node Analysis of Important Biological and Social Networks (2015.6-2020.6), from Yangling Institute of Modern Agricultural Standardization, and Project on Undergraduate Teaching Reform (2015.7-2017.7), from Sun Yat-sen University, China.

References

Barabasi AL, Albert R. 1999. Emergence of scaling in random networks. Science, 286(5439): 509

Cancho RF, Sole RV. 2001. Optimization in Complex Networks. Santafe Institute, USA

Watts D, Strogatz S. 1998. Collective dynamics of 'small world' networks. Nature, 393:440-442

Zhang WJ, Zhan CY. 2011. An algorithm for calculation of degree distribution and detection of network type: with application in food webs. Network Biology, 1(3-4):159-170

Zhang WJ. 2012a. Computational Ecology: Graphs, Networks and Agent-based Modeling. World Scientific, Singapore

Zhang WJ. 2012b. Modeling community succession and assembly: A novel method for network evolution. Network Biology, 2(2): 69-78

Zhang WJ, 2015. A generalized network evolution model and self-organization theory on community assembly. Selforganizology, 2(3): 55-64

Zhang WJ. 2016. Selforganizology: The Science of Self-Organization. World Scientific, Singapore